Centro de Investigación y de Estudios
Avanzados del Instituto Politécnico Nacional

Unidad Zacatenco
Departamento de Computación

Design of Multi-Objective Evolutionary Algorithms for Aeronautical Problems

A dissertation submitted by

Alfredo Arias Montaño

For the degree of

Ph.D. in Computer Science

Advisors:
Dr. Carlos Artemio Coello Coello

Dr. Efrén Mezura Montes

México, D.F. June 2012

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

Unidad Zacatenco
Departamento de Computación

Diseño de Algoritmos Evolutivos Multi-Objetivo Para Problemas de Aeronáutica

Tesis que presenta

Alfredo Arias Montaño

para obtener el grado de

Doctor en Ciencias en Computación

Directores de tesis:
Dr. Carlos Artemio Coello Coello

Dr. Efrén Mezura Montes

México, D.F. Junio de 2012

The time will come when diligent research over long periods will bring to light things which now lie hidden. A single lifetime, even though entirely devoted to the sky, would not be enough for the investigation of so vast a subject. And so this knowledge will be unfolded only through long successive ages. There will come a time when our descendants will be amazed that we did not know things that are so plain to them. Many discoveries are reserved for ages still to come, when memory of us will have been effaced. Our universe is a sorry little affair unless it has in it something for every age to investigate. Nature does not reveal her mysteries once and for all.

- Seneca, *Natural Questions*,
Book 7, first century

We do not ask for what useful purpose the birds do sing, for song is their pleasure since they were created for singing. Similarly, we ought not to ask why the human mind troubles to fathom the secrets of the heavens The diversity of the phenomena of Nature is so great, and the treasures hidden in the heavens so rich, precisely in order that the human mind shall never be lacking in fresh nourishment.

- Johannes Kepler, *Mysterium Cosmographicum*

Dedico esta tesis a la memoria de mi madre, Tirsa,
a mi padre, Antonio, y a mis hijos, Adriana y Alfredo (Aristo).

ABSTRACT

Optimization problems in many industrial applications are complex and a time consuming engineering process, that involves the requirement of meeting several design objectives and/or constraints. Most of the times this process is iterative. Optimal design in aeronautical/aerospace engineering is, by nature, a multiobjective-multidisciplinary and highly difficult problem. Aerodynamics, structures, propulsion, acoustics, manufacturing and economics, are some of the disciplines involved in this type of problems. Even if a single discipline is considered, many design problems have competing objectives. Advances in areas such as computational modeling, offer the possibility of accelerating and improving the design cycle by using automated design procedures. In consequence, during the last three decades, the process of aeronautical/aerospace engineering design has been clearly improved because of the dominant role that computational simulations have played in these areas (e.g., Computational Fluid Dynamics (CFD) simulations to perform aerodynamic analysis, Computational Structural Dynamics/Mechanics (CSD/M) through the use of the Finite Element Method (FEM) to process structural analysis).

Additionally, Multi-Objective Evolutionary Algorithms (MOEAs) have gained popularity in recent years as optimization methods in aeronautical/aerospace engineering design, mainly because of their simplicity, their ease of use and their suitability to be coupled to specialized numerical simulation tools. In spite of the considerable amount of research currently available on the use of MOEAs for solving these types of problems, there exists a continuous need to develop new MOEA techniques that can reduce the computational cost, measured in terms of the number of objective function evaluations, required for solving the complex type of problems commonly found in these disciplines.

This thesis addresses the key issue of efficiency of MOEAs when used for solving real-world multi-objective optimization problems, in particular for aeronautical engineering optimization problems, such as the aerodynamic shape optimization of two-dimensional airfoil sections, and/or wing section geometries, or turbine blades. We present several techniques in this regard.

In a first approach, we proposed a novel MOEA based on the use of *Differential Evolution*, which is a metaheuristic with very good convergence properties that has been found to be very effective and efficient when solving single-objective optimization problems, and that has been scarcely explored in the context of aeronautical/aerospace multi-objective optimization problems (MOPs), as indicated from a corresponding review of the field. In our proposed extension of *Differential Evolution* to solve MOPs, we incorporated two mechanisms for improving both the convergence towards the Pareto front and the uniform distribution of nondominated solutions along the Pareto front. These mechanisms correspond to the concept of local dominance and the use of an environmental selection based on a scalar function. Based on the experimental evidence, we conclude that our proposed approach outperforms several state-of-the-art MOEAs in solving numerical MOPs as well as in solving aerodynamic shape optimization multi-objective problems (ASO-MOPs). Our approach was also able to outperform an indicator-based MOEA with respect to the performance indicator, used in its selection process.

In a second approach, we have designed and implemented a parallel version of the first approach. The parallel implementation was based on the island paradigm. This parallel approach was assessed using different numerical MOPs and ASO-MOPs, and their results were compared against the corresponding results obtained with its serial counterpart. From the experimental results, we conclude that our proposed parallel approach is able to improve efficiency in terms of its execution time, i.e., it is able to attain similar Pareto-front approximations, with respect to those attained by its serial counterpart, in less computational time.

Finally, and in order to further improve the efficiency of the two previously proposed approaches, we designed and implemented in a third approach, a surrogate-based multi-objective evolutionary optimization technique. This approach makes use of multiple surrogate models which operate in parallel with the aim of combining their features when solving a costly multi-objective optimization problem. Our proposal was tested in five ASO-MOPs from a defined benchmark defined. From the results of this experimental study, we conclude that this approach can produce a substantial reduction in the number of objective function evaluations performed, reaching savings of up to 75% with respect to the same MOEA but not using surrogates.

RESUMEN

Los problemas de optimización en muchas aplicaciones industriales, son complejos y requieren de un proceso de ingeniería que consumen mucho tiempo, e involucran el satisfacer los requerimientos de múltiples objetivos y/o restricciones. En muchas ocasiones, este proceso es iterativo. El diseño óptimo en las áreas de ingeniería aeronáutica y aeroespacial, es de manera natural, un problema multi-objetivo y multidisciplinario y muy difícil. Aerodinámica, estructuras, propulsión, acústica, manufactura y economÃa, son algunas de las disciplinas involucradas en este tipo de problemas. Aún y cuando se considere una sola disciplina, muchos problemas de diseño poseen objectivos en conflicto. Los avances en áreas como el modelado por computadora, ofrecen la posibilidad de acelerar y mejorar el ciclo de diseño, mediante el uso de procedimientos automatizados. En consecuencia, durante las últimas tres décadas, el proceso de diseño en la ingeniería aeronáutica y aeroespacial, ha sido claramente mejorado, debido al rol dominante que juegan las simulaciones por computadora en estas áreas (por ejemplo las simulaciones de la Dinḿica de Fluidos Computacional (DFC) para realizar análisis aerodinámicos, y ls Dinámica/Mecánica Estructural Computacional (D/MEC) para la realización de análisis estructurales).

Adicionalmente, los Algoritmos Evolutivos Multi-Objetivo (AEMOs) han ganado popularidad en los últimos años como métodos de optimización empleados en tareas de diseño en la ingeniería aeronáutica y aeroespacial debido, principalmente, a su simplicidad, facilidad de uso, y su capacidad de ser fácilmente acoplados con herramientas de simulación numérica especializadas. A pesar de la considerable cantidad de investigación disponible actualmente, sobre el uso de los AEMOs para la solución de este tipo de problemas, existe una necesidad continua de desarrollar nuevas técnicas de AEMOs, que permitan reducir el costo computacional, medido en términos del número de evaluaciones de la función objetivo, requeridos en la solución de problemas complejos, encontrados comúnmente en estas disciplinas.

En esta tesis se aborda la mejora de la eficiencia de los AEMOs, cuando son empleados en la solución de problemas de optimización multi-objetivo que se presentan en el mundo real, en particular en problemas de optimización en la ingeniería aeronáutica, tales como la optimización de formas aerodinámicas en

perfiles aerodinámicos y/o secciones de ala, y geometrías de álabes de turbina. En esta tesis se proponen varias técnicas en este sentido.

En un primer enfoque, se propuso un nuevo AEMO basado en el uso de la metaheurística denominada *Evolución Diferencial*, la cual posee excelentes propiedades de convergencia, además de haber mostrado ser bastante eficiente y efectiva cuando se emplea en la solución de problemas de optimización con un solo objetivo. Adicionalmente, como lo indicó la revisión del estado del arte realizada, esta metaheurística ha sido poco explorada en el contexto de la solución de problemas de optimización multi-objetivo en las áreas de diseño en ingeniería aeronáutica y aeroespacial. En nuestra propuesta de extensión de *Evolución Diferencial*, para la solución de problemas de optimización multi-objetivo, se incorporaron dos mecanismos tendientes a mejorar, por una parte, la tasa de convergencia hacia el frente de Pareto, y por la otra, a obtener una buena distribución de soluciones a lo largo de la aproximación del frente de Pareto. Estos mecanismos corresponden al concepto de dominancia local y al uso de una selección de ambiente basada en funciones de escalarización. Derivado de las evidencias experimentales, podemos concluir que nuestro enfoque propuesto supera a varios AEMOs del estado del arte, en la solución de problemas de optimización multi-objetivo numéricos, así como en la solución de problemas de optimización multi-objetivo de formas aerodinámicas. Nuestro enfoque también es capaz de superar a un AEMO del estado del arte que emplea un indicador de desempeño en su proceso de selección, cuando el rendimiento de ambos algoritmos se mide con respecto a dicho indicador.

En un segundo enfoque, se diseñó e implementó la versión paralela del primer algoritmo. Esta implementación se basó en el paradigma paralelo de islas. Este enfoque paralelo fue evaluado empleando diferentes problemas multi-objetivo numéricos, así como de optimización de formas aerodinámicas, y sus resultados fueron comparados contra los obtenidos con la versión serial del algoritmo. Derivado de los resultados experimentales, concluimos que nuestro enfoque paralelo propuesto es capaz de mejorar la eficiencia en términos del tiempo de ejecución, es decir, es capaz de alcanzar aproximaciones del frente de Pareto similares a las obtenidas con la versión serial, pero en un tiempo de cómputo inferior.

Finalmente, y con el fin de mejorar aún más la eficiencia de los dos enfoques anteriores propuestos, se diseñó e implementó, en un tercer enfoque, una técnica de optimización evolutiva multi-objetivo basada en el uso de meta-modelos. Este tercer enfoque hace uso de múltiples meta-modelos, los cuales operan en paralelo con el objetivo de combinar sus cualidades y ventajas, cuando se emplean en la solución de problemas de optimización multi-objetivo que

tengan un alto costo computacional. Este enfoque fue evaluado en cinco problemas de optimización multi-objetivo de formas aerodinámicas del conjunto de problemas definido en esta tesis. Derivado de los resultados obtenidos en este estudio experimental, concluimos que nuestro tercer enfoque propuesto es capaz de producir una reducción sustancial en el número de evaluaciones de la función objetivo, alcanzado ahorros de hasta un 75% con respecto al mismo AEMO que no hace uso de meta-modelos.

AGRADECIMIENTOS

A continuación deseo expresar mi agradecimiento a diferentes personas e instituciones que hicieron posible la realización del presente trabajo.

Deseo agradecer sinceramente a mis asesores, al Dr. Carlos A. Coello Coello y al Dr. Efrén Mezura Montes, por aceptar dirigir el presente trabajo. También les agradezco profundamente la transmisión de sus conocimientos e ideas, su dedicación, tiempo y cuanto pudieron para la conclusión de este trabajo, y por depositar su confianza en mí.

Agradezco al Dr. Luis Gerardo de la Fraga, al Dr. Debrup Chakraborty, al Dr. Gregorio Toscano Pulido, al Dr. Carlos Eduardo Mariano Romero, y al Dr. Edgar Emmanuel Vallejo Clemente, por aceptar ser revisores de este trabajo, y por sus valiosos comentarios y observaciones.

Agradezco a mis profesores, que en las diferentes etapas de mi formación profesional, me han encausado y motivado a ser cada día mejor, y por la transmisión de sus conocimientos.

A mis padres, les agradezco su amor y cariño, así como el apoyo recibido en diferentes etapas de mivida. A mis hermanos, José Antonio, Juan Manuel, Martha Elizabeth, Lorena y Karen Patricia, gracias por el cariño y apoyo que siempre me han brindado.

Quiero agradecer especialmente a mis compañeros de doctorado, Adriana Lara, Luis Vicente Santana, Antonio López, María de Lourdes López, Saúl Zapotecas, Eduardo Vázquez, Adriana Menchaca, Arturo Yee, Cuauhtémoc Mancillas, con quienes conviví más de cerca durante estos últimos cuatro años y medio. Gracias a todos ellos por los gratos momentos de convivencia, y por su disposición a compartir sus ideas en los ámbitos académico y personal.

También agradezco al personal secretarial del departamento de Computación, Sofia Reza, Felipa Rosas y Erika Ríos por su valioso e incondicional apoyo en diversos trámites administrativos. Gracias también a los auxiliares de investigación, Ing. Arcadio Morales, por su apoyo técnico con el équipo de cómputo personal, y al Dr. Santiago Domínguez por su apoyo en el uso del équipo de cómputo paralelo del departamento, parte importante en los experimentos realizados durante este trabajo.

Agradezco sinceramente a tods aquellas personas que de alguna manera contribuyeron y apoyaron el desarrollo del presente trabajo, pero que por motivo de espacio sus nombres son omitidos.

Deseo agradecer al Instituto Politécnico Nacional y a la Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Ticoman, por brindarme las facilidades y apoyos necesarios para el desarrollo del presente trabajo.

El trabajo de investigación presentado en esta tesis se derivó y fue parcialmente apoyado con fondos del proyecto CONACyT titulado “Escalabilidad y nuevos esquemas hb́ridos en optimizaciń evolutiva multiobjetivo” (Ref. 103570), cuyo responsable es el Dr. Carlos A. Coello Coello. Adicionalmente, la investigación de esta tesis fue parcialmente apoyada con fondos del proyecto CONACyT titulado “Nuevos Modelos Bio-Inspirados en Optimización con Restricciones” (Ref. 79809), cuyo responsable es el Dr. Efrén Mezura Montes.

Agradezco al CONACyT por la beca otorgada durante la realización de estos estudios de doctorado, y muy especialmente al CINVESTAV por ofrecerme un ambiénte académico de calidad y excelencia.

PUBLICATIONS

The publications derived from this thesis are the following:

1.- Alfredo Arias Montaño, Carlos A. Coello Coello and Efrén Mezura-Montes. MODE-LD+SS: A Novel Differential Evolution Algorithm Incorporating Local Dominance and Scalar Selection Mechanisms for Multi-Objective Optimization, in *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, pp. 3284-3291, IEEE Press, Barcelona, Spain, July 18–23, 2010.

2.- Alfredo Arias Montaño, Carlos A. Coello Coello, and Efrén Mezura-Montes. pMODE-LS+SS: An Effective and Efficient Parallel Differential Evolution Algorithm for Multi-Objective Optimization, in Robert Schaefer, Carlos Cotta, Joanna Kolodziej and Günter Rudolph (Editors), *Parallel Problem Solving from Nature–PPSN XI, 11th International Conference, Proceedings, Part II*, pp. 21–30, Springer, Lecture Notes in Computer Science Vol. 6239, Krakow, Poland, September 2010 .

3.- Alfredo Arias Montaño, Carlos A. Coello Coello and Efrén Mezura-Montes. Evolutionary Algorithms Applied to Multi-Objective Aerodynamic Shape Optimization, in Slawomir Koziel and Xin-She Yang (editors), *Computational Optimization, Methods and Algorithms*, Chapter 10, pp. 211–240, Springer, Berlin, Germany, 2011, ISBN 978-3-642-20858-4

4.- Alfredo Arias Montaño, Carlos A. Coello Coello and Efrén Mezura-Montes. *Multi-Objective Evolutionary Algorithms in Aeronautical and Aerospace Engineering*, Accepted for its publication in IEEE Transaction on Evolutionary Computation. Acceptance date: September 7 2011. In press.

5.- Alfredo Arias Montaño, Carlos A. Coello Coello and Efrén Mezura-Montes. Multi-Objective Airfoil Shape Optimization Using a Multiple-Surrogate Approach, in *2012 IEEE Congress on Evolutionary Computation (CEC'2012)*, IEEE Press, Brisbane, Australia, June 10–15, 2012.

The following are some publications in which some ideas from this thesis were contributed.

1.- Luis V. Santana-Quintero, Alfredo Arias Montaño and Carlos A. Coello Coello. A Review of Techniques for Handling Expensive Functions in Evolutionary Multi-Objective Optimization, in Yoel Tenne and Chi-Keong Goh (editors), *Computational Intelligence in Expensive Optimization Problems*, pp. 29–59, Springer, Berlin, Germany, 2010, ISBN 978-3-642-10700-9 .

2.- Saúl Zapotecas Martínez, Alfredo Arias Montaño, Carlos A. Coello Coello: A nonlinear simplex search approach for multi-objective optimization. in *2011 IEEE Congress on Evolutionary Computation (CEC-2011)*, pp. 2367-2374, New Orleans, USA, June 5–8, 2011.

3.- Antonio López Jaimes, Alfredo Arias Montaño, Carlos A. Coello Coello: Preference incorporation to solve many-objective airfoil design problems. in *2011 IEEE Congress on Evolutionary Computation (CEC 2011)*, pp. 1605-1612, New Orleans, USA, June 5–8, 2011.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 PROBLEM STATEMENT

Optimization, by itself, is an important paradigm. It is also an essential part of research, both in science and engineering. In these areas, there are many cases where a research goal can be translated into an optimization problem. In all engineering disciplines we are always trying to optimize a figure of merit. i.e., whether to minimize the production cost and energy consumption for a product, or to maximize its profit, output performance and efficiency. We can state that in today's engineering practice, resources are limited, and designs are subject to many environmental and industrial constraints. In consequence, optimization has become more relevant. The optimal use of the available resources of any sort, requires a paradigm shifting in scientific thinking. This is due to the fact that current real-world applications have far more complicated and interacting factors and/or parameters that substantially affect the whole system performance.

Environmental and industrial demands, have by large transformed the actual process of design in science and engineering. The introduction of massive computing power, as well as the advances of the computational modeling paradigm, have motivated a shift from paper-based analytical system design towards digital models and computer simulations. Computer-aided design optimization is now involved in a wide range of design applications ranging from micro electromechanical systems (MEMS) to large commercial airplanes and space systems. Computational modeling is considered as a third paradigm of modern sciences, complementing the theoretical and experimental studies to problem solving. When coupled to good search algorithms, computational modeling can bring great benefits, leading to a successful optimization process.

With the development of ever increasing powerful optimization techniques, the research community is continually seeking new optimization challenges and is trying to solve increasingly more complicated problems. In real-world engineering problems, there are many areas in which designers are trying to achieve several goals or objectives simultaneously. For instance, in the aeronautical/aerospace industry, given an aircraft's mission and their respective design environmental operating conditions, some common objectives or goals to be minimized include aircraft's weight, cost, and fuel consumption. While all of these objectives are aimed to be minimized, at the same time, aircraft's performance and safety are goals or objectives to be kept at a maximum. From the above, it can be observed that engineers are continuously solving the problem of making trade-offs and producing designs that will satisfy as many requirements as possible, while industry, commercial and ecological standards are at the same time getting ever tighter. This example clearly shows the concept of *Multi-Objective Optimization (MOO)*.

Even when many real-world optimization problems can be reduced to the single-objective optimization case, very frequently it is difficult to capture all aspects of the problem at hand when adopting a single objective. In contrast to this latter approach, defining the optimization problem with multiple objectives, very often helps designers to have a better idea of the engineering optimization problem. Multiobjective optimization problems have been available for more than two decades, and the Operations Research community has developed a variety of approaches to deal with them. However, such approaches normally rely on specific assumptions and features of the problem, and in many cases they are only applicable to continuous objective functions and constraints. When dealing with complex real-world problems having, for example, very large and accidented search spaces, it is necessary to use alternative techniques, such as Evolutionary Algorithms (EAs).

MOO provides designers with trade-off solutions to choose from, in situations in which we aim to fulfill several (conflicting) objectives. The trade-off solutions that are obtained by multi-objective optimization techniques are referred to as the Pareto optimal set and their corresponding objective function values form the so-called Pareto front [41]. This sort of approach contrasts with traditional design optimization techniques, which only produce one (the best possible) solution without providing alternative choices to the designer.

Optimal design in aeronautical/aerospace engineering is, by nature, a multi-objective-multidisciplinary and highly difficult problem. Aerodynamics, structures, propulsion, acoustics, manufacturing and economics, are some of the disciplines involved in this type of problems. Even if a single discipline is considered, many design problems have competing objectives (e.g., to optimize a wing's lift and drag or a wing's structural strength and weight). During the last three decades, the process of aerospace engineering design has been clearly improved because of the dominant role that computational simulations have played in this area [133] (e.g., Computational Fluid Dynamics (CFD) simulations to perform aerodynamic analysis [97, 101], and Computational Structural Dynamics/Mechanics (CSD/M) through the use of the Finite Element Method

(FEM) to process structural analysis [66, 234]). The increasing demand for optimal and robust designs, driven by economic and environmental constraints, along with an increasing computing power, has improved the role of computational simulations, from being just analytical tools until becoming design optimization tools.

In spite of the fact that gradient-based numerical optimization methods have been successfully applied in a variety of aeronautical/aerospace design problems [140, 249, 148, 162, 90, 219] [1], their use is considered a challenge due to many difficulties found in practice: The design space is frequently multimodal and highly non-linear; evaluating the objective function (performance) for the design candidates is usually time consuming; the complexity of the sensitivity analysis in multidisciplinary design optimization increases as the number of disciplines involved becomes larger; among many others.

Based on the previously indicated difficulties, designers have been motivated to use alternative optimization techniques such as (EAs) [157, 169, 132, 54, 55, 178]. Multi-Objective Evolutionary Algorithms (MOEAs) have gained an increasing popularity as numerical optimization tools in aeronautical and aerospace engineering during the last few years [170, 171, 178, 7, 133]. These population-based methods mimic the evolution of species and the survival of the fittest, and, compared to traditional optimization techniques, they present the following advantages: They are robust, provide multiple solutions per run, are easy to parallelize and to hybridize, and, in some cases, are able to find novel solutions.

1.2 MOTIVATION

In spite of the considerable amount of research currently available on the use of MOEAs for solving aeronautical/aerospace engineering optimization problems [10], there exists a continuous need to develop new MOEA techniques that can reduce the computational cost, measured in terms of the number of objective function evaluations, required for solving the complex type of problems commonly found in these disciplines.

The motivation of this research work arises from the need to solve in an efficient way, real-world multi-objective optimization problems (MOPs), in particular for aeronautical engineering optimization problems, such as the aerodynamic shape optimization of two-dimensional airfoil sections, and/or wing

[1] It is worth noting that most of the applications using gradient-based methods have adopted them to find global optima or a single compromise solution for multi-objective problems.

section geometries, or turbine blades. The main interest in this particular type of problems is that they are considered very important in the design of any aircraft and propulsion systems, and are frequently coupled to optimization problems on other areas such as structural optimization.

The main motivation of this thesis was precisely to design and implement a new multi-objective evolutionary algorithm with an emphasis on efficiency. Efficiency here refers to obtaining the highest possible quality of solutions, at the expense of the lowest possible number of objective function evaluations. In this regard, our research focuses on the use of *Differential Evolution*, which is a metaheuristic with very good convergence properties that has been found to be very effective and efficient when solving single-objective optimization problems.

1.3 WORKING HYPOTHESIS

From the previous paragraphs, the main hypothesis for this research work is the following:

> A new multi-objective evolutionary algorithm can be designed and implemented for its application in solving, more efficiently, aeronautical engineering problems, namely, aerodynamic shape optimization problems. In the design and implementation of this new MOEA, the use of efficient metaheuristics, such as *Differential Evolution* or *Evolution Strategies*, as well as the design of evolutionary operators and mechanisms, leading to a more efficient design space exploration; will allow a reduction of the computational cost associated with the number of objective function evaluations, in the type of problems indicated above.

1.4 MAIN AND SPECIFIC GOALS OF THE RESEARCH WORK

This thesis deals with the design of efficient MOEAs and their application to the solution of real-world MOPs commonly found in aeronautical/aerospace system design, namely aerodynamic shape optimization problems. We present next the general and specific goals of this thesis.

1.4.1 *Main goal*

The main goal of this research work is to advance the state-of-the-art with respect to the efficiency of MOEAs for solving real-world MOPs commonly found in the aeronautical/aerospace engineering disciplines. Particularly, we are interested in solving aerodynamic shape optimization problems, but the proposed approach can be extended to other areas of aeronautical/aerospace engineering, and other engineering areas as well.

1.4.2 *Specific goals*

The specific goals of this thesis are the following:

- To gain a deep knowledge of the application of MOEAs when used for solving aeronautical engineering problems. The aim is to identify possible improvements that lead us to the design of a new MOEA, which is more efficient than state-of-the-art MOEAs used in this research field.

- To advance knowledge within evolutionary multi-objective optimization by developing efficient search schemes, based on metaheuristics, having good convergence properties, and by developing search techniques and strategies for better exploring the design search space, present in aeronautical engineering problems.

- To design and implement a new MOEA that is robust and efficient in solving aeronautical engineering problems, particularly aerodynamic shape optimization problems. The aim of the proposed approach, is that it will produce better Pareto front approximations than state-of-the-art MOEAs, for a fixed and reduced number of objective function evaluations.

- To validate the performance of the proposed approach. This validation will be done with benchmarks taken, both, from the available test suites in evolutionary multi-objective optimization, and from well known aerodynamic shape optimization problems found in the specialized literature. Performance will be assessed using measures usually adopted in the evolutionary multi-objective optimization literature.

1.5 DOCUMENT ORGANIZATION

This thesis is organized in 9 chapters. Besides this introductory chapter, the next two chapters are used to present general and background concepts, required, on the one hand, to make this document self-contained, and, on the other hand, to support the contents of the following chapters.

In Chapter 2 we include a general introduction to MOO, and we present the formal definition of Pareto dominance and Pareto optimality. In the second part of Chapter 2 we give a brief description of common approaches used to solve MOPs.

In the first part of Chapter 3, we present general concepts of MOEAs and we give a brief description of the main characteristics of several state-of-the-art MOEAs. They are classified as non Pareto-based, Pareto-based, and Indicator-based MOEAs. Also, we present other metaheuristics that can be considered MOEAs and that have been recently developed. In the second part of Chapter 3 we present general concepts of parallel MOEAs. We include this section because in one of our proposals we make use of the island-based parallel approach.

In Chapter 4 we present a description of the design process, as implemented in aeronautical/aerospace engineering. Next, we present a review on the use of MOEAs, when applied to the design process of aeronautical/aerospace engineering problems. In this review we highlight the type of MOEAs used, the type of MOPs solved, as well as the operators commonly used. In the final part of this chapter we present some remarks on the use of MOEAs, identifying five main types of approaches when implementing MOEAs in aeronautical/aerospace engineering: (1) surrogate-based optimization, (2) hybrid MOEA optimization, (3) robust design optimization, (4) multidisciplinary design-optimization, and (5) data-mining and knowledge extraction.

Chapter 5 is devoted to present our proposed MOEA approaches. In this chapter we present two approaches. The first, called *MODE-LD+SS*, is a MOEA that adopts the evolutionary operators from differential evolution. Additionally, the proposed algorithm incorporates two mechanisms for improving both the convergence towards the Pareto front and the uniform distribution of nondominated solutions along the Pareto front. These mechanisms correspond to the concept of local dominance and the use of an environmental selection based on a scalar function. The second approach corresponds to *pMODE-LD+SS*, which is a parallel MOEA, based on the serial version of *MODE-LD+SS*, and on the island pMOEA paradigm.

Next, in Chapter 6, we describe the experimental setup used to evaluate the performance of our proposed approaches. In this experimental setup we describe the test MOPs used from the available literature, and we provide the definition of seven aerodynamic shape optimization problems, similar to those found in the aeronautical engineering literature. Additionally, we also describe the quality indicators selected for assessing the performance of our proposed approaches and for comparing them with respect to state-of-the-art MOEAs.

In Chapter 7 we present the results of applying the different proposed MOEA approaches, namely *MODE-LD+SS* and *pMODE-LD+SS*, for solving the selected test MOPs defined in Chapter 6. We also apply them to the aerodynamic shape optimization MOPs previously defined. In all cases, the performances obtained with our approaches are compared to those obtained by state-of-the-art MOEAs.

In Chapter 8, we present a third proposed approach consisting of a surrogate method, which is designed to be used in the context of MOEAs. The proposed approach is evaluated, by solving five different aerodynamic shape multi-objective optimization problems defined in Chapter 6. Its performance and its results are compared against those obtained with the serial version of our proposed *MODE-LD+SS*.

Finally, Chapter 9 provides the main conclusions of this thesis, as well as some possible paths for future research.

2 MULTI-OBJECTIVE OPTIMIZATION

2.1 INTRODUCTION

Optimization is a common task in many engineering and scientific disciplines. We conceive *optimization* as a search process in a problem, by means of which, we are willing to attain the *best* result for it, given some figure of merit or performance, prescribed operating conditions and/or constraints. For example in engineering design, common goals are to minimize the production cost or to maximize the desired benefit. It is usual in any optimization process to define the problem in terms of some decision variables, which comprise the vector of decision variables $\vec{x}$, or the *decision vector*.

$$\vec{x} = [x_1, x_2, \cdots, x_n]^T \tag{1}$$

Then the *goal* or *objective* can be defined in terms of this decision vector $f(\vec{x})$. Thus, *optimization* can be defined as the process of finding the conditions in the problem, i.e. the set of values in the decision vector, that give the minimum value for the goal or objective, measured by the objective function $f(\vec{x})$ [1].

Global optimization is the process of finding the global minimum within some search space. The single-objective global optimization problem is formally summarized in the following definition:

Definition 1 ***Global minimum:*** *Given a function* $f : \mathcal{X} \subseteq \mathbb{R}^n \to \mathbb{R}$, $\mathcal{X} \neq \emptyset$, *for* $\vec{x} \in \mathcal{X}$ *the value* $f^* = f(\vec{x}^*)$ *is called a global minimum if and only if:*

$$\forall \vec{x} \in \mathcal{X} : f(\vec{x}^*) \leqslant f(\vec{x}) \tag{2}$$

Then, $\vec{x}^*$ is the global minimum solution, i.e. the *optimal solution*, $f^* = f(\vec{x}^*)$ is the corresponding objective function, and the set $\mathcal{X}$ is the search space.

[1] Without loss of generality, minimization is assumed in the following definitions, since any maximization problem can be transformed into a minimization one, simply multiplying the objective function by -1.

The problem of determining the global minimum solution is called the global optimization problem.

However, there are many real-world problems that involve the simultaneous optimization of *multiple objectives* or goals, which are often incommensurable and in conflict among them. While in *single-objective* optimization problems (SOP) the optimal solution is clearly defined, this same condition does not hold for the *Multi-Objective Optimization Problem* (MOP). Different to the single-objective optimization case, where a single optimum is searched for, in *Multi-Objective Optimization* (MOO), there is rather a set of alternative *trade-offs*, to search for. This set of solutions are generally known as *Pareto-optimal* solutions [13, 88]. Next, some basic concepts for multiobjective optimization are presented. This is followed by a brief description of some common approaches used to solve MOPs.

2.2 MULTI-OBJECTIVE PROBLEM DEFINITION

MOPs, contrary to single-objective optimization ones, might present a possible infinite set of solutions, defined by combinations of values in the decision vector, and when are evaluated, produce vectors of objective functions, whose components represent the trade-offs in the objective space. From this set of solutions, a *decision maker (DM)*, chooses a solution or a set of them based on some preferences for the problem at hand.

A MOP, can be mathematically defined as follows:

Definition 2 ***Multiobjective optimization problem:*** *A general MOP, considers a decision vector $\vec{x} \in \mathcal{X}$ ($\mathcal{X}$ being the decision space) with n parameters or decision variables, a set of k objective functions, and a set of $q = m + p$ constraints, m of which are inequality constraints, and p are equality constraints. Objective functions and constraints are functions of the decision variables. A MOP solution minimizes the components of a vector $\vec{f}(\vec{x})$. Formally,*

$$\text{minimize } \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \tag{3}$$

subject to:

$$g_i(\vec{x}) \leqslant 0\,, \quad i = 1, 2, \ldots, m \tag{4}$$

$$h_i(\vec{x}) = 0\,, \quad i = 1, 2, \ldots, p \tag{5}$$

Evaluating the objective functions in a MOP $f : \mathcal{X} \rightarrow Z$, maps decision variables vectors $\vec{x} \in \mathcal{X}$ onto vectors in the objective function space $Z \subseteq \mathbb{R}^k$. This situation is illustrated in Figure 1 for the case of $n = 2$, and $k = 2$.

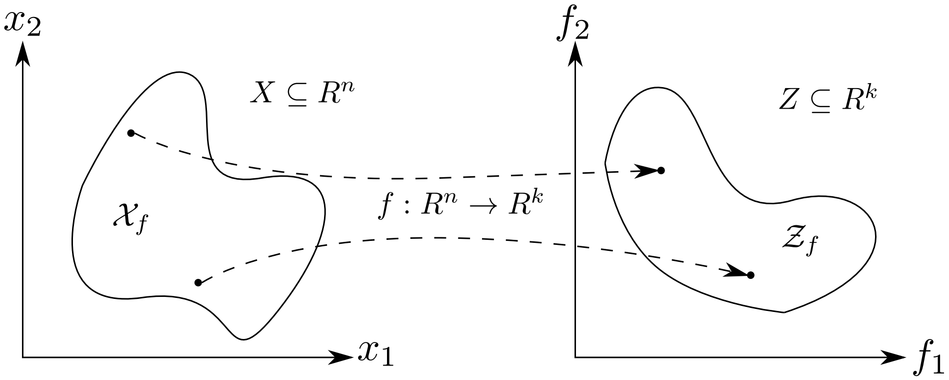


Figure 1: Mapping between decision variables space and objective function space.

Considering the constraints defined in a MOP, next we present the definition for the feasible set.

Definition 3 ***Feasible set:*** *The feasible set* $X_f \subseteq X$ *is defined as the set of decision vectors* $\vec{x}$ *that satisfy the constraints*

$$\mathcal{X}_f = \{\vec{x} \in \mathcal{X} | g(\vec{x}) \leqslant 0 \wedge h(\vec{x}) = 0\} \tag{6}$$

The image of $\mathcal{X}_f$ *gives the corresponding feasible region in the objective space and is denoted as* $\mathcal{Z}_f = f(\mathcal{X}_f) = \bigcup_{x \in \mathcal{X}_f} \{f(\vec{x})\}$

One major characteristic in a MOP is that objectives measure different performances in the problem, and can be dependent or independent and non-commensurable, i.e. they are measured in different units and scales. What makes a MOP difficult is the common situation when the individual optima, corresponding to the distinct objective functions, are sufficiently different. Then the objectives are conflicting and cannot be optimized simultaneously. In fact, "perfect" MOP solutions, where all decision variables satisfy the defined constraints in the MOP, and the objective functions attain a global minimum, may not even exist. The above condition, makes clear that a new notion of optimality is required for MOPs.

2.3 PARETO DOMINANCE

In single-objective optimization, the feasible set is completely (totally) ordered according to the objective function f: for two solutions $\vec{x}, \vec{y} \in \mathcal{X}_f$ either $f(\vec{x}) \leqslant f(\vec{y})$ or $f(\vec{y}) \leqslant f(\vec{x})$. The goal is to find the solution (or solutions) that gives the minimum value of f. However, when several objectives are involved, the situation changes: X_f is, in general, not totally ordered, but partially ordered [186].

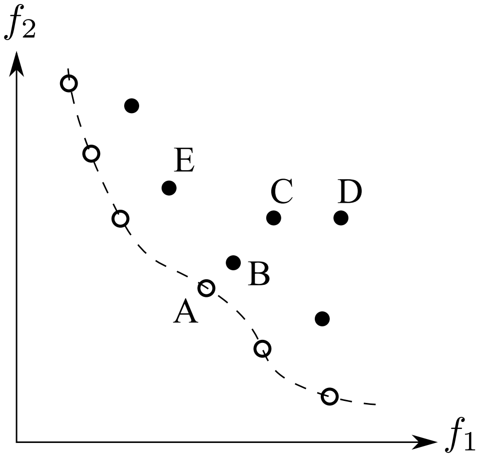


Figure 2: Illustration of the Pareto dominance concept.

This is illustrated in Figure 2. The solution represented by point B is better than the solution represented by point C; it provides better performance in both objectives f_1 and f_2. In any MOP, since trade-offs are searched for, it would be even preferable if it would only improve one objective, as is the case for solutions C and D. In this case, both solutions perform equally in objective function f_2, but solution C provides better performance than D in objective function f_1. In order to express this situation, we present next the definition of Pareto Dominance.

Definition 4 ***Pareto dominance:*** *A vector of decision variables $\vec{x} \in \mathbb{R}^n$ dominates another vector of decision variables $\vec{y} \in \mathbb{R}^n$, (denoted by $\vec{x} \prec \vec{y}$) if and only if $\vec{x}$ is partially less than $\vec{y}$, i.e.,* $\forall i \in \{1, \ldots, k\} : f_i(\vec{x}) \leqslant f_i(\vec{y}) \wedge \exists i \in \{1, \ldots, k\} : f_i(\vec{x}) < f_i(\vec{y})$.

Using the notion of Pareto dominance illustrated in Figure 2, it holds that $B \prec C$, $C \prec D$, and, as a consequence, $B \prec D$. However, when comparing solutions B and E, neither can be said to be superior, since $B \not\prec E$ and $E \not\prec B$.

Although the solution associated with E is better than B in objective f_1, it provides higher value in the objective f_2 than the solution represented by B. Therefore, two decision vectors $\vec{x}$, $\vec{y}$ can have three possibilities within MOPs, regarding the Pareto-dominance relation ($\prec$):

$\vec{x} \prec \vec{y}$ ($\vec{x}$ dominates $\vec{y}$) $\vec{y} \prec \vec{x}$ ($\vec{y}$ dominates $\vec{x}$) $\vec{x} \not\prec \vec{y}$ and $\vec{y} \not\prec \vec{x}$ ($\vec{x}$ is

indifferent to $\vec{y}$)

From the latter condition, we present next the definition for nondominated vector:

Definition 5 ***Nondominated vector:*** *A vector of decision variables* $\vec{x} \in \mathcal{X} \subset \mathbb{R}^n$ *is* **nondominated** *with respect to* $\mathcal{X}$*, if there does not exist another* $\vec{x}' \in \mathcal{X}$ *such that* $\vec{f}(\vec{x}') \prec \vec{f}(\vec{x})$.

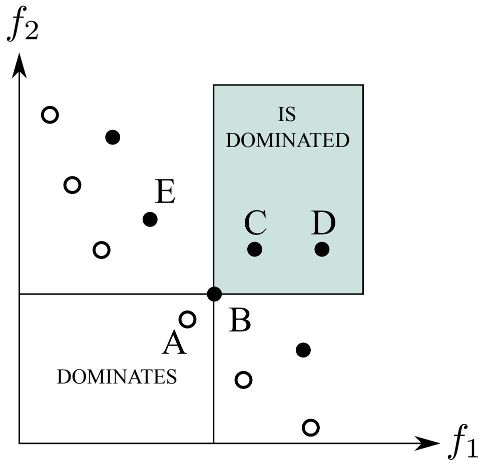


Figure 3: Illustration of Pareto dominance regions in a MOP.

In Figure 3, the shaded rectangle contains the region in objective space that is dominated by the decision vector represented by B; any solution in this region will be dominated by solution B. The clear rectangle contains the objective vectors whose corresponding decision vectors dominate the solution associated with B, as is the case of solution A. All solutions for which the resulting objective vector is in neither rectangle are indifferent or nondominated with respect to the solution represented by B.

2.4 PARETO OPTIMALITY

Once we have defined the concepts of Pareto dominance and nondominated vector, we can introduce the optimality concept for a MOP. From Figure 3 we can say that solution A is nondominated with respect to solutions B, C, D and E. If we consider all the solutions in the figure, we can also conclude that solution A is nondominated with respect to all of them. This condition means that solution A is optimal in the sense that it can not be improved by any other solution, in any objective, without causing a degradation in at least one of them. Therefore all such solutions are denoted as *Pareto optimal*. Next we present its formal definition.

Definition 6 ***Pareto optimality:*** *A vector of decision variables* $\vec{x}^* \in \mathcal{X}_f \subset \mathbb{R}^n$ *(*$\mathcal{X}_f$ *is the feasible region) is* **Pareto-optimal** *if it is nondominated with respect to the other vectors in* $\mathcal{X}_f$*.*

In Figure 3, the circle points represent the Pareto optimal solutions for the MOP, i.e., they are nondominated to each other. Now, the difference in optimality between the single-objective and the MOP must be clear. In a MOP, there is no single optimal solution, but a set of them, which represent the best possible trade-offs among the objectives. No point in this set can be said to be superior to any other in the same set. It is up to the decision maker to select one or a subset of them, based on some preference information for the problem at hand. With this definition of optimality, we present next the corresponding definition for *Pareto optimal set*.

Definition 7 ***Pareto optimal set:*** *The* **Pareto optimal set** $\mathcal{P}^*$ *is defined by:*

$$\mathcal{P}^* = \{\vec{x} \in \mathcal{X}_f | \vec{x} \text{ is Pareto-optimal}\}$$

The Pareto optimal set contains the globally optimal solutions (i.e., all the global nondominated solutions). Similar to the case of single-objective optimization, where local optima can exist in the search space, in the case of MOPs it is possible to have local Pareto optimal sets. Finally, we present the definition of *Pareto front*.

Definition 8 ***Pareto front:*** *The* **Pareto front** $\mathcal{PF}^*$ *is defined by:*

$$\mathcal{PF}^* = \{\vec{f}(\vec{x}) \in \mathbb{R}^k | \vec{x} \in \mathcal{P}^*\}$$

From this definition and from its representation in Figure 4 (the right subfigure), all Pareto optimal solutions lie on the boundary of the feasible objective space.

When solving a MOP the goal is determining the Pareto optimal set from the set $\mathcal{X}_\mathsf{f}$ of all the decision variable vectors that satisfy (4) and (5). Thus, we aim to find not one, but the set of solutions representing the best possible trade-offs among the objectives (the Pareto optimal set).

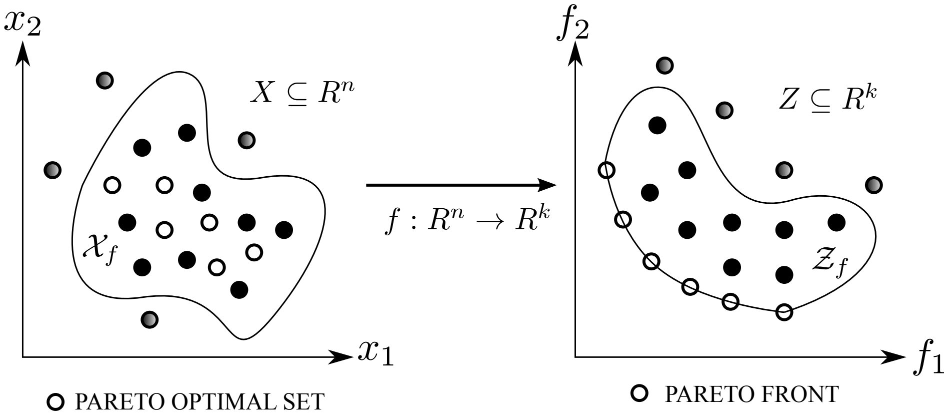


Figure 4: Illustration of the Pareto-optimal Set and its mapping to the Pareto front in objective space.

2.5 APPROACHES TO SOLVE MOPS

The notion of Pareto-optimality and other concepts presented in previous sections, are only the first steps towards solving a multiobjective problem. With this notion, we are able to classify the set of Pareto-optimal solutions, but in some cases we are only able to have an approximation of it. Since in general, MOPs provide a set solutions, in practice the solution of a MOP requires the selection of only one solution among the above set. This is known as the *decision making process* [88]. Depending on how the computation and the decision processes are combined in the search for compromise solutions, three broad classes of multiobjective problem solution approaches can be identified [42]:

- **A priori articulation of preferences:** The decision maker expresses preferences in terms of an aggregating function which combines individual objective values into a single utility value, and ultimately makes the problem single-objective, prior to optimization.

- **A posteriori articulation of preferences:** The decision maker is presented by the optimizer with a set of candidate nondominated solutions, before expressing any preferences. The compromise solution is chosen from that set.
- **Progressive articulation of preferences:** Decision making and optimization occur at interleaved steps. At each step, partial preference information is supplied by the decision maker to the optimizer, which, in turn, generates better alternatives according to the information received.

In these approaches, preferences refer to a decision-maker's opinions concerning points in the objective space. With approaches that involve a posteriori articulation of preferences, the decision-maker imposes preferences directly on a set of potential solution vectors. Then, theoretically the final solution reflects the decision-maker's preferences accurately. On the other hand, with a priori articulation of preferences, one must quantify opinions before actually viewing points in the objective space. In this sense, the term preference often is used in relation to the relative importance given to different objective functions. Nonetheless, this articulation of preferences is fundamentally based on opinions concerning anticipated points in the objective space.

Next, we describe some common methods in each of these approaches. Most of them can be viewed as traditional methods that make use of single objective optimization techniques, and which have been extended to the multiobjective optimization case.

2.5.1 *A priori articulation of preferences*

Under this approach, the most commonly adopted methods correspond to: (a) linear aggregating functions, (b) lexicographic ordering, and (c) goal programming methods.

(a) Linear aggregating functions. For this method, the MOP is converted into a single objective optimization problem defined by the following scalar function:

$$\bar{f} = \sum_{i=1}^{k} \omega_i f_i(x) \tag{7}$$

where $\omega_i \geqslant 0$, $i = 1, 2, \ldots, k$ are the weighting factors representing the decision maker's opinion for each objective (i.e., objective importance).

It is usually assumed that the objectives are normalized, since they can be incommensurable. It is also normally assumed that: $\sum_{i=1}^{k} \omega_i = 1$.

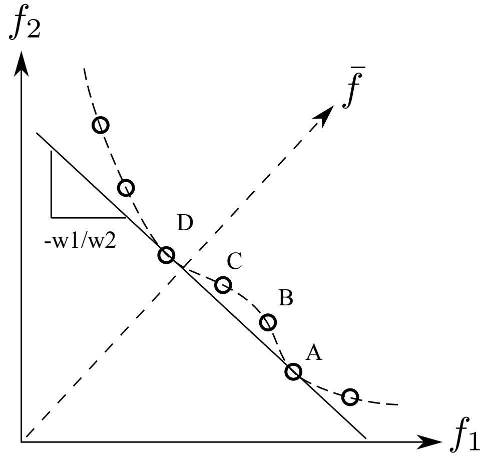


Figure 5: Illustration of the weighted approach method.

One major drawback of this approach is that it can not generate all Pareto optimal solutions in cases where the Pareto front is non-convex. This condition is illustrated in Figure 5. For fixed weights ω_1 and ω_2, a solution is sought to minimize $\bar{f} = \omega_1 f_1(x) + \omega_2 f_2(x)$, which defines a line with slope $-\frac{\omega_1}{\omega_2}$ and intercepts $\frac{\bar{f}}{\omega_2}$ in objective function space (solid line in Figure 5). Graphically, the optimization process corresponds to moving this line downwards until no feasible objective vector is below it and at least one feasible vector is on it, in this case two points: A and D. However, points B and C will never minimize $\bar{f}$.

In order to tackle the previous drawback, the weighted Tchebycheff model can be used instead. This approach uses a reference point z^* which must be beyond the *ideal point* (this point corresponds to that in objective space, having as components, the minimum for each objective function when considered separately). The weighted Tchebycheff model is given by:

$$\bar{f} = \min \max_i [\omega_i |f_i(x) - f_i^*|] \tag{8}$$

The Tchebycheff technique is shown in Figure 6 where the reference point z^* is the origin. The linear aggregating function approach is easy to implement but it is less common than Pareto-based approaches in the current literature.

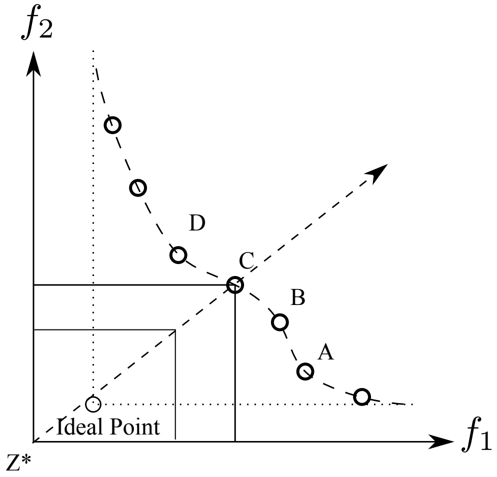


Figure 6: Illustration of the Tchebycheff approach method.

(b) Lexicographic ordering. In this method, the decision maker ranks the objectives in order of importance. Afterwards, the optimum solution is obtained by minimizing the objective functions in sequence from the higher rank objective to the lower rank objective, i.e., the problem is formulated as:

$$\begin{aligned} & min\ f_i(\vec{x}) \\ & x \in \mathcal{X}_f \\ & subject\ to : f_j(\vec{x}) \leqslant f_j(\vec{x^*}), \\ & j = 1, 2, \ldots, i-1,\ i\ >\ 1,\ i = 1, 2, \ldots, k \end{aligned} \tag{9}$$

Here, i represents a function's position in the preferred sequence, and $f_j(\vec{x})$ represents the optimum for the jth objective function, found in the jth iteration. The optimal value found for each objective is now considered as a constraint for subsequent or inferior objective optimizations. In cases in which the decision maker can not rank the importance of the objectives, they can be chosen in a random manner.

(c) Goal programming method [32]. This is one of the earliest techniques specifically designed to deal with multiobjective optimization problems. For this method, the decision maker has to assign targets or goals that wishes to achieve for each objective. These values are incorporated in the problem as additional constraints. The optimizer tries to minimize the absolute deviation from the targets defined for each objective. The simplest form of this method can be formulated as:

$$\min \bar{f} = \sum_{t=1}^{k} |f_i(x) - T_i| \qquad \text{subject to } x \in \mathcal{X}_f \tag{10}$$

where T_i denotes the target or goal set by the decision maker for the objective function $f_i(x)$.

2.5.2 *A posteriori articulation of preferences*

The most common methods associated to this sort of approach correspond to (a) linear aggregating functions, (b) ϵ-constraint method, and (c) normal boundary intersection (NBI).

(a) Linear aggregating functions. In this case, this method corresponds to an extension of the previous one presented in the case of a priori articulation of preferences, but now, the optimizer systematically varies the weight vectors to sample the Pareto front, and instead of only one solution, a set of Pareto optimal solutions is presented to the decision maker. From this set he/she chooses only one solution. The drawbacks of using linear aggregating functions in the previous case, extrapolates to this approach, i.e. no Pareto-optimal solutions are obtained in non-convex regions of the Pareto front. However if the weighted Tchebycheff model is used, the method can be applied to non-convex Pareto fronts.

(b) ϵ-constraint method. For this method, $k-1$ out of k objectives are transformed into constraints. One important aspect for this technique is that is not biased towards convex regions of the Pareto front. The optimization problem, stated for the remaining objective h becomes:

$$\begin{aligned} &\min f_h \\ &\text{subject to}: \ f_i \leqslant \epsilon_i \ (1 \leqslant i \leqslant k, i \neq h) \\ &\text{and } x \in \mathcal{X}_f \end{aligned} \tag{11}$$

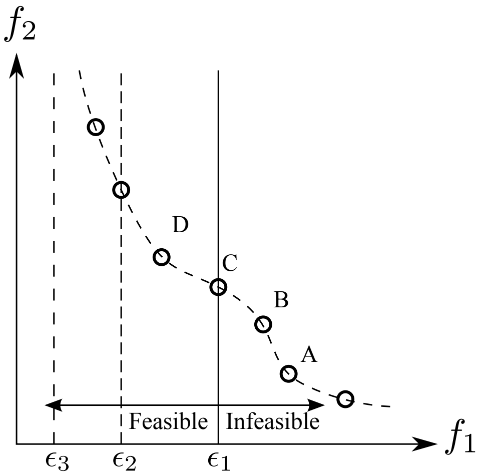


Figure 7: Illustration of the ϵ-Constraint method.

The upper bounds ϵ_i are the parameters that are varied by the optimizer in order to find multiple Pareto-optimal solutions. As illustrated in Figure 7, the ϵ-constraint method is able to obtain solutions associated with non-convex parts of the Pareto front. Setting $h = 2$ and $\epsilon = \epsilon_1$ (solid line in Figure 7) makes the solutions represented by A and B infeasible regarding the constraints set, while the solution C minimizes f_2 among the remaining solutions. One drawback of the method is that if the constraint ϵ is not chosen appropriately, as when $\epsilon = \epsilon_3$ in Figure 7, the feasible set might be empty, i.e., no solution exists for the optimization problem. The decision maker must know suitable ranges of the objective functions for the problem at hand, in order to avoid this latter condition.

(c) normal boundary intersection (NBI) method. It was proposed by Das and Dennis [48] and Das [47] as a response to the defficiencies in the weighted sum approach. The NBI method provides means for obtaining an even distribution of Pareto-optimal points for a consistent variation of a user supplied parameter. One of its advantages is that it works in non-convex Pareto fronts. The method assumes that the global minima f_i^*, $i = 1, 2, \ldots, k$ are known. With these, the convex hull of the individual minima (CHIM) is established. Figure 8 shows the case for a two objective MOP. The basic idea of the algorithm is to find the intersection of the boundary of the feasible vector space, and a direction normal to the CHIM. The algorithm provides an easy mechanism for giving an evenly spread set of points on the Pareto front, by directing the search from points evenly distributed along the CHIM. However if the Pareto front has a very complex shape, the method might identify non Pareto-optimal solutions, and may even find local Pareto-optimal solutions.

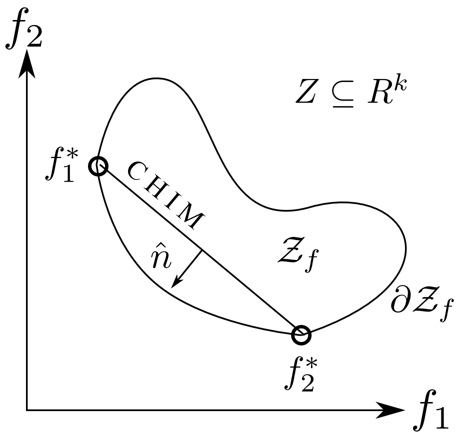


Figure 8: Illustration of the NBI method.

2.5.3 *Progressive articulation of preferences*

These methods work in three main phases:

1. Find a nondominated solution, given some initial preferences provided by the decision maker.
2. Present this solution to the decision maker and get feedback from him/her regarding the quality of the solution, and modify the preferences accordingly, and
3. Repeat steps (1) and (2) above, until the decision maker is satisfied or no further improvement in the solution can be obtained.

These methods are generally referred to as interactive methods. They are designed under the hypothesis that the decision maker is unable to indicate a priori preference information either, because of the complexity of the problem or because of the lack of knowledge about it. During the interactive process, the decision maker learns about the problem as he/she is confronted to different possible solutions to it.

Drawbacks of these approaches are that solutions depend on how well the decision maker can articulate his/her preferences, and a higher effort is required from him/her during the whole search process.

Common methods that correspond to this type of approach include (a) the STEM method, and (b) Steuer's method.

(a) STEM method: This method was proposed by Benayoun et al. [14], and operates using the opinion of the decision maker, in order to reduce the solution space successively. The general optimization problem is restated as a L_p-norm problem (min-max formulation) and uses bounded and weighted objectives:

$$\begin{aligned}
&\min \bar{f} \ = \left\{ \textstyle\sum_{i=1}^{k} \left(\omega_i^h \left(f_i(\vec{x}) - f_i^* \right) \right)^p \right\}^{\frac{1}{p}} \\
&\text{subject to} : x \in \mathcal{X}_f^h \\
&\omega_i^h > 0, \ \textstyle\sum_{i=1}^{k} \omega_i^1 = 1
\end{aligned} \tag{12}$$

In the above expression, h is an iterate counter, p is the parameter in the L_p-norm. The weights are required for solving the min-max formulation, and for equalizing the magnitude of the different objectives. They are not crucial to the outcome of the optimization process, as the final solution depends on the bounds for the objectives, rather than on the variation of the weights. The solution of the optimization problem $\bar{f}$ is compared with the ideal solution f^*. If some components of $\bar{f}$ are acceptable, but some others are not, the decision maker must decide relaxing at least one of the objectives. Then, the upper bound of the jth objective is adjusted to $\bar{f}_j + \Delta f_j$. The solution space $\mathcal{X}^{h+1}$ is reduced by the new constraint $f_j \leqslant \bar{f}_j + \Delta f_j$. The weighting of the jth objective is set to zero, and the optimization problem is solved again, now in the reduced solution space. After the second iteration, the decision maker might be satisfied with the obtained solution, or he/she prefers to relax the boundaries of another function, and the process starts over. In summary, the algorithm progressively reduces the solution space by introducing new constraints on the different objectives.

(b) Steuer's method. This method is based on progressively sampling small subsets of the nondominated set by using progressively changing weights in a weighted sum approach. One example of this type of method is presented by Steuer and Choo [226]. In this method, and for each iteration, the decision maker has to choose between P alternative nondominated solutions. Typically, the number of iterations t is taken as the same number of objectives k in the problem, and $|P| \gtrsim k$. For the algorithm, first the ideal solution f^* is obtained, then it proceeds by minimizing a weighted Tchebycheff metric of the distance between a set of proposed solutions and the ideal solution. For the weightings, a diverse set of weight vectors $\omega \in \Omega$ are used:

$$\Omega = \left\{ \omega \in R^k | \omega_i \in \left[l_i^h, u_i^h \right] \right\} \tag{13}$$

During the solution process, the iteration counter h increases, and the range $\left[l_i^h, u_i^h\right]$ in which the weight vector ω is selected, decreases. In this way, the algorithm progressively focuses on a subset of the nondominated set. The decision maker interacts with the algorithm to guide it towards a portion of the nondominated set that he/she prefers.

2.6 REMARKS FOR MULTIOBJECTIVE OPTIMIZATION

What makes attractive to traditional approaches for solving MOPs, like the ones presented in the previous sections, is that any well-designed and robust algorithm for solving single-objective optimization problems, can be easily adapted and implemented to work with them. Also in the context of global optimization, and during the 60's and 70's, many heuristic methods were developed to be capable of dealing with several complexities and nonlinearities in the fitness landscapes of large scale problems commonly found in real-world engineering optimization. These heuristic-based methods were easy to incorporate in the traditional approaches to solve MOPs. However, many developed techniques were sensitive to the shape of the Pareto-optimal front and/or required previous knowledge of the problem being solved, which in many cases might not be available. Moreover, traditional methods have in common that they need to perform several independent runs to obtain an approximation of the Pareto-optimal set. Thus, synergies can not be exploited, causing high computation overhead.

Recently, Evolutionary Algorithms (EAs) have become an alternative to traditional optimization methods, by means of which, large search spaces can be explored and due to their population-based nature, multiple trade-off solutions for MOPs can be generated in a single run when using them. In the next chapter we present a review of these techniques in the context of the solution of multi-objective optimization problems.

3 MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

3.1 INTRODUCTION

Most real-world optimization applications are hard to solve. Algorithms to solve this type of problems can be so specialized, that they can only be applied to a small range of problems, or they can be instead, more general, but rather inefficient. Some general search heuristics might require high computation time and will eventually fail if the problem's search space is very large, i.e., large scale problems considering the decision variable space. Also, hill-climbing algorithms face problems in searching for optimal solutions when the problem space is multimodal[1], since in such cases, they will get stuck in local optima most of the time.

In order to tackle difficult optimization problems with large, multimodal and accidented search spaces, a number of metaheuristics have been proposed. From the many metaheuristics currently available, Evolutionary Algorithms (EAs) are perhaps the most popular [41]. EAs are inspired by nature and one of its major advantages, as compared to other methods, is that they only need little problem specific knowledge and can, therefore, be applied to a broad range of problems. EAs need a fitness function Φ defined for the problem at hand, which represents the figure of merit that we wish to optimize. Because of their stochastic nature, EAs can be applied to discontinuous, non-differentiable and possibly noisy and/or highly nonlinear search spaces. Because of their flexibility and ease of use, EAs have been used in a wide range of applications [59, 163]. The following are some of the reasons, for which EAs are well suited for complex optimization problems:

- They are improvement-driven. In all cases EAs are designed to continuously improve the fitness function defined in the problem.
- EAs are inherently quantitative, therefore they are well suited for parameter optimization.

[1] Multimodal search spaces contain not only one global optimum but many suboptima, which might deceive a simple search algorithm.

- EAs allow the incorporation of a wide variety of extensions and constraints than cannot be provided in traditional methods.

- EAs are robust, balancing at the same time efficiency and efficacy.

- EAs are easily coupled to other optimization techniques. For example the use of memetic algorithms allows the combination of global and local search processes.

- EAs can naturally be extended to multi-objective optimization problems [41, 248].

EAs are stochastic search and guided optimization heuristics derived from natural principles, namely from evolutionary theory. The basic idea is that if only the fittest individuals of a population can reproduce, and the other individuals die, the whole population will improve (in terms of fitness), and we will eventually converge to very fit individuals (which correspond to the global optimum or a very good approximation of it). If an additional mechanism such as mutation is added, the population can also explore the search space and new individuals are likely to be generated with an increased selection probability and they can inherit this property to their descendants. In summary, the population dynamics follows the basic rule of Darwinian evolutionary theory, as stated next:

As many more individuals of each species are born than can possibly survive; and as, consequently, there is a frequently recurring struggle for existence, it follows that any being, if it varies however slightly in any manner profitable to itself, under the complex and sometimes varying conditions of life, will have a better chance of surviving, and thus be naturally selected (On the Origin of Species by Means of Natural Selection, 1859)

The above statement can be described in short as the “survival of the fittest” principle from Darwin’s evolutionary theory.

To solve optimization problems with an EA, the individuals of a population have to represent a possible solution (in terms of some decision variables vector $\vec{x} = [x_1, x_2, \ldots, x_n]$) for a given problem and its selection probability is set proportional to the quality of the represented solution using the objective function $f(\vec{x})$ defined in the problem at hand, i.e., the quality of the represented solution is expressed in terms of a *fitness* function Φ for the individual. An EA follows the basic scheme shown in Algorithm 1.

In the first step, a population of random solutions is created, and then the EA generational loop is entered. At each generation, the individual’s fitness Φ_i is

Algorithm 1 Basic EA Algorithm

```
$t \leftarrow 0$
Generate an initial population $P(t = 0)$
while Stopping criterion not met do
    Evaluate the fitness $\Phi_i$ for each individual in $P(t)$
    Compute the probability selection $p_i$ for each individual in $P(t)$ based on its fitness value $\Phi_i$.
    Select the fittest $P'(t)$ as parents from $P(t)$
    Apply the Evolutionary Operators (EVOPs) of crossover and mutation to create an offspring population $P(t + 1)$.
    Apply elitism
    $t \leftarrow t + 1$
end while
```

obtained which is further used to assign a probability of selection p_i. Individuals with higher fitness values will receive higher selection probabilities, and will be able more times to participate in the creation of offspring (descendants) for the next generation. From the selected parents, an offspring population is created by applying the Evolutionary Operators (EVOPs), namely crossover and/or mutation. In all EAs, descendants can be imperfect clones of the parents with small variations (this corresponds to the naturally occurring mutations in nature), or the descendants are a combination of multiple parents (this corresponds to the sexual reproduction in nature) or both. Finally, a form of elitism must be introduced to avoid losing good individuals, through the use of EVOPs. Elitism passes the best individual(s) in the population intact to the following generation.

An important advantage of EAs is that they don't start the search from a single solution but depart, instead, of a whole population of (randomly generated) solutions. The use of a population makes EAs more resistant to premature convergence towards a local optima in multimodal search spaces. However, the stochastic nature makes it hard to guarantee convergence to the global optimum, except for very specific cases. Nevertheless, their population-based nature and their stochastic operators make EAs good candidates to solve complex (mainly nonlinear) MOPs. Next we present some basic characteristics required for the design of multi-objective evolutionary algorithms, and give a brief description of some representative MOEAs created since the early days of this research area.

3.2 MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

It is worth indicating that traditional EAs require some modifications in order to deal with multi-objective optimization problems. The main two are the following:

1. All the nondominated solutions should be considered equally good by the selection mechanism. This means that a different notion of fitness is required for dealing with multi-objective optimization problems. The most popular mechanism to deal with this problem is called Pareto ranking and was introduced by Goldberg [75]. This approach assigns a rank to each solution based on its Pareto dominance, such that nondominated solutions are all sampled at the same rate. However, in the early days of MOEAs, several mechanisms not based on Pareto optimality were adopted with EAs [41].

2. EAs tend to converge to a single solution if run long enough, because of stochastic noise [75]. Therefore, a mechanism to maintain diversity is required. This component is known as the *density estimator*. Fitness sharing [76] was the earliest density estimator, but many others have been proposed over time, including clustering [272], entropy [64], adaptive grids [127] and crowding [51], among others.

MOEAs can be classified in several ways [41]. However, for the purposes of this work, we decided to adopt a simple high-level classification that considers only three types of MOEAs: (a) Non-Pareto-based, (b) Pareto-based, and (c) Indicator-based. The first group contains MOEAs that do not adopt the concept of Pareto optimality in their selection mechanism, whereas the second comprises those MOEAs that adopt Pareto optimality in their selection mechanism. Finally, the third group considers more recently developed MOEAs which adopt a performance measure in their selection process.

3.2.1 *Non-Pareto based algorithms*

Some of the most popular non-Pareto-based MOEAs are the following:

- **Lexicographic method**: The user ranks the objectives of the problem in a decreasing order and the optimization proceeds from higher to lower order objectives, one at a time. Once an objective is optimized, the aim is to improve as much as possible the following objective(s) without decreasing the quality of the previous one(s) [41]. This sort of approach normally generates a single nondominated solution, but if instead of using a fixed objective as the most important, it is randomly chosen, several solutions can be generated in one run.[2]

- **Aggregating functions**: All the objectives are added up into a single (scalar) value which constitutes the objective to be optimized. Since objectives tend to be defined in very different ranges, a normalization is normally required. Also, weights tend to be assigned to each objective in order to define preferences from the user [41]. Varying the weights during the run allows, in general, the generation of different nondominated solutions in one run [111, 81].

- **Population-based methods**: A number of sub-populations (usually as many as the number of objective functions of the problem) are generated from a main population of an EA. Each sub-population optimizes a single objective function and then all the sub-populations are merged and mixed. The aim is that, when performing crossover, individuals that are good in one objective will recombine with individuals that are good in another one [215]. This sort of approach produces several nondominated solutions in a single run, but it typically misses good compromises among the objectives because of the way in which individuals are selected in each population [41].

3.2.2 *Pareto based algorithms*

Among the Pareto-based methods, there are two sub-classes: the non-elitist MOEAs and the elitist MOEAs. Non-elitist MOEAs do not retain the nondominated solutions that they generate and could, therefore, lose them after applying the evolutionary operators. Elitist MOEAs retain these solutions either in an external archive or in the main population.

The most representative non-elitist MOEAs are the following:

- **Nondominated Sorting Genetic Algorithm** (NSGA): It was proposed by Srinivas and Deb [225]. It is based on several layers of classifications of the individuals. Before selection is performed, the population is

[2] The reader can find similarities with the approach described in the previous chapter. However in that chapter, the discussion was related to the use of approaches in which traditional mathematical programming methods are used, and in this case the use of evolutionary algorithms is implied. The same comment applies to the aggregating function described in this chapter.

ranked on the basis of nondomination: all nondominated individuals are classified into one category (with a dummy fitness value, which is proportional to the population size, in order to provide an equal reproductive potential for these individuals). To maintain the diversity of the population, these classified individuals are shared with their dummy fitness values. Then this group of classified individuals is ignored and another layer of nondominated individuals is considered. The process continues until all individuals in the population are classified. Since individuals in the first front have the maximum fitness value, they always get a higher selection probability than the rest of the population. Figure 9 illustrates the Pareto ranking and sharing mechanisms used in NSGA. The σ_{share} value is a parameter defined by the user. Consequently, the algorithm's performance is strongly dependent on the choice of it, sometimes resulting in less efficient performance of NSGA.

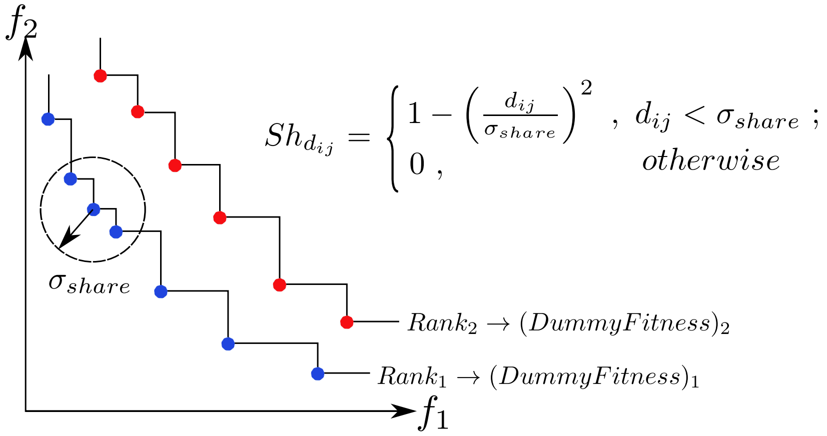

f_2
f_1
σ_{share}
$Sh_{d_{ij}} = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^2 & , d_{ij} < \sigma_{share} ; \\ 0 , & otherwise \end{cases}$
$Rank_2 \rightarrow (DummyFitness)_2$
$Rank_1 \rightarrow (DummyFitness)_1$

Figure 9: Illustration of the Pareto Ranking and Sharing mechanisms used in NSGA

- **Niched-Pareto Genetic Algorithm** (NPGA): Proposed by Horn et al. [89]. It uses a tournament selection scheme based on Pareto dominance. The basic idea of the algorithm is the following: Two individuals are randomly chosen and compared against a subset from the entire population (typically, around 10% of the population). If one of them is dominated (by the individuals randomly chosen from the population) and the other is not, then the nondominated individual wins. When both competitors are either dominated or nondominated (i.e., there is a tie), the result of the tournament is decided through fitness sharing [76].

- **Multi-Objective Genetic Algorithm** (MOGA): This algorithm was proposed by Fonseca and Fleming [68]. For this approach, the rank of a certain individual corresponds to the number of individuals in the current population by which it is dominated plus one. All nondominated individuals are assigned the lowest possible rank (i.e., one), while dominated ones receive as rank the number of individuals that dominate them plus one, i.e. $\mathrm{rank}(i) = 1 + q_i$ where q_i is the number of individuals that dominate individual i in the objective space. Also in this algorithm, fitness sharing is implemented and applied in the objective space in order to obtain a good distribution of solutions along the Pareto front. One important aspect in the sharing mechanism is that MOGA calculates its value depending on the population size and on the current maximum and minimum values of the objectives. Figure 10 illustrates the ranking mechanism used in MOGA.

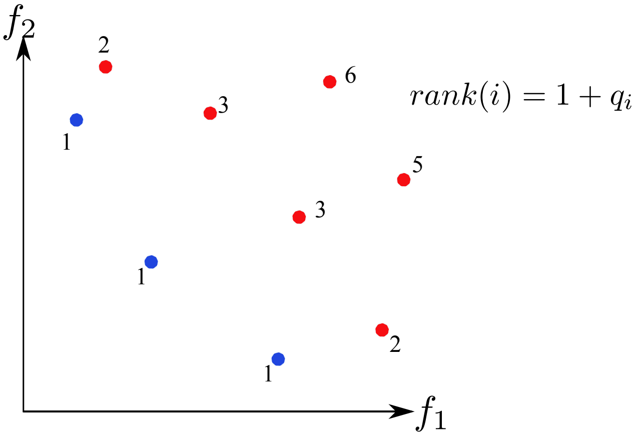


Figure 10: Illustration of the ranking mechanism used in MOGA

Among the most popular Pareto-based elitist MOEAs, we have the following:

- **Strength Pareto Evolutionary Algorithm** (SPEA): Zitzler and Thiele introduced this MOEA in [272]. It uses an archive containing nondominated solutions previously found (the so-called external nondominated set). At each generation, nondominated individuals are copied to the external nondominated set, removing the dominated solutions. For each individual in this external set, a *strength* value is computed. This strength is similar to the ranking value of MOGA [68], since it is proportional to the number of solutions to which a certain individual dominates. The fitness of each member of the current population is computed according to the strengths of all external nondominated solutions that dominate it. The fitness assignment process of SPEA considers both closeness to the true Pareto front and even distribution of solutions at the same time. Thus, in SPEA, instead of using niches based on distance (as MOGA and NPGA), Pareto dominance is adopted to ensure that the solutions are properly distributed along the Pareto front. Although this approach does not require a niche radius, the effectiveness of this approach relies on the size of the external nondominated set, since such a set participates in the selection process of SPEA. In fact, since the external nondominated set participates in the selection process of SPEA, if its size grows too large, it might reduce the selection pressure, thus slowing down the search. Because of this, the authors decided to adopt a technique that prunes the contents of the external nondominated set so that its size remains below a certain threshold. The approach adopted for this sake was a clustering technique called "average linkage method" [164].

- **Strength Pareto Evolutionary Algorithm 2** (SPEA2): It was proposed by Zitzler et al. [274] and has three main differences with respect to its predecessor [272] : (1) it incorporates a fine-grained fitness assignment strategy which, for each individual, takes into account both the number of individuals to which it dominates and the number of individuals that dominate it; (2) it uses a nearest neighbor density estimation technique which guides the search more efficiently, and (3) it has an enhanced archive truncation method that guarantees the preservation of boundary solutions.

- **Pareto Archived Evolution Strategy** (PAES): This algorithm was introduced by Knowles and Corne [129]. PAES consists of a (1+1) evolution strategy (i.e., a single parent that generates a single offspring) in combination with a historical archive that records the nondominated solutions previously found. This archive is used as a reference set against which each mutated individual is being compared. Such a historical archive is the elitist mechanism adopted in PAES. However, an interesting aspect of this algorithm is the procedure used to maintain diversity which consists of a crowding procedure that divides objective space in a recursive manner. Each solution is placed in a certain grid location (see Figure 11) based on the values of its objectives (which are used as its "coordinates" or "geographical location"). A map of such grid is maintained, indicating the number of solutions that reside in each grid location. Since the procedure is adaptive, no extra parameters are required (except for the number of divisions of the objective space). The archive is fixed in size, and once its upper bound is reached, a new generated solution is inserted and the archive is pruned, by deleting individuals in the most crowded grid cells.

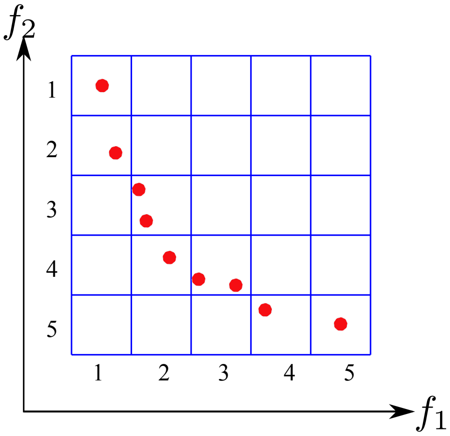


Figure 11: Illustration of the grid archive used in PAES

- **Nondominated Sorting Genetic Algorithm II** (NSGA-II): This approach was proposed by Deb et al. [51] as an improved version of the NSGA. In the NSGA-II, solutions are ranked using a nondominated sorting scheme, and the density of solutions surrounding a particular solution in the population is estimated by computing the average distance of two points on either side of this solution along each of the objectives of the problem. This value is the so-called *crowding distance* (see Figure 12). During selection, the NSGA-II uses a crowded-comparison operator which takes into consideration both the nondomination rank of an individual in the population and its crowding distance (i.e., nondominated solutions are preferred over dominated solutions, but between two solutions with the same nondomination rank, the one that resides in the less crowded region is preferred). The NSGA-II does not use an external memory as the other MOEAs previously discussed. Instead, the elitist mechanism of the NSGA-II consists of combining the best parents with the best offspring obtained (i.e., a $(\mu + \lambda)$-selection). Due to its clever mechanisms, the NSGA-II is much more efficient (computationally speaking) than its predecessor, and its performance is so good, that it has become very popular in the last few years, becoming a landmark against which other MOEAs have to be compared [266].

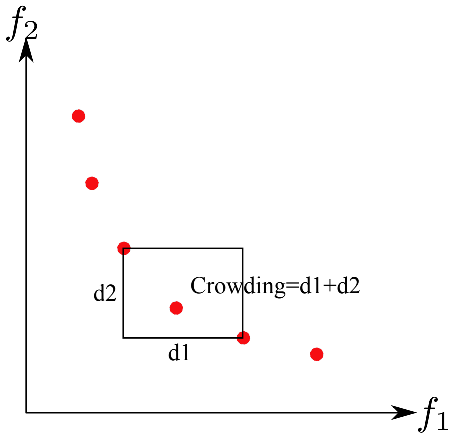


Figure 12: Illustration of the crowding mechanism used in NSGA-II

3.2.3 *Indicator-based algorithms*

In the multi-objective optimization scenario the main goal is to find a good approximation of the Pareto-optimal solutions, however no formal definition exists for what a good approximation means. From the previous descriptions of several developed MOEAs we can distinguish two main characteristics attained in the approximation to the Pareto-optimal set, namely: (i) To minimize the distance to the true Pareto-optimal set, which in many real-world MOPs, is not known; and (ii) to maximize the diversity along the Pareto-optimal set. Both these characteristics by themselves can be considered a MOP with two objectives [50, 41]. In most popular MOEAs, they are solved in terms of Pareto ranking of the individuals for the first characteristic, and in terms of some density information such as niching and crowding for the second characteristic.

However, for each MOEA and depending on their implemented mechanisms, different user preferences are implicitly assigned to each characteristic. Since different MOEAs might give also different outcomes, i.e. different approximations to the Pareto-optimal set, their comparison is based on different performance measures proposed in the specialized literature [79, 273, 124, 275]. In this sense, more recently designed MOEAs have considered the use of a performance measure or indicator during the evolution process, more precisely either in their mating selection mechanism and/or in their environmental selection mechanism. Here we refer to this type of MOEAs as Indicator-based MOEAs. Next we describe two algorithms (IBEA and SMS-EMOA) belonging to this class.

- **Indicator-Based Evolutionary Algorithm** (IBEA): It was proposed by Zitzler and Künzli [270]. The main idea of this algorithm is to first define the optimization goal in terms of a binary performance measure or indicator, and then to directly use this measure in the selection processes. This MOEA can be considered as a general indicator-based one, since any binary indicator can be used in the fitness assignment function for each one of the solutions in the current population. The fitness function definition is:

  $$\mathtt{Fitness}(\vec{x}) = \sum_{\vec{y} \in P \setminus \{\vec{x}\}} -e^{-I(\{\vec{x}\},\{\vec{y}\})} / \kappa \tag{14}$$

  In equation (14), P is the actual population and κ is a scaling factor which needs to be defined by the user and depends on the problem being solved. Also, this fitness function definition requires that the binary quality indicator $I(\{\vec{x}\},\{\vec{y}\})$ be dominance preserving[3]. Next, we present the formal definition for a dominance preserving binary quality indicator.

  Definition 9 *A binary quality indicator* $I(\{\vec{x}\},\{\vec{y}\})$ *is denoted as dominance preserving if (i)* $\vec{x} \prec \vec{y} \Rightarrow I(\{\vec{x}\},\{\vec{y}\}) < I(\{\vec{y}\},\{\vec{x}\})$, *and (ii)* $\vec{x} \prec \vec{y} \Rightarrow I(\{\vec{z}\},\{\vec{x}\}) \geqslant I(\{\vec{z}\},\{\vec{y}\})$ *for all* $\vec{x}$, $\vec{y}$, $\vec{z} \in \mathcal{X}$.

  With this condition, the fitness assignment scheme is also Pareto-dominance compliant [275]. The fitness assignment mechanism tries to rank the population members according to their usefulness, regarding the reformulated optimization goal, i.e., to maximize/minimize the performance measure or indicator. In summary, the proposed fitness function measures the *"loss in quality"* in the binary quality indicator if solution $\vec{x}$ is removed from the actual population.

  In the basic algorithm, IBEA performs binary tournaments for mating selection, and implements environmental selection by iteratively removing the worst individual from the population, in terms of the binary quality indicator measure, and updating the fitness values of the remaining individuals in the population. In their proposed approach, authors make use of the binary addition ϵ-indicator $I_{\epsilon+}$ and the Hypervolume inidicator I_{HV}. One particular aspect of IBEA is that for both, mating selection and environmental selection processes, comparisons are made in a pairwise

[3] Some binary quality indicators with this property can be found in [275].

sense, reducing in consequence the computational overhead in computing the binary indicator values.

- **S-Metric Selection - Evolutionary Multi-Objective Algorithm** (SMS-EMOA): It was proposed by Beume et al. [17]. For this algorithm the hypervolume (or S-Metric) contribution is used in the environmental selection process. SMS-EMOA is a steady state algorithm in which only one solution is created at a time and inserted into the actual population for performing the environmental selection. Then, for each solution in the extended population, its contribution to the hypervolume measure is computed as the difference of the hypervolume measure with and without it. This difference is assigned as fitness to each solution in the population:

$$Fitness(\vec{x}) = I_{Hv}(P) - I_{Hv}(P\backslash\{\vec{x}\}) \tag{15}$$

In equation (15), P corresponds to the extended population, i.e., including the newly generated solution. Since the maximization of the hypervolume measure attains both goals of convergence towards the Pareto-optimal solutions and a good distribution of solutions along the Pareto-front approximation [67, 130], the solution with the less contribution to the Hv measure is then discarded. In Figure 13, the basic ranking mechanism of solutions used in SMS-EMOA is illustrated. In this figure the number close to each solution corresponds to its rank, based on the hypervolume contribution, which is depicted as the shaded area to the right of each solution. Also, in this figure it can be observed that the Pareto extreme solutions receive the first k higher ranks (k is the number of objectives in the MOP) in order to avoid losing them.

At the beginning of the evolutionary process, many solutions in the current population can be dominated and, therefore, they do not contribute to the hypervolume measure of the Pareto-front approximation. For these cases, the SMS-EMOA algorithm relies on the Pareto ranking approach used in the NSGA-II algorithm, and the hypervolume measure contribution is computed for each rank layer of solutions. In consequence, the discarded solution will be selected as the less contributing in the hypervolume measure but in the highest rank layer.

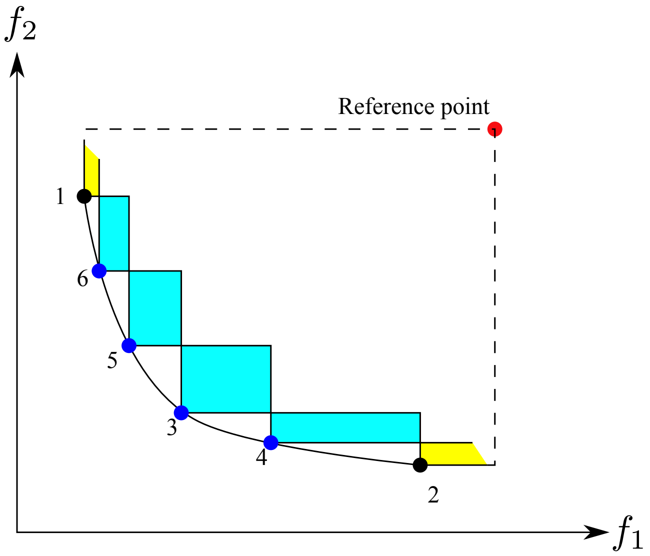


Figure 13: Illustration of the ranking mechanism of solutions based on the hypervolume contribution used in SMS-EMOA.

3.3 OTHER METAHEURISTICS

There are several other multi-objective metaheuristics available. Next, three of them are briefly discussed:

- **Particle Swarm Optimization** (PSO): This metaheuristic is inspired on the choreography of a bird flock which aim to find food [121]. It can be seen as a distributed behavioral algorithm that performs (in its more general version) a multidimensional search. The implementation of the algorithm adopts a population of particles, whose behavior is affected by either the best local (i.e., within a certain neighborhood) or the best global individual. PSO has been successfully used for both continuous nonlinear and discrete binary optimization [63]. For extending PSO to deal with MOPs, the main issues are: (1) how to select particles (to be used as leaders) in order to give preference to nondominated solutions over those that are dominated?, (2) how to retain the nondominated solutions found during the search process in order to report solutions that are nondominated with respect to all the past populations and not only with respect to the current one?, and 3) how to maintain diversity in the swarm in order to avoid convergence to a single solution? Normally, mechanisms very similar to those adopted with MOEAs (namely, Pareto-based selection and external archives) have been adopted in multi-objective particle swarm optimizers (MOPSOs). However, the addition of other mechanisms (e.g., a mutation operator) is also relatively common in MOPSOs. An important number of multi-objective versions of PSO currently exist (see for example [201]), and this remains as a very active research area.

- **Differential Evolution** (DE): This metaheuristic was proposed by Kenneth Price and Rainer Storn [227, 189] to optimize problems over continuous domains. The core idea is to use vector differences for perturbing a vector population, and it aims to estimate the gradient in a region (rather than in a point). DE performs mutation based on the distribution of the solutions in the current population. In this way, search directions and possible step sizes depend on the location of the individuals selected to calculate the mutation values. Several DE variants are possible, and they differ in the way in which the parents are selected and in the form in which recombination and mutation takes place (see [189] for more information on DE). The high success of DE in single-objective optimization has made it an interesting candidate for solving MOPs. The main issues for extending DE to multi-objective optimization are very similar to those of PSO (i.e., how to select parents, how to store nondominated solutions and how to maintain diversity in the population). As with MOPSOs, very similar mechanisms to those adopted by MOEAs have been used with multi-objective differential evolution (MODE). A variety of MODE approaches currently exist (see for example [161]), and this also remains as a very active research area . It is worth noting that MODEs are often considered MOEAs [41].

- **MOEA based on Decomposition** (MOEA/D): The multiobjective evolutionary algorithm based on decomposition [266] is a recent multiobjective evolutionary algorithmic framework. It is based on conventional aggregation approaches where a MOP is decomposed into a number of scalar objective optimization problems (SOPs). The objective of each SOP, also called subproblem, is a (linearly or nonlinearly) weighted aggregation of the individual objectives. Neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors. Subproblem i is a neighbor of subproblem j if the weight vector of subproblem i is close to that of subproblem j. Each subproblem is optimized in MOEA/D by using information mainly from its neighboring subproblems. In a simple version of MOEA/D, each individual subproblem keeps one solution in its memory, which could be the best solution found so far for the subproblem. It generates a new solution

> by performing genetic operators on several solutions from its neighboring subproblems, and updates its memory if the new solution is better than the old one for the subproblem. A subproblem also passes its newly generated solution on to some (or all) of its neighboring subproblems, which will update their current solutions if the received solution is better. A major advantage of MOEA/D is that scalar objective local search can be used in each subproblem in a natural way since its task is optimizing a scalar objective subproblem.

Although many other MOEAs exist (see for example [40, 265]), it is not the intention of this chapter to be comprehensive. The interested reader may refer to [41, 50, 268] for more information on this topic. The main advantages of MOEAs are their generality, ease of use and the fact that they require little or no specific domain information to operate. Also, they are less susceptible to the specific features of the problem (e.g., shape or continuity of the Pareto front) than traditional mathematical programming techniques [41].

3.4 PARALLEL MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Since MOEAs can solve hard MOPs, they have become increasingly popular in this research area, and many research efforts are continuously conducted both, for solving mathematical and real-world optimization problems [41, 38]. Once the abilities of MOEAs for generating multiple tradeoff solutions have been made evident, researchers have become interested in improving their efficiency and their effectiveness, in terms of how fast or cheaply the MOP can be solved. This natural aim in reducing the execution time or the computational resources needed by the MOEA, automatically lead researchers to consider MOEA parallelization and distributed processing in solving MOPs.

A major computational bottleneck in many MOEA applications to real-world design MOPs is the excessive amount of time required to evaluate both, the objective functions as well as the constraints defined in the MOP. This condition is aggravated when considering the population-based nature of any MOEA. In consequence, MOEA parallelization might be one of several possible alternatives for improving their computational efficiency. With this sort of technique, "expensive" (in terms of CPU time) objective/constraint function evaluation can be completed in less wall clock time if the computational load is distributed over several processing units. On the other hand, if some fixed computational time is allowed, parallel processing might also help to evaluate more solutions in the MOEA, i.e. improving its effectiveness and resulting in a better (possibly larger and higher fidelity) Pareto-optimal set approximation. This later condition is beneficial in any MOP solving, since the aim is to identify a possible large set of Pareto-optimal solutions. Therefore, a parallel MOEA or pMOEA, for short, might be the preferred choice of implementation for solving complex real-world applications where computationally intensive objective/constraint functions are the bottleneck.

3.4.1 *pMOEA motivation*

In designing pMOEAs for complex real-world applications, one option is to apply parallel function decomposition techniques, or an approach in which the MOEA population is decomposed or distributed spatially across a given set of processors. The first approach refers to the case of evaluating the objective function(s) in parallel and using several processors. This case is very common in many scientific and engineering areas, where the problem is so complex that a parallel simulation code is used in order to have solutions in a reasonable time frame. Finally, the reader can anticipate that in some cases the combination of both approaches might also be used.

In order to fully benefit from a pMOEA design, one must first identify the MOEA components that can be concurrently executed and, in consequence, are subject to be parallelized. In the parallelization approaches referred above, parallelizing the objective function evaluation(s) is a simple and potentially useful idea, but in this case only the MOEA efficiency is improved by reducing the execution time. However, effectiveness improvement is another issue that can be considered in any pMOEA design. Although some pMOEAs are more effective than their serial counterpart, this effectiveness improvement does not come without a cost, which may be in some cases an increased execution time. These are tradeoffs researchers must have to consider when designing a pMOEA.

In summary, the motivation behind a pMOEA design, is to find as good or better MOP solutions in less time than its serial MOEA counterpart, use less resources, and/or search more of the solution space, i.e., increasing both efficiency and effectiveness. With these considerations in mind, we describe next three main pMOEA computational paradigms commonly used and that correspond to: (a) master-slave, (b) island, and (c) diffusion. Some authors [246, 154] consider a fourth paradigm, namely “hierarchical” or “hybrid”, which may sometimes be seen as a combination of the other three.

3.4.2 *Master-slave pMOEA paradigm*

In this paradigm, objective function evaluations are distributed among several slave-processors, while there exists a master-processor which is in charge of executing tasks such as applying EVOPs, Pareto-ranking, mating and environmental selection, distributing/collecting of subpopulations, etc. This can be seen as the simplest pMOEA paradigm and is also fairly simple to implement. However, its search and exploration abilities are identical to the case of its corresponding serial MOEA. In this paradigm, the number of slave-processors used has a direct influence only in the execution time, as long as the time communication for transferring data between processes is very small as compared to the corresponding time needed in the objective/constraint function evaluation. The scheme of this paradigm is graphically illustrated in Figure 14.

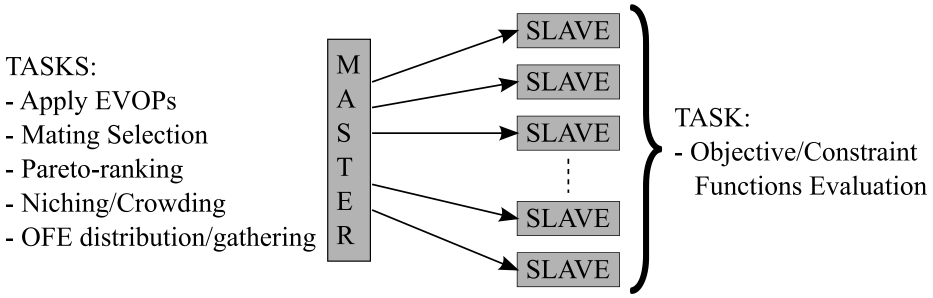


Figure 14: Example of the master-slave pMOEA paradigm

As indicated before, the master-process is in charge of controlling and orchestrating the global pMOEA process execution, and distributing the objective evaluations and finally gathering their results. Nonetheless, this master process can also participate in performing some objective function evaluations. It is important to note that in this paradigm, if the objective function evaluations are complex and time consuming, a substantial computational speedup can be obtained. Also in this paradigm, the objective function distribution can be implemented in three different ways:

(1) The pMOEA population is evenly distributed in the slave processors and all of them evaluate the k objectives and the q constraint functions.

(2) The pMOEA population is evenly distributed by sets of members in the population, across k sets of slave processors to perform one of the k objective function evaluations, and,

(3) For each member of the pMOEA, evenly distribute each objective function evaluation across multiple processors.

In using the master-slave paradigm for a pMOEA design, a detailed analysis of the objective/constraint function evaluation time is required since, otherwise, there is a risk of arriving to a poor load balancing across different slave-processors, with a considerable detriment in pMOEA efficiency. Finally, in this sense, we can say that efficiency is the main and only objective searched for in this paradigm.

3.4.3 *Island pMOEA paradigm*

This paradigm is inspired on the phenomenon of natural populations evolving in relative isolation, such as in the case of ocean island chains with limited migration. Since this paradigm is frequently implemented on distributed memory computers, it is also referred to as distributed pMOEAs (also called multiple-population or multiple-demes pMOEAs). For this paradigm, different communication topologies can be implemented (for example ring, mesh, torus, triangle, hypercube, etc.). Evidently different population dynamics, and in consequence, different population models will result, as a consequence of the different communication topologies used. Figure 15 illustrates some examples for the island pMOEA paradigm.

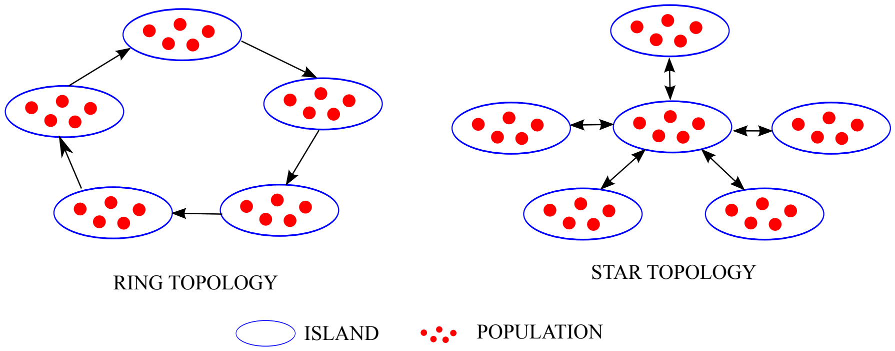


Figure 15: Example of the island pMOEA paradigm

The pMOEA island paradigm can be seen as an overall population divided into a number of independent and separate subpopulations or demes. An alternative view is to see this paradigm as a set of multiple MOEAs executing simultaneously with a small population size each. Also in this paradigm, migration of individuals can occur at certain intervals or epochs in the evolutionary process. Then, the pMOEA design using this paradigm must accordingly identify migration and replacing policies, the number of individuals that will migrate, the migration policies (i.e., who will migrate and who will be replaced) as well as the interval or epoch at which the migration will take place. The selection of these parameters will allow a thorough gene mixing within each deme or island, but will also restrict/permit gene flow between different islands. By now, it can be anticipated that the island pMOEA paradigm is very attractive, since the use of different EVOPs, MOEA parameters, random number generators and seeds, and even the use of multiple MOEAs, can enhance the search process in the MOP design space. In this regard, four different island pMOEA schemes can be identified:

(1) Homogeneous island pMOEA. In this case all islands execute identical MOEA/parameters,

(2) Heterogeneous island pMOEA. For this situation, all island execute different MOEA/parameters,

(3) Each island evaluates different objective functions subsets, and

(4) Each island can represent and/or search different regions of the genotype or phenotype domains.

3.4.4 *Diffusion pMOEA paradigm*

In this paradigm, each processor holds only one or very few MOEA solutions as a population. The diffusion term for this paradigm comes from the neighborhood communication structure between processes. In this case EVOPs are applied only within these defined neighborhoods, which in some cases can overlap, and their geometry and/or communication topology can be a square, a rectangle, a cube or other shape depending on the number of dimensions associated with the pMOEA topological design. The philosophy behind this paradigm is that when good solutions appear in different areas or regions of the local topology, their genetic information is spread or slowly diffused through the entire population due to the overlapping and/or communication in the neighborhoods defined.

This pMOEA paradigm is illustrated in Figure 16 where a grid communication topology is used and the neighborhood is defined on a square with four processors (in the left), and with a diamond shape neighborhood with five processors (in the right). In this paradigm there is no migration like in the island paradigm but genetic information is shared in the neighborhood. Also, the communication, depending on the topology, can be very costly within a neighborhood, since in some cases, the genetic information required for the EVOPs cannot be retrieved directly from an adjacent processor, but has to be obtained through another processor. Similar to the island pMOEA, different communication topologies can be devised in this case such as ring, torus, etc.

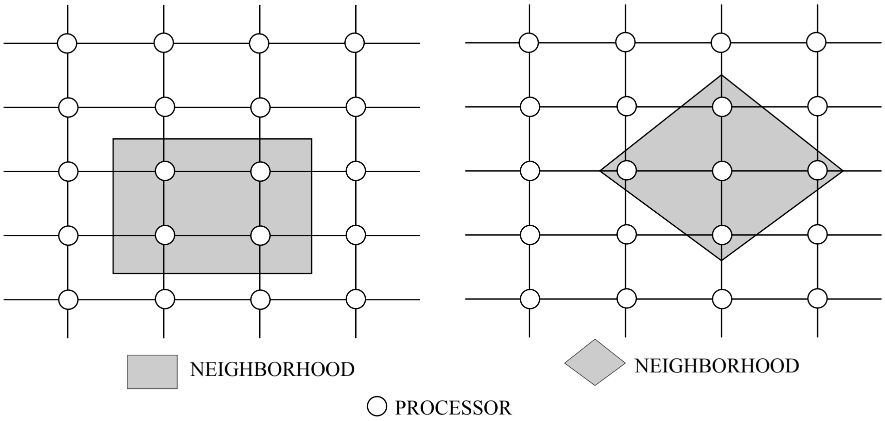


Figure 16: Example of the diffusion pMOEA paradigm.

4

MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS IN AERONAUTICAL/AEROSPACE ENGINEERING

4.1 AERONAUTICAL/AEROSPACE ENGINEERING DESIGN PROCESS

In aeronautical/aerospace engineering there is always a trend for designing and building more complex systems/designs, which can be attributed to ever increasing technology improvements. Simultaneously to this trend, there is also a pressure for developing aeronautical/aerospace systems/designs which are environment friendly, faster, and which are also available at competitive prices, and meet high quality standards. In order to satisfy these market requirements and/or constraints, aeronautical/aerospace manufacturing companies are focusing their efforts on the product development processes. In this sense, one major aim has been to improve the efficiency of the development process itself, and many methods have been implemented to analyze and to manage the design process [118, 177, 205]. Another issue has been to develop tools and techniques that efficiently supports the design of these complex systems/designs, which has produced a wealth and sophisticated computerized engineering tools [99, 120, 204, 71]. As the computational capacities of computers increase, the fidelity of the simulations and numerical optimization, and their common use in the process of new design efforts, also increases.

In spite of the advantages of using computer simulations and numerical optimizations, as indicated above, a great part of the design process remains intuitive and the participation of a human decision-maker is still needed. For this decision making process, analytical techniques, simulation models, and numerical optimizations, are nowadays of great value and allow to obtain substantial improvements in many aeronautical/aerospace designs, in a timely fashion and with less costs, as compared to the same practice of aeronautical/aerospace engineering design 30 years ago.

Aeronautical/aerospace engineering design is an iterative process where new design proposals are generated and evaluated. The iterative part of the design process or design cycle (cf. Figure 17) consists of: *synthesis*, *analysis*, and *decision* [199, 25]. In this iterative design process, the current design(s) is(are) evaluated, i.e., its (their) expected performance(s) is(are) obtained using either analytical tools, numerical simulations or experiments, which are then com-

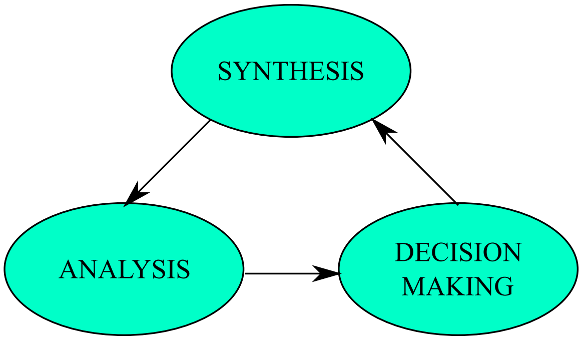


Figure 17: Principal activities in the design cycle

pared to the requirements of the proposed system design. If the proposed design(s) does(do) not meet the requirements, it is (they are) modified and evaluated again in search for the best possible design(s). Figure 18, reproduced from Raymer [199] and defined as the "design wheel", clearly shows the iterative design process, commonly used by aeronautical/aerospace companies.

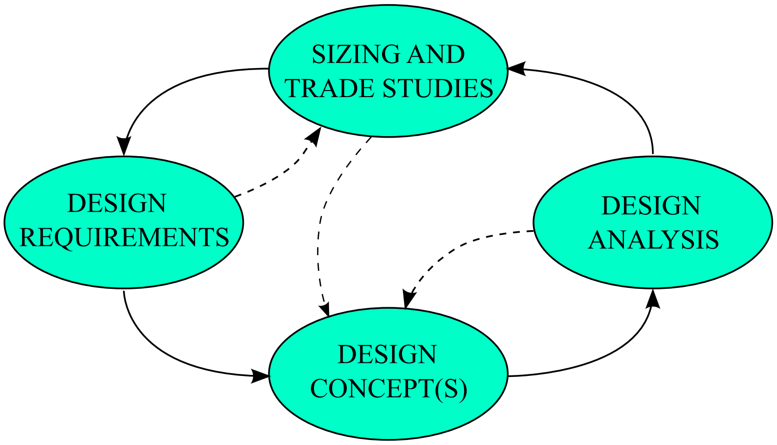


Figure 18: Iterative design process or "design wheel" [199].

The aeronautical/aerospace engineering design process comprises three major phases: (i) *conceptual design*, (ii) *preliminary design*, and (iii) *detailed design* [199, 25]. In each of these phases (see Figure 19), design concepts are analyzed to determine their compliance with the performance requirements, as well as their manufacturability and economical viability. As previously indicated, the design process cannot be considered as serial, but as a cyclic process, in which many design iterations are required. This iterative process is mainly executed between the first two phases where numerical optimization has its greatest impact, and where the goal of optimization is to refine the design, prior to the *detailed design* phase in which design production is initiated. Next, we give a brief description of the characteristics of these three phases.

- **Conceptual design**: Being the conceptual design phase the earliest one in the design process, it has an emphasis on finding the best design concepts, ensuring designers that they are heading into the correct design path, guaranteeing to meet all design performance requirements and a minimal development cost. A key aspect of this phase is that is a very fluid process, and the general design layout is continuously being changed, both to incorporate knowledge of the design problem, obtained from the analysis of the starting concept(s) and to evaluate potential improvements to the current design. In this phase, tradeoff studies are very important to balance the aeronautical/aerospace systems performance and/or constraints, because the design(s) substantially evolves in a short period of time. Also important in this design phase is that a finite number of possible designs, instead of only one, are studied to determine which design approach will be preferred. The optimization methods used during the conceptual design phase focus on tradeoffs studies and on the overall design characteristics rather than in finer details of the concept. Due to the evolving nature of the design in the conceptual design phase, the use of sophisticated analysis tools is precluded, mainly because of the time and computational requirements of them. Instead, robust classical analysis methods are used.

- **Preliminary design**: The preliminary design phase can be said to start when major changes in the designs are done. i.e., concept(s) were found to meet all the requirements and constraints established for the design of the aeronautical/aerospace system; or the design requirements have been updated to attain a feasible design in terms of the present/near-future technology availability and/or development costs. The design configuration arrangement of the aeronautical/aerospace system can be expected to remain about as shown on current design drawings, although minor revisions may occur. At some point late in the preliminary design phase, even minor changes are stopped when a decision is made to frozen the aeronautical/aerospace system configuration. It is during the preliminary design phase where specialists of the individual disciplines such as aerodynamics, structures, control, propulsion, etc. will design and analyze their respective portion of the aeronautical/aerospace system. The ultimate goal of the preliminary design phase is to prepare the company for the detailed design phase, also called the full scale development.

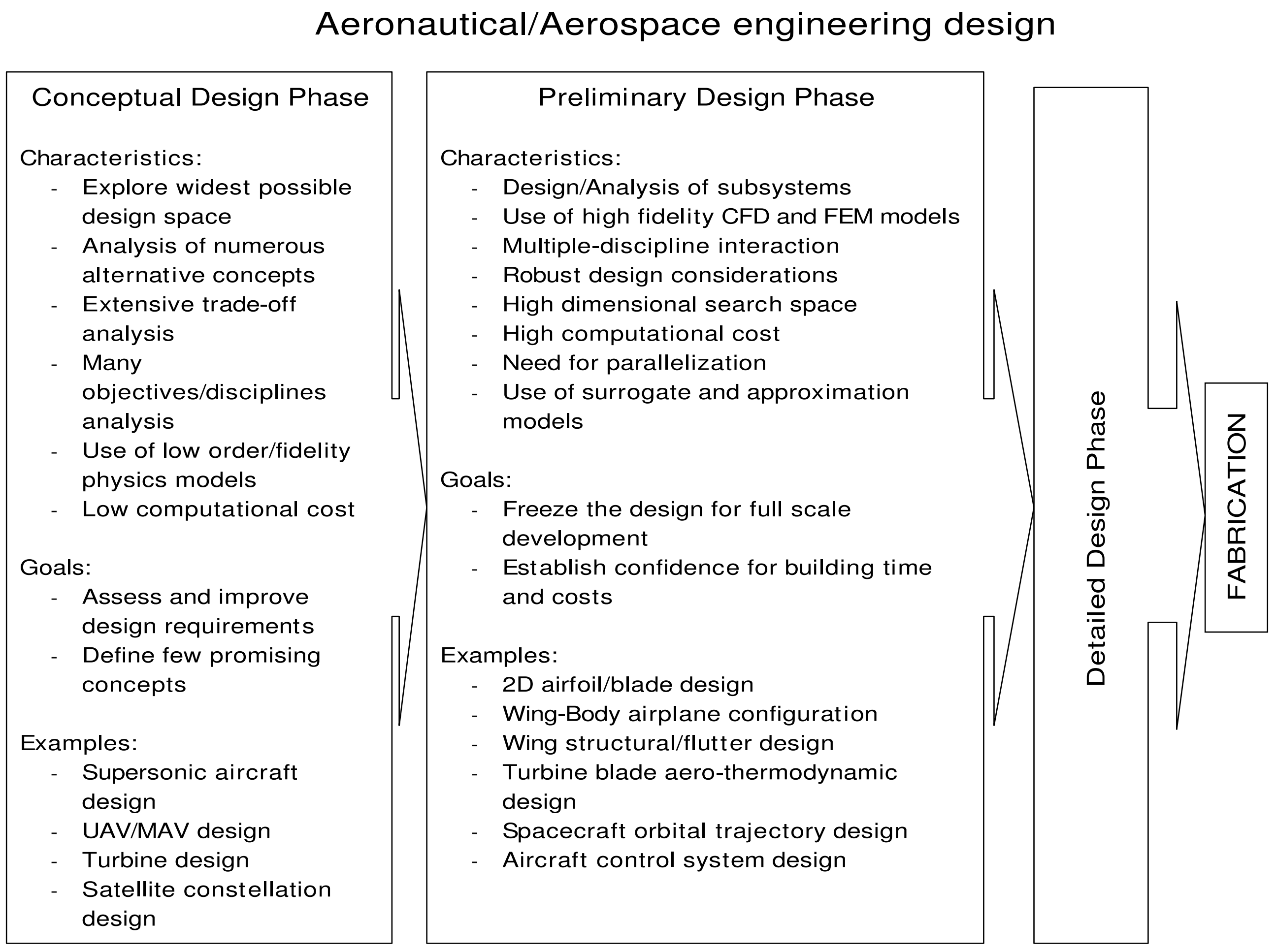

Aeronautical/Aerospace engineering design
Conceptual Design Phase
Characteristics:
- Explore widest possible design space
- Analysis of numerous alternative concepts
- Extensive trade-off analysis
- Many objectives/disciplines analysis
- Use of low order/fidelity physics models
- Low computational cost
Goals:
- Assess and improve design requirements
- Define few promising concepts
Examples:
- Supersonic aircraft design
- UAV/MAV design
- Turbine design
- Satellite constellation design
Preliminary Design Phase
Characteristics:
- Design/Analysis of subsystems
- Use of high fidelity CFD and FEM models
- Multiple-discipline interaction
- Robust design considerations
- High dimensional search space
- High computational cost
- Need for parallelization
- Use of surrogate and approximation models
Goals:
- Freeze the design for full scale development
- Establish confidence for building time and costs
Examples:
- 2D airfoil/blade design
- Wing-Body airplane configuration
- Wing structural/flutter design
- Turbine blade aero-thermodynamic design
- Spacecraft orbital trajectory design
- Aircraft control system design
Detailed Design Phase
FABRICATION

Figure 19: Major phases in aeronautical/aerospace engineering design

The preliminary design phase is characterized by a maturation of the selected design approach. During this phase, which takes the longest time from the whole design process, the design evolves with an ever-increasing level of understanding, as well as an ever-increasing level of design and analysis detail, giving designers a stronger confidence that the design will work. It is also during this phase when, specialists who are experts in the various design disciplines and aeronautical/aerospace subsystems are given the overall design concept and are asked to evaluate and to optimize the design concept in their areas of expertise. Being aeronautical/aerospace systems complex designs, during this phase of the design, it is common to find modifications to the proposed design, as defined in the conceptual design phase, requiring further iterations and refinements of the design concept. Following such revisions, the design optimization must be redone because any change to the design layout will likely affect the inputs and hence the outputs for any optimization process.

- **Detailed design**: This phase in the aeronautical/aerospace design process begins with the design of the actual pieces to be fabricated. This last part of the design process is characterized by a large number of designers preparing detailed drawings or CAD files with actual fabrication geometries and dimensions. Also during this last phase, thousands of small pieces not considered in the two preceding design phases are designed and built. Every small piece of the whole aeronautical/aerospace system must be designed in detail, hence the name of this last design phase. Another important aspect of the detailed design is called production design. In this process, specialists define how the aeronautical/aerospace system will be fabricated, considering the smallest and simplest subassemblies and building up to the final assembly processes. It can happen at this phase that production designers wish to modify the design for ease of manufacture, having in consequence a major impact on system's performance and/or cost. Compromises are inevitable, but the design must still meet the original requirements. In order to produce satisfactory designs, a common practice is to use a systems thinking and design approaches [30], or a design for manufacture approach [44, 23, 57].

4.2 OPTIMIZATION PROCESS IN AERONAUTICAL/AEROSPACE ENGINEERING DESIGN USING MOEAS

Based on the previous paragraphs, it could be seen that aeronautical/aerospace engineering design is essentially an optimization process. In order to automate and therefore to speed up the whole design process, the optimization tasks need to be formalized and the use of an optimization algorithm needs to be introduced into the design process. Design search and optimization is the term used to describe the application of formal optimization methods to the problem of engineering design. It is important to recognize that *'search'* indicates that along with the aim for optimal designs, the design activity is in many cases an exploratory process where there are no fixed endpoints and no obviousl optimal solutions to the complex problem being dealt with in any aeronautical/aerospace engineering system.

Optimal design in aeronautical and aerospace engineering is, by nature, a multiobjective, multidisciplinary and highly difficult problem. Aerodynamics, structures, propulsion, acoustics, manufacturing and economics, are some of the disciplines involved in this type of problems. In fact, even if a single discipline is considered, many design problems in aeronautical/aerospace engineering have conflicting objectives (e.g., to optimize a wing's lift and drag or a wing's structural strength and weight). Motivated by industrial and economical demands, the process of engineering design in aeronautics has undergone a major transformation, and during the last three decades, this process has been clearly improved because of the dominant role that computational simulations have played in this area [133] (e.g., Computational Fluid Dynamics (CFD) simulations to perform aerodynamic analysis and design [97, 98, 100, 101] and Computational Structural Dynamics/Mechanics (CSD/M) through the use of the Finite Element Method (FEM) to process structural analysis [66, 234]). The increasing demand for optimal and robust designs, driven by economics and environmental constraints, along with the advances in computational intelligence and the increasing computing power, have improved the role of computational simulations, from being just analysis tools until becoming design optimization tools.

In spite of the fact that gradient-based numerical optimization methods have been successfully applied in a variety of aeronautical/aerospace design problems,[1] [140, 249, 148, 162, 90, 219] their use is considered a challenge due to the following difficulties found in practice:

[1] It is worth noting that most of the applications using gradient-based methods have adopted them to find global optima or a single compromise solution for multi-objective problems.

1. The design space is frequently multimodal and highly non-linear.

2. Evaluating the objective function (performance) for the design candidates is usually time consuming, due mainly to the high fidelity and high dimensionality required in the simulations.

3. By themselves, single-discipline optimizations may provide solutions which not necessarily satisfy objectives and/or constraints considered in other disciplines.

4. The complexity of the sensitivity analyzes in Multidisciplinary Design Optimization (MDO[2]) increases as the number of disciplines involved becomes larger.

5. Unlike single-discipline optimization, in MDO no unique solution is obtained. Instead, a set of trade-off solutions are searched for.

Based on the previously indicated difficulties, designers have been motivated to use alternative optimization techniques such as EAs [157, 169, 132, 54, 55, 178]. MOEAs have gained an increasing popularity as numerical optimization tools in aeronautical and aerospace engineering during the last few years [170, 171, 178, 7, 133]. These population-based methods mimic the evolution of species and the survival of the fittest, and compared to traditional optimization techniques, they present the following advantages:

(a) *Robustness:* In practice, they produce good approximations to optimal sets of solutions, even in problems with very large and complex design spaces. Instead of a single-point search with gradient information, MOEAs use a population of design candidates (i.e., they perform a multipoint search) and are less prone to get trapped in local optima. Additionally, they can manage non-differentiable, mixed real-discrete and highly non-linear objective functions/fitness landscapes.

(b) *Multiple solutions per run:* As MOEAs use a population of candidates, they are designed to generate multiple trade-off solutions in a single run. Evidently, the generation of more solutions also involves a higher computational time when dealing with expensive applications. Thus, the number of solutions to be generated by a MOEA in the applications discussed in this chapter tends to be low, unless surrogate models are adopted.

[2] Multidisciplinary Design Optimization, by its nature, can be considered as a multi-objective optimization problem, where each discipline aims to optimize a particular performance metric.

(c) *Easy to parallelize:* The design candidates in a MOEA population, at each generation, can be evaluated in parallel using different paradigms. This can be useful in problems involving objective functions that are costly to evaluate (something common in aeronautical and aerospace applications).

(d) *Simplicity:* MOEAs use only the objective function values for each design candidate. They do not require a substantial modification or complex interfacing for using a CFD (Computational Fluid Dynamics) or CSD/M (Computational Structural Dynamics/Mechanics) code. This situation substantially reduces the cost related to code writing and tuning every time a new application is envisaged. Furthermore, designers can easily make use of in-house developed and/or commercial codes previously validated.

(e) *Easy to hybridize:* Along with the simplicity previously stated, MOEAs also allow an easy hybridization with alternative methods, e.g., memetic algorithms, which additionally introduce specifities to the implementation, without significantly affecting the simplicity of MOEAs.

(f) *Novel solutions:* In many cases, gradient-based optimization techniques converge to designs which have little variation even if produced with very different initial setups. In contrast, the inherent explorative capabilities of MOEAs allow them to produce, some times, novel and non-intuitive designs.

An important volume of information has been published on the use of MOEAs in aeronautical and aerospace engineering applications (mainly motivated by the advantages previously addressed). In this chapter, we provide a review of some representative works. From this review, we can observe the variety of design stages and diverse problems in which MOEAs have been used.

4.2.1 *Use of MOEAs for conceptual design optimization*

Traditionally, the aeronautical/aerospace *conceptual design* phase has been conducted with the help of databases, statistics, and regression/low-order engineering models as well as company's/designer's accumulated experience. The main outcome of this design phase has been to determine a few promising *design concepts* to be further analyzed in the *preliminary design* phase, in which numerical simulations or experimental setups are developed to verify and refine the design. Additionally, tradeoff analyzes are performed in order to identify unreasonable or conflicting requirements. This latter task has been limited because of the large design spaces that needs to be explored, and a holistic (multidisciplinary) vision of the design is required when multiple disciplines are involved in the design. Nowadays, with the increasing computing power available, low-cost/fidelity numerical simulations have spread toward the *conceptual design* phase, making it possible to benefit from the *exploration* of large design spaces with reduced time and low computational cost. Additionally, it is possible to envision performing *trade-off* analysis of the multi-objective and/or multidisciplinary designs. Both of these characteristics are inherent in the use of MOEAs for the class of applications reported next:

- Oyama and Liou [180, 182] addressed the conceptual design of rocket engine pumps, for a centrifugal single and multi-stage pump design. In both cases two objectives were defined: (i) maximization of total head in the pump, and (ii) minimization of the pump input power. Side constraints were considered for the design variables range, defining the pump geometry. An additional operating constraint was imposed for the static pressure at the rotor tip in order to detect the inception of cavitation, being crucial to prevent this condition for the optimal design. The authors adopted MOGA with fitness sharing [76], blended crossover (BLX-α) and uniform random mutation. Conceptual designs were evaluated using a one dimensional meanline pump flow-modeling method, which provides a fast modeling of turbopumps for rocket engines at very low computational cost. For the first conceptual design case, a total of 498 different nondominated solutions were obtained, while 660 were found in the second case. Authors noted that improvements in the objective functions were within 1% in both objectives with respect to a reference design.

- Buonanno and Mavris [29] addressed the conceptual design of a small supersonic aircraft, considering seven objectives: (i) weight, (ii) range, (iii) takeoff balanced field length, (iv) loudness, (v) overpressure, (vi) flight Mach number, and (vii) cabin size. Some of them were minimized, while others were maximized. An application example presented by the authors comprised a set of up to 64 design variables (both continuous and discrete variables were considered), describing the aircraft geometry and the mission requirements. The authors used a parallel hybrid subjective/quantitative MOEA, in which the fitness of an individual was a combination of both quantitative and qualitative metrics, with the latter being defined by a human evaluator. A parallel-MOEA) (pMOEA), based on the injection island genetic algorithm [58], was adapted for this MOP. The strategy consisted on assigning one objective function per island and solving a two-objective optimization problem. The second objective for each island was constructed as a goal attainment metric based on the mission requirements for the aircraft. In this way, each island obtained a set of solutions excelling in its assigned objective and representing a trade-off with respect to the project goals. After a certain number of generations, the nondominated solutions from the islands were sent to a central island which solved the seven-objective problem formulated as a goal attainment problem. Each island used SPEA2. The nondominated solutions from the central island were transferred back to each of the islands and the process was repeated until satisfactory solutions were obtained. The authors used physics-based analysis tools for performance prediction. Low-order/fidelity models were used for the involved disciplines: aerodynamics, propulsion, stability and control, economics, aeroelasticity, manufacturing and acoustics, along with modules for weight estimation and geometry parameterization.

- Valliyappan and Simpson [245] solved a conceptual design optimization for a general aviation aircraft product family of small propeller driven GAA (General Aviation Aircraft) to be scaled around the 2, 4, and 6 seats configurations, and which can cruise from 150 to 300 knots and have a range from 800 to 1000 miles. The aim of this study was to explore the design space in order to find the trade-off between platform commonality and individual product performance within the aircraft family. The MOP comprised four objective functions which were defined by means of a goal programming formulation, where the deviations of each goal from their targets were minimized. For this sake, a set of 7 goals (aspiration levels), and a set of 7 constraints were defined. The first two objectives measured the technical and economical related goals within the family, respectively. The third objective measured the total constraint violation for the whole family. Finally, objective four measured the variance index or degree of commonality in variables within the product family. Design candidates were defined with a set of 14 continuous/discrete design variables, and the evaluation of the aircraft performance was done via NASA's GASP (General Aviation Synthesis Program). The authors used the NSGA-II. A special encoding was adopted in order to contain a set of commonality controlling genes (one gene per variable), followed by a concatenation of genes defining the design variables of each product in the product family.

- Rajagopal et al. [194] investigated an Unmanned Aerial Vehicle (UAV) conceptual design. Two objectives were considered: (i) the maximization of the endurance (the time an airplane can fly given a payload and a given fuel weight) and (ii) the minimization of the wing weight. Six design variables were used, four of them being wing-geometry related parameters (aspect ratio, wing loading, taper ratio, thickness to chord ratio) and the other two being UAV's operational parameters (loiter velocity and altitude). Additionally, constraints were imposed on the performance parameters of the UAV design. These included: (1) wing weight, (2) rate of climb, (3), stall speed, and (4) maximum speed at sea level condition. NSGA-II with real-numbers encoding and the SBX crossover operator was adopted. This MOEA was coupled to Raymer's RDS software, which is based on the design methods described in [199], in order to evaluate the performance of each design candidate. The authors reported that a Pareto front was obtained with a total of 11 solutions.

- Kuhn et al. [134] developed a multidisciplinary conceptual design methodology for its application to hybrid airship design (aerostatic lift and aerodynamic lift). Two objectives were considered: (i) minimization of the total mass, and (ii) maximization of the payload. Thirteen constraints were imposed, related to stress levels in the components. A set of 18 mixed real/discrete variables were used to represent the geometry of the airship and its structural properties. The optimization tool adopted was a MOEA called GAME (Genetic Algorithm for Multicriteria Engineering) [143], which is based on Evolution Strategies (ES). The evaluation of the objective functions was done with models varying in fidelity, ranging from interpolation models to FEM models. The latter was used for the structural analysis using a FEM commercial software. A Hybrid Universal Ground Observer (HUGO) airship demonstrator was designed, with a total of 10,000 design candidates being evaluated.

- Jing and Shuo [112] presented the conceptual design of an air-breathing hypersonic cruise vehicle. Five design objectives were considered: (i) maximization of the lift-to-drag ratio, (ii) minimization of the stagnation temperature, (iii) maximization of the thrust-to-drag ratio, (iv) maximization of the airframe volume, and (v) minimization of the Radar Cross Section (RCS). Constraints were imposed on variables ranges, flow flux and Mach number at inlet conditions, trimmed angle of attack and rolling angle, and static stability and maneuverability margins as well. 21 design variables were used to define the geometry of the design candidates. The authors adopted MOGA with the following features: real numbers encoding, arithmetic crossover, Gaussian mutation, steady-state reproduction and fitness sharing. Constraint handling was done by an accurate penalty strategy. Additionally, for further improvement of the solutions, a simulated annealing algorithm[3] was adopted as a local search engine. The objectives were evaluated using simplified models with reduced computational cost. Only three globally nondominated solutions could be generated. Such solutions were further evaluated and compared against a reference design. The authors noted that these solutions were better in all the objectives than the reference design (i.e., they dominated it).

- Xiaoqing et al. [261] evaluated the multiobjective optimization of hypersonic waverider shape generation. Three objectives were considered: (i) lift-to-drag ratio, (ii) vehicle's volume, and (iii) vehicle's volumetric ratio. No information is given, concerning constraints, thus it is assumed that only side constraints on variable ranges are considered. The base section of the waverider was defined by means of analytical shape functions (i.e., fourth-order polynomials), keeping to a minimum the number of design variables. The authors explored two different techniques: (a) cone derived waverider, and (b) osculating cone derived waverider. The authors adopted the NSGA-II with an improved crowding mechanism.

- Theisinger and Braun [235] identified hypersonic entry aeroshell shapes in order to find trade-off designs with increased landed mass capabilities. Three objectives were considered: (i) drag-area, (ii) static stability and (iii) volumetric efficiency. This particular spacecraft design problem was driven by planetary entry-descent-landing performance requirements and thermal/structural limitations, which are naturally conflicting. All objectives were maximized and two constraints were imposed to

[3] Kirkpatrick et al. [123] pointed out the analogy between an "annealing" process and optimization: a system state is analogous to the solution of an optimization problem. The free energy of the system (to be minimized) corresponds to the cost of the objective function to be optimized; the slight perturbation imposed on the system to change it to another state corresponds to a movement into a neighboring position (with respect to the local search state); the cooling schedule corresponds to the control mechanism adopted by the search algorithm; and the frozen state of the system corresponds to the final solution generated by the search algorithm (using a population size of one). These analogies led to the development of the so-called *simulated annealing* algorithm.

the volumetric efficiency and on the lift-to-drag ratio. Side constraints were applied to the design variables in order to obtain designs fitting with the current launch systems. Aeroshell shape was described by a bi-parametric, cubic by quadratic, non-uniform rational B-spline 3D surface, allowing them to define the optimization problem with 20 design variables, including the aeroshell angle of attack. The authors adopted the version of the NSGA-II available in the *iSIGHT* commercial software. Additionally, the objective function evaluations were performed with the estimated flow field around the aeroshell using a physics-based simulation, namely the Newtonian impact theory. The Mars Science Laboratory Aeroshell was adopted as a reference design. The authors found several design candidates that performed better than the reference design in the three objectives under consideration.

Analysis of the use of MOEAs in conceptual design

Table 1 summarizes the application of MOEAs in conceptual design optimization problems. From this table and the previous review, it can be observed that NSGA-II is the most frequently adopted approach. The common use of Pareto-based approaches seems to corroborate the hypothesis from some authors regarding the suitability of Pareto optimality to drive the search at the preliminary stages of design [254]. It should be clear that the use of MOEAs is computationally expensive, which is the reason why analytic and/or low-order engineering models are adopted in most cases. Only in a few applications, researchers seem to rely on low-order physics-based models [29], and variable-fidelity physics-based models [134]. Nevertheless, we believe that in the near future, MOEAs will become a standard practice, as the computing power available continues to increase each year. It is also worth noting that MOEAs are flexible enough as to allow their coupling to both engineering models and low-order physics-based models without major changes. They can also be easily parallelized, since MOEAs normally have low data dependency. Finally, it is worth indicating the advantage of incorporating a subjective evaluation scheme for cases in which the search must be controlled, disallowing the generation of impractical design solutions as reported by Buonanno and Mavris [29].

An aspect that is important to emphasize is the poor scalability of Pareto-based MOEAs as we increase the number of objectives [128]. Many of the applications previously described considered a low number of conflicting objectives (two or three in most cases). Although MOEAs can still be used in high-dimensional objective spaces, it is required to use mechanisms different from the traditional Pareto-based selection [96]. This issue, however, does not

Ref	NObj	NCons	NVars	VarType	Algorithm	Operators	Physics Model	NPop	Gmax	Remarks
[182]	2	s.c.	11	Continuous	MOGA	Fitness sharing, BLX-α crossover, uniform random mutation, Best-N selection	Mean line pump flow modeling	120	30	None
[29]	7	s.c.	64	Mixed continuous/discrete	SPEA2	Hierarchical crossover operator	Multiple disciplines low order/fidelity models	N/A	N/A	Island based parallel interactive GA with subjective evaluation
[245]	4	s.c.	14	Mixed continuous/discrete	NSGA-II	SBX crossover and polynomial mutation	Low order models	20	150	Objectives defined by means of goal programming technique
[194]	2	4	6	Continuous	NSGA-II	SBX crossover and polynomial mutation	Multiple disciplines, low order and database models	N/A	N/A	None
[134]	2	s.c.	18	Mixed continuous/discrete	GAME	Evolution strategies' mutation operator	Multiple disciplines with low fidelity and FEM models	400	25	None
[112]	5	6	21	Continuous	MOGA	Arithmetic crossover, gaussian mutation, fitness sharing and steady-state reproduction	Multiple disciplines simplified models	300	300	Constraint handling using exact penalty method, and simulated annealing as a local search operator
[261]	3	s.c.	5	Continuous	NSGA-II	SBX crossover, polynomial mutation and improved crowding mechanism	Inviscid flow model	N/A	N/A	None
[235]	3	2	20	Continuous	NSGA-II	SBX crossover and polynomial mutation	Newton impact theory	N/A	N/A	None

NObj = Number of objectives; NCons = Number of constraints; NVars = Number of design variables; VarType = Type of variables; NPop = Population size; Gmax = Maximum number of generations; N/A = Not available; s.c. = Only side constraints are adopted.

Table 1: Summary of MOEAs applied to conceptual design optimization problems.

seem to be a major concern in most of the applications reviewed above. A remarkable exception is the work reported in [29] in which the authors deal with a problem having seven objectives. The authors adopt in this case a parallel MOEA based on the concepts of co-evolution of multiple populations. This approach seems to produce acceptable results in this high-dimensional search space. Another issue that seems to be a common concern in this first group of applications is the encoding of the decision variables. Since this sort of application normally has mixed decision variables (e.g., discrete and continuous), authors tend to propose their own *ad-hoc* encodings, which also require specialized crossover and mutation operators associated to them. It should also be evident that in this first type of applications, authors paid little or no attention to the fine-tuning of parameters of their MOEAs. This may be due to the obvious difficulties to perform a careful statistical analysis when dealing with very expensive objective functions. However, other possible alternatives such as self-adaptation or on-line adaptation have not been properly addressed by researchers in this area yet [239]. If such self-adaptation and on-line adaptation mechanisms are unaffordable, at least the use of relatively high mutation rates is suggested, combined with a plus selection mechanism that combines the population of parents with the population of offspring and retains the best half. This will increase the selection pressure but will maintain enough diversity as to avoid premature convergence. Finally, it is worth mentioning the use of external files (or archives) as a viable alternative to reduce objective function evaluations and perform a more accurate search. This sort of mechanism can be particularly useful when combined with relaxed forms of Pareto dominance such as ϵ-dominance [144], which allows to regulate convergence, and has not been adopted by researchers working in this first group of applications.

4.2.2 *Use of MOEAs for 2D geometries and airfoil shape optimization*

Aeronautic and aerospace systems are, in general, complex engineering systems. Their analysis and design is a very complex task. There exist, however, many engineering design cases where this complexity can be tackled by analyzing basic components of the complete system, on which reduced/simplified models can be used as the basis for analyzing the whole system. Examples of these conditions are the design of 3D complex shapes such as wings and turbine blades, where the analysis of their 2D building sections (airfoils) is frequently performed prior to the analysis of the complete 3D geometry. In other cases, the geometry for the system can be such that its operating conditions can be estimated by analyzing its sectional properties. Examples of this latter condition are the aircraft engine inlets/nozzles, where the flow can be assumed as two-dimensional or axisymmetrical. In this section, some applications of MOEAs for these types of problems are presented.

- Yamaguchi and Arima [263] dealt with the optimization of a transonic compressor stator blade in which three objectives were minimized: (i) pressure loss coefficient, (ii) deviation outflow angle, and (iii) incidence toughness. The last objective function can be considered as a robust condition for the design, since it is computed as the sum of the pressure loss coefficients at two off-design incidence angles. The airfoil blade geometry was defined by twelve design variables. The authors adopted MOGA with real-numbers encoding, fitness sharing and intermediate crossover. Aerodynamic performance evaluation for the compressor blade was done using Navier-Stokes CFD simulations. The optimization process was parallelized, using 24 processors in order to reduce the computational time required. In order to promote diversity, during the first few generations, parents were selected from individuals with the first two lowest rank values (i.e., dominated individuals were also selected) and later on, only nondominated individuals were selected.

- Benini and Toffolo [16] addressed the development of high-performance airfoils for its application in axial flow compressors. They minimized two objectives: (i) nondimensional pressure ratio, and (ii) the pressure loss coefficient reduced from the unit value. Constraints were imposed on the design conditions, and were evaluated at 5 different flow-field points, in order to obtain airfoils being at least equal in performance to the reference airfoils adopted by the authors. The airfoil geometry was defined using three Bézier curves. In total 9 designs variables were used to define the airfoil geometry, its length, pitch, and incidence. A special procedure was used to avoid generating either useless or invalid airfoil geometries. The MOEA used by the authors is based on an elitist $(\mu + \mu)$ evolution strategy, which adopted binary encoding. In their implementation, μ offspring were generated using crossover and were mutated with a random-based mechanism. Repeated solutions (clones) were replaced by randomly-generated individuals. In the selection process, the combined population of parents and offspring were Pareto-ranked but considering also a diversity metric defined as a function of the minimal normalized Euclidean distance (in decision variable space) of each individual to its closest neighbor. The best μ individuals were retained as members of the following generation. The evaluation of the objective functions was done by means of CFD simulations with a high computational cost. The nondominated solutions generated by the authors were found to be superior in performance to the reference airfoils, using NACA 65 family airfoils.

- Naujoks et al. [165] addressed an airfoil design problem in which extreme Pareto optimal solutions were defined for two operational design points (two competing objectives): one for high lift performance at low speed condition and the other one for low drag performance at high speed condition. The airfoil was represented by two Bézier curves, and a total of 12 design variables were adopted. No constraints were defined, other than side constraints (upper and lower limits for the design variables). The authors used an approach called MODES (Multi Objective De-randomized Evolution Strategy). In this case, a (1+10)-DES (Derandomized Evolution Strategy) was adopted, which means that only one parent was used to produce the offspring. The aerodynamic evaluation of the design candidates is performed using a CFD Navier-Stokes simulation with a high computational cost. It is worth noting, however, that for the examples presented by the authors, a budget of only 1000 evaluations was considered. Although this was a very small number of objective function evaluations, the authors reported the generation of good approximations of the Pareto front. In a further paper, Naujoks et al. [166] proposed to use a (20+20)-MODES strategy, along with a selection mechanism inspired on the NSGA-II. The results presented with this additional selection mechanism were very similar to those obtained before, both in terms of quality of the Pareto approximation and in terms of the spread of the nondominated solutions along the Pareto front.

- Beume et al. [17] proposed the SMS-EMOA (SMS stands for S-metric[4] selection) strategy. The approach was used to solve a multi-objective airfoil design problem. As in the previous case, Pareto extreme solutions were defined by three operational conditions for lift, drag and pitching moment coefficients. The optimization problem was to find trade-off solutions minimizing the drag values for the three flow conditions, while not losing lift and keeping the pitching moment within a 2% range from the reference design points. Additionally, geometrical constraints were

[4] The **hypervolume** (also known as the S metric or the Lebesgue Measure) of a set of solutions measures the size of the portion of objective space that is dominated by those solutions collectively. It has been proved that the maximization of this performance measure is equivalent to finding the Pareto optimal set [67], and this has also been empirically verified by some researchers [61].

included for the airfoil shape. These last constraints were treated in a direct manner, discarding all infeasible solutions, previous to a CFD simulation. Results for this application were presented and compared with those obtained by using NSGA-II, in both cases with a limited budget of 1,000 function evaluations.

- Rai [193] dealt with the robust optimal aerodynamical design of a turbine blade airfoil shape, taking into account the performance degradation due to manufacturing uncertainties. Two objectives were considered: (i) to minimize the variance of the pressure distribution over the airfoil's surface, and (ii) to maximize the probability of constraint satisfaction. Only one constraint was considered, related to the minimum thickness of the airfoil shape. The constraint-handling technique adopted was the one developed by the same author and reported in [192]. The airfoil shape parameterization consisted of eight decision variables but in the experiments presented, only two of them were used for perturbing one airfoil side (the pressure side). The author adopted a multi-objective differential evolution (MODE) approach [189]. Its main features included a mechanism to reduce the set of nondominated solutions in case its size exceeded a certain (pre-defined) threshold. This was done to promote diversity in the population. It also adopted an intermediate population whose size was twice as large as the original and which was Pareto ranked so that only the first half was retained for the next generation. The author used a high-fidelity CFD simulation on a perturbed airfoil geometry in order to evaluate the aerodynamic characteristics of the airfoil generated by MODE. The simulation follows a probability density function that is observed for manufacturing tolerances. This process required a high computational cost, which the author attempted to reduce by using an artificial neural network [216] Response Surface Model.

- Ray and Tsai [197] considered an airfoil shape design optimization problem with two objectives to be minimized: (i) the ratio of the drag-to-lift squared coefficients, and (ii) the squared moment coefficient. Constraints were imposed on the flow Mach number and angle of attack. Airfoil shapes were defined by the PARSEC representation [223]. This airfoil representation allowed to define the geometry of an airfoil with 11 design variables which are more related to its aerodynamic performance than in other type of airfoil representations. The optimizer used is a multi-objective particle swarm optimizer (MOPSO) [4]. A particular feature of this application was that the particle swarm scheme was based on movements for the particles of one position to another in the design space, rather than on an update of an individual's velocity as done in the standard particle swarm optimization algorithm. The aim of this scheme was a reduction in the number of user-defined inputs. The flow solver utilized corresponds to an Euler code which was able to capture nonlinearities in the flow such as shock waves. In their results, the authors obtained a set with 32 nondominated solutions. In a related work, Ray and Tsai [198] presented a parallel implementation of this MOPSO for airfoil shape optimization. This approach was also hybridized with a gradient-based algorithm. Contrary to standard hybridization schemes where gradient-based algorithms are used to improve the nondominated solutions obtained (i.e., as a local search engine), in this approach the authors used the gradient information to repair solutions not satisfying the equality constraints. This repairing algorithm was based on the Marquardt-Levenberg algorithm [156, 147]. During the repairing process, a subset of the design variables was used, instead of the whole set, in order to reduce the dimensionality of the optimization problem to be solved.

- Obayashi et al. [173] studied the aerodynamic design of cascade airfoils shapes. The problem considered three objective functions: (i) pressure rise, (ii) flow turning angle, and (iii) total pressure loss. The first two objectives were maximized and the third one was minimized. The authors used a real-coded MOGA. Objective evaluation was performed using a 2D Navier-Stokes code for flow evaluation. The same MOEA was also used for the design of a four-stage compressor [181, 173]. In this second application, two objective functions were maximized: (i) total pressure ratio and (ii) isentropic efficiency. The MOP consisted of 80 design variables, and one constraint on the flow conditions, in order to avoid designs with flow separation. The evaluation was done using flow simulations based on the streamline curvature method in which solutions are obtained iteratively, causing a high computational cost even when an engineering model is used. The nondominated solutions obtained by the authors outperformed a baseline design in both objective functions by an amount of 1%.

- D'Angelo and Minisci [46] solved a subsonic airfoil shape optimization problem, in which two objective functions were minimized: (i) drag force coefficient, and (ii) lift force coefficient difference with respect to a reference value. The airfoil geometry was parameterized using Bézier curves both for its camber line and for its thickness distribution. Five design variables were used and constraints were imposed on the extreme values of the objective functions. The authors adopted MOPED (Multi-Objective Parzen-based Estimation of Distribution) [45], which uses the Parzen method to build a probabilistic representation of the nondominated solutions, with multivariate dependencies among the decision variables. The authors included three modifications to improve MOPED: (a) the use of a Kriging model by which solutions were evaluated without resorting to costly computational simulations, (b) the use of evolution control to keep the evolution from converging to false Pareto fronts, and (c) the hybridization of the algorithm with some mechanisms from NSGA-II (selection and ranking of solutions). Aerodynamic evaluations were performed by using a CFD simulation code, tailored for aerodynamic airfoil analysis. The authors indicated that this subsonic airfoil shape optimization problem presented difficulties associated to more complex problems: The true Pareto front was discontinuous and partially converged solutions (when divergence was detected, the iterative process was stopped) from the aerodynamic simulation code introduced irregularities in objective function space. The approximation model reduced the number of objective function evaluations in a significant manner (to one sixth of their original value).

- Bing et al. [19] presented the aerodynamic shape optimization for a 2D Hypersonic inlet and 2D SERN (Single-Expansion-Ramp Nozzle) used in scramjet engines. Two applications were presented, one with two objectives and the other with three objectives. For the first optimization example a 2D Hypersonic engine inlet was considered, and the aim was to maximize the two following objectives: (i) pressure recovery, and (ii) static pressure rise. Constraints on the design variables, inlet geometry and flow condition at exit, were imposed. The inlet geometry was defined using four decision variables. The evaluation of the design performance required high fidelity CFD Navier-Stokes simulations since the flow physics was highly nonlinear for the operating flow conditions indicated. The results of both the NSGA-II and the Neighborhood Cultivation Genetic Algorithm (NCGA) [255] were compared. The second problem considered the same inlet design previously defined, with the additional objective of minimizing the inlet drag coefficient. From the results presented by the authors, in both cases, the NCGA algorithm performed better than NSGA-II, obtaining more nondominated solutions with a better spread along the Pareto front.

- Brown et al. [27] addressed the optimization design of a scramjet inlet considering two objectives: (i) total pressure recovery factor, and (ii) variation of pressure recovery factor for a ± 5% change in free stream Mach number. The first objective was maximized, while the second was minimized. According to the design problem, geometric constraints were defined in order to remove physically unrealistic solutions. Additionally, operational flow constraints were considered to guarantee the auto-ignition in the engine. This condition required a certain range for pressure, temperature and Mach number in the flow at specific locations. The inlet was considered as a 2-D geometry and consisted of three flat ramps and a cowl at the combustion chamber inlet. In this case, 12 design variables were adopted. The MOEA adopted used a selective breeding process that ranked solutions according to the constraints, and also on the basis of the desirability of the values of the objectives (according to the user's preferences). The objective functions consisted of hypersonic flow conditions in which strong shock waves were present. The authors did not report the cardinality of the set of nondominated solutions that they obtained, but they reported the generation of a considerably high number of nondominated solutions.

- Congedo et al. [43] dealt with the airfoil shape optimization for transonic flows of Bethe-Zel'dovich-Thompson (BZT) fluids. In this case, two design conditions were explored, both for a non-lifting airfoil, and for a lifting airfoil. In the second case, the MOP considered two design objectives: (i) maximization of lift at BZT subcritical conditions, and (ii) minimization of wave drag while maximizing lift for supercritical BZT flow conditions. The geometry of the airfoil shape was represented with a Bézier curve with 16 2D control points, i.e., 32 decision variables, from which 10 are constants used to control the leading edge and trailing edge positions as well as the leading edge slope. Thus, the problem consisted of 22 variables. The only constraint included was the thickness to chord ratio of the airfoil, which was adjusted to its specified value, once a design was generated, and prior to the flow solution. The authors used the NSGA with a sigma-share formula given in [191], which takes into account the population size and the number of objectives. They chose parameters in such a way that less than 1,000 objective function evaluations were performed. The authors reported that all the solutions that they obtained outperformed the baseline design as well as the designs obtained using traditional design methods.

- Shimoyama et al. [222] developed a novel optimization approach for robust design. In their approach, a design for multi-objective six-sigma (DFMOSS) [221] was applied for the robust aerodynamic airfoil design of a Mars exploratory airplane. The core of the design methodology was, on the one hand, the concept of *Robust Design*[5] and, on the other, its multi-objective nature. The idea of the DFMOSS methodology was to incorporate a MOEA to simultaneously optimize the mean value of an objective function, while minimizing its standard deviation due to the uncertainties indicated above. The airfoil shape optimization problems considered two cases: a robust design of (a) airfoil aerodynamic efficiency (lift-to-drag ratio), and (b) airfoil pitching moment constraint. In both cases, only the variability in the flow Mach number was taken into account. The authors adopted MOGA. The airfoil geometry was defined using Bézier curves both for the upper and for the lower surfaces. 6 control points were used, resulting in 12 design variables. The aerodynamic performance of the airfoil was evaluated by CFD simulations using the Favre-Averaged compressible thin-layer Navier-Stokes equations. Eighteen robust nondominated solutions were obtained in the first test case. From this set, almost half of the population attained the 6σ condition. In the second test case, more robust nondominated solutions were found, and they satisfied a sigma level as high as 25σ.

- Szöllös et al. [229] addressed the aerodynamic shape optimization of the airfoil geometry of a standard-class glider, considering three objectives: (i) maximize gliding ratio at high flight speed, (ii) maximize gliding ratio at average weather conditions, and (iii) minimize sink rate at low turning speeds. All these objectives are specified in terms of airfoil's aerodynamic lift and drag coefficients as well as flight operating conditions in terms of the Reynolds number (Re) and the Mach number (M). Constraints are considered for: (a) airfoil's maximal lift coefficient at landing flight conditions, (b) maximum airfoil's thickness to chord ratio, (c) trailing edge thickness, and (d) pitching moment coefficient (C_m) which is required not to be worse than a reference airfoil design. The authors introduced a new MOEA called *multi-objective micro-genetic algorithm with range adaptation, based on* ϵ*-dominance* or $\epsilon\mu$ARMOGA. This approach is inspired on the Adaptive Range Multi-Objective Genetic Algorithm (ARMOGA) [208]. ARMOGA incorporates two archiving tech-

[5] Robust design deals with the idea of designing and/or developing a product that has minimal variance in its characteristics and meets the exact performance desired.

niques: a global archive, which stores all the best solutions obtained so far, and a recent archive, which stores the best solutions of the past previous generations. Solutions from the second archive participate in the parent selection process. ϵμARMOGA introduces two additional mechanisms. The first corresponds to the use of a small population size (i.e. the use of a micro-genetic algorithm as in [131, 40]), coupled with the use of an external file for storing the nondominated solutions obtained so far. The second mechanism corresponds to the use of the concept of ϵ-dominance [145], which is a relaxed form of Pareto dominance that has been used as an archiving strategy that allows to regulate convergence. The authors initialized the population using a Latin Hypercube Sampling (LHS) technique, and the main population was reinitialized at every certain number of generations, based on the average and standard deviation of the decision variables. The objective functions were evaluated using a CFD simulation code. The authors obtained feasible solutions with improvements on the order of 10%, 8% and 7-10% for the first, second and third objectives, respectively, with respect to a reference airfoil design.

Analysis of the use of MOEAs in 2D geometries and airfoil shape optimization

Table 2 summarizes the application of MOEAs in 2D geometries and airfoil shape optimization problems. From this table and the previous discussion, we can see that, as before, a wide variety of Pareto-based elitist MOEAs have been used in this domain. It is also worth noting the use of MOEAs in *robust design*, in which solutions are evaluated with off-design operating conditions and manufacturing tolerances. Such solutions are thus representing more realistic designs. Several authors report improved designs when adopting MOEAs, but unsuccessful cases have also been reported. The cases in which MOEAs fail to produce improved designs seem to be associated to situations in which the baseline design had been already improved in a significant manner, or when the search space is so highly constrained that it is difficult to move to better regions. Again, the high computational cost associated to the use of MOEAs is evident. In spite of the advantages of Pareto-based MOEAs, it is also evident that, when dealing with expensive objective functions such as those of the above applications, the use of careful statistical analysis of parameters is unaffordable. Thus, the parameters of the MOEAs discussed in this section were simple guesses or taken from values suggested by other researchers. It is also important to note that some researchers have suggested clever approaches that allow the use of very small population sizes, although surrogate models have also been

Ref	NObj	NCons	NVars	VarType	Algorithm	Operators	Physics Model	NPop	Gmax	Remarks
[263]	3	s.c.	12	Continuous	MOGA	Intermediate crossover and fitness sharing	Navier-Stokes	100	30	Robust design optimization
[16]	2	5	9	Discrete	$(\mu + \mu)$-ES	Gaussian mutation, Goldberg's Pareto ranking, crowding based on euclidian distance in decision space	Navier-Stokes	100	200	None
[165]	2	s.c.	12	Continuous	(1+10)-MODES	Adaptive derandomized mutation strategy, selection based on the NSGA-II	Navier-Stokes	1	N/A	Use of a maximum of 1,000 designs
[17]	3	2	12	Continuous	SMS-EMOA	Adaptive derandomized mutation strategy, steady-state selection based on hypervolume measure	Navier-Stokes	20	N/A	Use of a maximum of 1,000 designs
[193]	2	1	8	Continuous	MODE	DE's crossover and mutation operators	Navier-Stokes	10	25	Robust design optimization, use of ANN RSM
[197]	2	2	11	Continuous	MOPSO	N/A	Euler model	100	50	None
[173]	3	s.c.	N/A	Continuous	MOGA	N/A	Navier-Stokes	64	75	None
[181]	2	1	80	Continuous	MOGA	N/A	Streamline curvature method	300	1000	None
[46]	2	s.c.	5	Continuous	MOPED	N/A	Coupled boundary layer potential flow panel method	N/A	N/A	Use of Kriging model
[19]	3	2	4	Continuous	NCGA	N/A	Parabolized Navier-Stokes	100	50	None
[27]	2	N/A	12	Continuous	N/A	Elitist selective inter-breeding, ranking of solutions according to constraints and user defined preferences, weighted variable recombination	Navier-Stokes	100	100	None
[43]	2	1	22	Continuous	NSGA-II	SBX crossover and polynomial mutation	Euler flow with thermodynamical model for dense gases	36	24	None
[222]	2	s.c.	12	Continuous	MOGA	Stochastic universal sampling, blended crossover, uniform mutation, best-N selection	Favre-Averaged compressible thin-layer Navier-Stokes	64	100	Robust design optimization based on 6σ
[229]	3	4	12	Continuous	$\epsilon\mu$ARMOGA	SBX crossover, no mutation is used, external file storage based on ϵ-dominance	Coupled boundary layer potential flow panel method	4	2000	Reinitialization of population is used for diversity preserving, instead of mutation

NObj = Number of objectives; NCons = Number of constraints; NVars = Number of design variables; VarType = Type of variables; NPop = Population size; Gmax = Maximum number of generations; N/A = Not available; s.c. = Only side constraints are adopted.

Table 2: Summary of MOEAs applied in 2D geometries and airfoil shape optimization problems

employed, as in the previous section. Nevertheless, the use of other simpler techniques such as fitness inheritance or fitness approximation [200] seems to be uncommon in this domain and could be a good alternative when dealing with high-dimensional problems. Additionally, the authors of this group of applications have relied on very simple constraint-handling techniques, most of which discard infeasible individuals. Alternative approaches exist, which can exploit information from infeasible solutions and can make a more sophisticated exploration of the search space when dealing with constrained problems (see for example [159]) and this has not been properly studied yet. Finally, it is worth emphasizing that, in spite of the difficulty of these problems and of the evident limitations of MOEAs to deal with them, most authors report finding improved designs when using MOEAs, even when in all cases a fairly small number of fitness function evaluations was allowed. This clearly illustrates the high potential of MOEAs in this domain.

4.2.3 *Use of MOEAs for 3D complex physics/shape optimization*

Sophisticated aeronautical/aerospace systems possess in most cases, complex three-dimensional shapes and/or are designed to operate in complex physical environments. Examples of such complex three-dimensional shapes are those of turbine/propeller blades, and complete aircraft configurations. Complex three-dimensional physics are present for high speed flow over wings and turbine/propeller blades, in which shock waves can arise, affecting the design performance. For these cases, the MOP cannot be simplified by the use of reduced models, such as two-dimensional simulations, as done in the applications of the previous section. Next, we will discuss applications of MOEAs in which their authors deal with these 3D complex physics/shape optimization problems.

- Sasaki et al. [209] and Obayashi et al. [172] solved a multi-objective aerodynamic wing shape optimization problem in which they minimized three objectives: (i) drag coefficient for transonic cruise, (ii) drag coefficient for supersonic cruise, and (iii) bending moment at the wing root for supersonic cruise condition. The set of constraints comprised lift coefficient at both transonic and supersonic cruise conditions, wing area and maximum airfoil thickness. The variables for this design were 66 in total, and defined the wing planform shape, airfoil chord and thickness distribution at several wing stations, as well as wing twist angles at the same airfoil locations. The authors adopted MOGA and the design candidates were evaluated by a high-fidelity Navier-Stokes CFD flow simulation. The evaluation process was parallelized using the master-slave paradigm. In a further paper, Sasaki et al. [210] used the same algorithm for the aerodynamic optimization of a supersonic transport wing-body configuration. In this application, two objectives were considered: (i) drag coefficient and (ii) difference in Darden's equivalent area distribution. Constraints on the lift coefficient were imposed during the optimization, and on the length and volume of the fuselage. The aim of the second objective was to achieve low sonic boom characteristics. For this problem, the number of variables increased to 131, as the fuselage geometry was added in this case. The aerodynamic evaluation for the first objective was performed by an Euler CFD simulation to considerably reduce the computational time with respect to the use of a Navier-Stokes CFD simulation. Nonetheless, the optimization process was parallelized using the master-slave paradigm. Two test cases were considered, each one having different upper/lower limits for the section nearby the wing-body intersection.

- Sasaki and Obayashi [212] solved a problem similar to the previous one [210] and obtained analogous results, but in this case, the ARMOGA algorithm was used. Also, and in order to incorporate constraints, an extended Pareto ranking method based on constraint-dominance was used [69].

- Ng et al. [168] addressed a multiobjective wing platform and airfoil shape optimization problem. The MOP aimed to redesign the reference ONERA M6 wing minimizing two objectives: (i) W/Wo, which is the ratio for the design wing weight with respect to the reference ONERA M6 wing weight, and (ii) CD/CDo, which is the ratio of the design wing drag coefficient with respect to that of the reference wing. The first objective was evaluated using a semi-empirical equation, while the second was obtained from a multigrid Euler CFD simulation. Constraints were imposed on the flow Mach number and constant lift coefficient. No special constraint handling technique was used, but the CFD code was instructed to vary the angle of attack, subjected to a tolerance, in order to satisfy this equality constraint. This technique can be seen as a mechanism to repair solutions. The wing platform was represented by 5 design variables: (a) taper ratio, (b) wing sweep angle, (c) twist angle, (d) aspect ratio, and (e) thickness-to-chord ratio. The airfoil used for the wing corresponded to the symmetric airfoil used in the ONERA M6 wing, and was the same across the wing. The optimizer used was based on the PSO algorithm described in Ray et al. [197]. The authors presented results for two test cases: the first with 4 steps and the second with 8 steps. In the first case 10 nondominated solutions were obtained, while 11 were found in the second case. In both cases, all the nondominated designs were better in the first objective function compared to the reference wing, and for the second objective, almost half of the population were better while the rest were worse, with respect to the reference wing. An Adaptive Search Space Operator (ASSO) technique was used by the authors to give the algorithm the possibility of adapting decision variables bounds by shrinking/expanding the boundaries of the design space.

- Lian and Liou [150] addressed the optimization of a three-dimensional rotor blade, namely the redesign of the NASA rotor 67 compressor blade, a transonic axial-flow fan rotor, which was the first of a two-stage compressor fan. Two objectives were considered in this case: (i) maximization of the stage pressure rise, and (ii) minimization of the entropy generation. Constraints were imposed on the mass flow rate to have a difference less than 0.1% between the new one and the reference design. The blade geometry was constructed from airfoil shapes defined at four span stations, with a total of 32 design variables. The authors adopted MOGA. The optimization process was coupled to a second-order RSM, which was built with 1,024 design candidates using the Improved Hypercube Sampling (IHS) algorithm. 12 design solutions were selected from the RSM-Pareto front obtained, and such solutions were verified with a high fidelity CFD simulation. The objective function values slightly differed from those obtained by the approximation model, but all the selected solutions were better in both objective functions than the reference design. Similar work was presented by Lian and Liou [151] but minimizing the blade weight instead of the entropy generation. Similar performance results were obtained with lighter blades. More recently, Kim and Liou [122] presented the design of three new MOEAs, including additional mechanisms to the basic MOGA algorithm indicated before. Such mechanisms included: an elite-preserving approach (EP-MOGA), a modified sharing function (EP-MOGAS), and a gradient-based directional operator (EP-MOGAS-D).

- Holst [86] presented the aerodynamic optimization of a wing-body configuration in which two objective functions were maximized: (i) lift-to-drag ratio, and (ii) configuration volume. Constraints were imposed on the operating flow condition at transonic Mach number and at a fixed lift. The problem had 66 decision variables which controlled the wing geometry, its position along the fuselage and the section shape of the fuselage at some specified fuselage stations. The author adopted MOGA. The proposed approach was able to reduce the fuselage cross section in the vicinity of the wing-fuselage juncture, which is a common practice in aerodynamic design for the transonic flow regime.

- Sasaki et al. [211] solved an aerodynamic MOP for a turbine compressor stage. The main aim was to improve three aerodynamic objectives, by identifying the trade-offs among them in the baseline condition: (i) isentropic efficiency, (ii) blockage, and (iii) flow loss. Equality constraints on the design were imposed, intended mainly to maintain the flow and operating conditions similar to those of the baseline geometry: Stage loading, mass flow rate, stage exit whirl angle and pressure ratio. Such equality constraints were transformed into inequalities, and thresholds were reduced as the optimization proceeded. The three-dimensional shape of the blade was re-designed from the baseline geometry, by defining parameters that allowed: (a) axial movement of sections along the engine axis, (b) circumferential movement of sections, (c) solid body rotation of sections based on trailing edge position, and (d) control on the number of blades. In total, 28 design variables were used per compressor stage. The authors adopted ARMOGA. The aerodynamic evaluation was performed with high fidelity Reynolds-Averaged Navier-Stokes CFD tools to analyze a compressor stage. The CFD analysis comprised the rotor/stator interaction. The authors presented two application examples, the first of which had a fixed number of rotor/stator blades. The optimization process was able to improve the baseline design while 8 designs satisfied all the constraints. Efficiency was improved within 1%, even when infeasible solutions were considered. After analyzing the trade-off among the objectives from the first test case, a second test case was proposed, considering the number of rotor/stator blades as an additional variable, and changing the approximation function in the radial direction. In this case, a B-spline function was used instead of the cubic-spline adopted in the previous case. Results from this second test case achieved an efficiency improvement of 1.5%. In this case, 14 feasible designs were generated, from which only 4 were nondominated.

- Benini [15] extended a previous work from Benini and Toffolo [16] for a three-dimensional transonic compressor rotor design optimization problem in which two objective functions were maximized: (i) total pressure ratio, and (ii) adiabatic efficiency. Constraints were imposed on the design conditions as to obtain the mass flow of a reference design, the NASA Rotor 37. The blade geometry used in the transonic compressor rotor was parameterized by Bézier curves defining the mean camber line and the thickness distribution. Three profiles along the blade span were defined: at hub, midspan and tip. A total of 23 decision variables defined the 3D compressor rotor geometry. The author used the MOEA described in [16], which is based on evolution strategies. The performance evaluation of the designs was done using high fidelity Navier-Stokes CFD simulations. The authors noted that the nondominated solutions produced were clustered around the reference design point, due to a tight constraint imposed on the flow mass rate, which did not allow the algorithm to explore a wider region of the search space. Nevertheless, the author was able to obtain improvements in both objective functions using the proposed approach.

- Chiba et al. [33] explored the trade-offs among four aerodynamic objective functions in the optimization of a wing shape for a Reusable Launch Vehicle (RLV). The objective functions were: (i) the shift of the aerodynamic center between supersonic and transonic flight conditions, (ii) pitching moment in the transonic flight condition, (iii) drag in the transonic flight condition, and (iv) lift for the subsonic flight condition. The first three objectives were minimized while the fourth was maximized. These objectives were selected for attaining control, stability, range and take-off constraints, respectively. The RLV definition comprised 71 design variables to define the wing platform, wing position along the fuselage and airfoil shape at prescribed wingspan stations. The authors adopted ARMOGA, and the aerodynamic evaluation of the RLV was done with a Reynolds-Averaged Navier-Stokes CFD simulation. A trade-off analysis was conducted with 102 nondominated individuals generated by the MOEA.

- Song and Keane [224] performed the shape optimization of a civil aircraft engine nacelle. The primary goal of the study was to identify the trade-off between aerodynamic performance and noise effects associated with various geometric features for the nacelle. For this, two objective functions were defined: i) scarf angle, and ii) total pressure recovery. The nacelle geometry was modeled using 40 parameters, from which 33 were considered design variables. The authors adopted the NSGA-II with a commercial CFD software for evaluating the three-dimensional flow characteristics. Due to the large size of the design space to be explored, as well as the simulations being time consuming, a Kriging-based surrogate model was adopted in order to keep the number of designs being evaluated with the CFD tool to a minimum. The authors reported difficulties in obtaining a reliable Pareto front (there were large discrepancies between two consecutive Pareto front approximations). They attributed this behavior to the large number of variables in the design problem, and also to the associated difficulties to obtain an accurate Kriging model for these situations. In order to alleviate this situation, they performed an analysis of variance (ANOVA) test to find the variables that contributed the most to the objective functions. After this test, they presented results with a reduced surrogate model, employing only 7 decision variables. The authors argued that they obtained a design similar to a reference one, but requiring a lower computational cost because of the use of this reduced Kriging model.

- Jeong et al. [107] investigated the improvement of the lateral dynamic characteristics of a lifting-body type re-entry vehicle in transonic flight condition. Two objectives were minimized: (i) the derivative of the yawing moment, and (ii) the derivative of the rolling moment. The MOP involved four design variables, and two solutions were sought: The first one without constraints, and the second one constraining the lift-to-drag ratio for the lifting-body type re-entry vehicle. The authors adopted the Efficient Global Optimization for Multi-Objective Problems (EGOMOP) algorithm developed by Jeong et al. [104]. This algorithm was built upon the ideas of the EGO and ParEGO Algorithms from Jone et al. [113] and Knowles et al. [125], respectively. For the exploration of the nondominated solutions, the authors adopted MOGA. Due to the geometry of the lifting body and the operating flow condition of interest, namely high Mach number and strong vortex formation, the evaluation of the objectives was done by means of a full Navier-Stokes solver. Since the objectives were actually derivatives, multiple flow solutions were required to determine their values in a discrete manner, considerably increasing the total computational time due to a large number of calls of the CFD code. The authors were able to find better geometry configurations than the baseline one, with better lateral dynamic characteristics, both for the unconstrained and for the constrained instances.

- Lee et al. [146] presented the robust design optimization of an ONERA M6 wing shape. The robust optimization was based on the concept of the Taguchi method in which the optimization problem is solved considering uncertainties in the design environment, in this case, the flow Mach number. The problem had two objectives: (i) minimization of the mean value of an objective function with respect to variability of the operating conditions, and (ii) minimization of the variance of the objective function of each candidate solution, with respect to its mean value. In the sample problems, the wing was defined by means of its planform shape (sweep angle, aspect ratio, taper ratio, etc.) and of the airfoil geometry, at three wing locations (each airfoil shape was defined with a combination of mean lines and camber distributions), using a total of 80 design variables to define the wing designs. Geometry constraints were defined by upper and lower limits of the design variables. The authors adopted the Hierarchical Asynchronous Parallel Multi-Objective Evolutionary Algorithm (HAPMOEA) [77], which is based on evolution strategies, incorporating the concept of Covariance Matrix Adaptation (CMA). The aerodynamic evaluation was done with a CFD simulation. It is worth noting that HAPMOEA uses, during the evolutionary process, a hierarchical set of CFD models, varying the grid resolution of the solver (three levels are used), as well as different population sizes (depending on the grid resolution). The authors presented two solutions, with and without uncertainties. In the latter case the problem considered two design points (at two different operating conditions), and the algorithm found the trade-off solutions between these two design points. For the case of the design with uncertainties, the optimization problems found the trade-off solutions considering the minimization for the mean value of the objective function (the inverse of the lift-to-drag ratio for the wing) and its variance with respect to the mean value. From the results presented by the authors, the Pareto fronts were continuous and exhibited a concave geometry for the trade-off solutions. 12 solutions were obtained in the robust design of the wing and all the nondominated solutions presented a shock-free flow both at the upper and at the lower surface of the wing. Additionally, the nondominated solutions showed a better behavior, in terms of aerodynamic performance (lift-to-drag ratio) with a varying Mach number, as compared to the baseline design. In these examples, the authors used three grid-levels (model resolution): fine, intermediate, and coarse. During the evolutionary process, the individuals were moved from the coarse to the fine levels and viceversa. A total of 1100 individuals were evaluated.

- Oyama et al. [183] applied a design exploration technique to extract knowledge information from a flapping wing MAV (Micro Air Vehicle). The flapping motion of the MAV was analyzed using multi-objective design optimization techniques in order to obtain nondominated solutions which were analyzed with Self Organizing Maps (SOMs) in order to extract knowledge about the effects of the flapping motion parameters on the objective functions. The conflicting objectives considered were: (i) maximization of the time-averaged lift coefficient, (ii) maximization of the time-averaged thrust coefficient, and (iii) minimization of the time-averaged required power coefficient. The problem had five design variables and the geometry of the flying wing was kept fixed. Constraints were imposed on the averaged lift and thrust coefficients so that they were positive. The authors adopted MOGA. Due to the nature of the complex flow in this problem, the objective functions were obtained by means of CFD simulations, solving the unsteady incompressible Navier-Stokes equations. Objective functions were averaged over one flapping cycle. The purpose of the study was to extract trade-off information from the objective functions and the flapping motion parameters such as plunge amplitude and frequency, pitching angle amplitude and offset, and phase difference. In order to minimize the turnaround computational time, the evaluation of the objective functions was parallelized using a cluster of workstations. From the results obtained, the authors extracted extreme nondominated solutions which were further analyzed to understand their flow physics for each objective in particular.

- Arabnia and Ghaly [8] presented the aerodynamic shape optimization of turbine stages in three-dimensional fluid flow, so as to minimize the adverse effects of three-dimensional flow features on the turbine performance. Two objectives were considered: (i) maximization of isentropic efficiency for the stage, and (ii) minimization of the streamwise vorticity. Additionally, constraints were imposed on: (1) inlet total pressure and temperature, (2) exit pressure, (3) axial chord and spacing, (4) inlet and exit flow angles, and (5) mass flow rate. The blade geometry, both for rotor and stator blades, was based on the E/TU-3 turbine which is used as a reference design to compare the optimization results. The multi-objective optimization consisted of finding the best distribution of 2D blade sections in the radial and circumferential directions. For this, a quadratic rational Bézier curve, with 5 control points was used for each of the two blades. The authors adopted NSGA. Both objective functions were evaluated by using a 3D CFD flow simulation. The authors adopted an artificial neural network (ANN) based RSM. The ANN model with backpropagation, contained a single hidden layer with 50 nodes, and was trained and tested with 23 CFD simulations, sampling the design space using the LHS technique. The optimization process was undertaken by using the ANN model to estimate both the objective functions, and the constraints. Finally, the nondominated solutions obtained were evaluated with the actual CFD flow simulation. The authors indicated that they were able to obtain design solutions which were better than the reference turbine design.

- Tani et al. [233] solved a rocket engine turbopump blade shape optimization design which considered three objective functions: (i) shaft power, (ii) entropy rise within the stage, and (iii) angle of attack of the next stage. The first objective was maximized while the others were minimized. The design candidates defined the turbine blade aerodynamic shape and consisted of 58 design variables. The authors adopted MOGA. The objective function values were obtained from a CFD Navier-Stokes flow simulation. The authors reported solutions that were better than a baseline design turbopump blade shape. Indeed, improvements on the three objective functions were of 8%, 30% and 40%, respectively, as compared to the baseline design.

Analysis of the use of MOEAs in 3D complex physics/shape optimization

Table 3 summarizes the application of MOEAs in 3D complex physics/shape optimization problems. For this group of applications, a common point is that 3D complex shapes and/or complex physics models are considered, which requires, in most cases, the use of high dimensional design space and/or sophisticated simulation tools. For both cases, the design optimization search becomes highly computationally expensive (some authors report times in the order of days or even months for the problems that they solved). Such applications require approaches that can minimize their high computational cost. Some authors relied on parallelization techniques for this sake (see for example [209]). An interesting parallel approach is the one reported by Lee et al. [146], in which the evaluation of the objectives is done in an asynchronous manner, with a scheme that resembles an island model [41]. Such asynchronous parallel MOEAs are uncommon in the specialized literature, in spite of their high potential in the sort of applications reported in this section. Another alternative is the use of surrogate models, which are adopted by a number of works reported in this section. For example, Lian and Liou [150, 151], used a second order RSM, Song and Keane [224] used a Kriging-based model, Lee et al. [146] adopted hierarchical CFD models (i.e., models with varying mesh sizes, which produce approximations at a reduced computational cost), and Arabnia and Ghaly [8] adopted an artificial neural network. The use of approximate models can be seen as an advantage, but also presents drawbacks, for example, for large dimensional design spaces, as indicated by Song and Keane [224]. Another alternative is to adopt simpler approximation mechanisms such as fitness inheritance [160] and fitness approximation [230]. Another aspect worth emphasizing is that most authors adopted MOEAs with real-numbers encoding, rather than with binary encoding. This is relatively common when dealing with engineering applications having a high number of decision variables. The lack of modern diversity maintenance approaches such as archiving techniques (see for example [144, 83, 218]) is also evident within the applications of this section, although there are some interesting exceptions. For example, Sasaki and Obayashi [212] adopted two external archives for their MOEA. Also interesting is the proposal of Holst [86] of using "bins" (this approach is similar in its operation to the adaptive grid adopted in PAES [129]). However, it is worth noting that both, Sasaki & Obayashi's and Holst's approaches quickly degrade their performance as the number of objectives increases.

An interesting area worth exploring is the design of mechanisms that allow a better (i.e., more intelligent) exploration of the search space. For example, Sasaki [208], and Ng et al. [168] use statistics gathered from the population in order to guide the search. Such approach, however, requires a good diversity maintenance mechanism in order to avoid an excessive selection pressure that would produce premature convergence.

In spite of the large number of constraint-handling techniques currently available for evolutionary algorithms [39, 159], in most of the works reported in this section there is a noticeable lack of them. The use of good constraint-handling techniques is particularly useful when the optimum solutions lie on the boundary between the feasible and the infeasible regions, which is normally the case in multi-objective optimization [41]. Their use can contribute to a better (i.e., more efficient and effective) exploration of the search space in the presence of constraints.

Ref	NObj	NCons	NVars	VarType	Algorithm	Operators	Physics Model	NPop	Gmax	Remarks
[209]	3	4	66	Continuous	MOGA	Fitness sharing, BLX-α crossover, best N selection	Navier-Stokes	64	30	None
[172]	3	4	66	Continuous	MOGA	Fitness sharing, averaged crossover, best N selection	Navier-Stokes	64	70	None
[210]	2	3	131	Continuous	MOGA	Fitness sharing, BLX-α crossover, best N selection	Euler/Navier-Stokes	64	20	None
[212]	2	3	131	Continuous	ARMOGA	Fitness sharing, BLX-α crossover, best N selection	Euler/Navier-Stokes	64	20	Design variables ranges are adapted every M generations, based on the statistics of the archive and current population
[168]	2	2	5	Continuous	MOPSO	Adaptive Search Spacing Operator (ASSO)	Euler	N/A	N/A	The ASSO operator allows to extend the initial design space
[150]	2	1	32	Continuous	MOGA	Fitness sharing, BLX-α crossover, best N selection, random uniform mutation	Reynolds-Averaged Navier-Stokes	N/A	N/A	Use of RSM
[86]	2	2	66	Continuous	MOGA	Masking array to activate/deactivate the design variables, selection based on bins of the nondominated archive, random average crossover, local and global mutation operators	Potential flow	34	N/A	None
[211]	3	4	28	Continuous	ARMOGA	Stochastic universal sampling, SBX crossover, polynomial mutation, best N selection, Pareto ranking incorporating constraints	Reynolds-Averaged Navier-Stokes	16	20	Grid-enabled parallel computation
[15]	2	1	23	Discrete	$(\mu + \mu)$-ES	Gaussian mutation, Goldberg's Pareto ranking, crowding based on Euclidian distance in the decision space	Navier-Stokes	20	100	None
[33]	4	s.c.	71	Continuous	ARMOGA	Fitness sharing, BLX-α crossover, best N selection	Reynolds-Averaged Navier-Stokes	8	30	None
[224]	2	s.c.	33	Continuous	NSGA-II	SBX crossover, polynomial mutation	Navier-Stokes	60	20	Use of Kriging model
[107]	2	1	4	Continuous	MOGA	N/A	Navier-Stokes	N/A	N/A	None
[146]	2	s.c.	80	Continuous	HAPMOEA	ES mutation operator with Covariance Matrix Adaptation (CMA-ES), distance dependent mutation, tournament selection	Navier-Stokes	*	N/A	* Population sizes are 20, 40 and 60 for fine, medium and coarse CFD mesh grids, 1100 design candidates evaluated
[183]	3	2	5	Continuous	MOGA	Fitness sharing, roulette wheel selection,BLX-α crossover, random uniform mutation, Pareto based constraint handling	Navier-Stokes	N/A	N/A	Knowledge extraction from the multi-objective optimization process
[8]	2	5	10	Continuous	NSGA	N/A	Navier-Stokes	N/A	N/A	ANN model
[233]	3	s.c.	58	Continuous	MOGA	Stochastic universal sampling, BLX-α crossover, best N selection	Navier-Stokes	16	50	None

NObj = Number of objectives; NCons = Number of constraints; NVars = Number of design variables; VarType = Type of variables; NPop = Population size; Gmax = Maximum number of generations; N/A = Not available; s.c. = Only side constraints are adopted.

Table 3: Summary of MOEAs applied in 3D complex physics/shape optimization problems

4.3 COMMENTS ON DIFFERENT OPTIMIZATION APPROACHES

In the previous section we have reviewed several applications of MOEAs for different aeronautical/aerospace engineering problems. This review comprises the main two design phases of *conceptual design* and *preliminary design*, as defined in the first section of this chapter. From this review we can also identify the following common optimization approaches used:

- Surrogate-based optimization,
- Hybrid MOEA optimization,
- Robust design optimization,
- Multidisciplinary design-optimization, and
- Data-mining and knowledge extraction.

4.3.1 *Comments on surrogate-based optimization*

Surrogate models are built to approximate computationally expensive functions. The main objective in constructing these models is to provide a reasonably accurate approximation to the real functions, while reducing by several orders of magnitude the computational cost. Surrogate models range form Response Surface Methods (RSM) based on low-order polynomial functions, Gaussian processes or Kriging, Radial Basis Functions (RBFs), Artificial Neural Networks (ANNs), to Support Vector Machines (SVMs). A detailed description of each of these techniques is beyond the scope of this chapter, but the interested reader is referred to Jin [109] for a comprehensive review of these and other approximation techniques.

The accuracy of the surrogate model relies on the number and on the distribution of samples provided in the search space, as well as on the selection of the appropriate model to represent the objective functions and constraints. One important fact is that Pareto-optimal solutions based on the computationally cheap surrogate model do not necessarily satisfy the real objective function evaluation. So it is necessary to verify the whole set of Pareto-optimal solutions found from the surrogate, which can render the problem very time consuming. If discrepancies are large, this condition might attenuate the benefit of using a surrogate model. The verification process is also needed in order to update the surrogate model. This latter condition raises the question of how often in the design process it is necessary to update the surrogate model. There are no general rules for this, and many researchers rely on previous experiences and trial and error guesses.

4.3.2 *Comments on hybrid MOEA optimization*

One of the major drawbacks of MOEAs is that they are very time demanding, due to the relatively high number of objective function evaluations that they typically require. This has motivated a number of approaches to improve their efficiency. One of them consists in hybridizing a MOEA with a gradient-based method. In general, gradient-based methods converge quickly for simple topologies of the objective functions but will get trapped in a local optimum if multi-modal objective functions are considered. In contrast, MOEAs can normally avoid local minima and can also cope with complex, noisy objective function topologies. The basic idea behind this hybridization is to resort to gradient-based methods, whenever the MOEA convergence is slow.

Experience has shown that hybridizing MOEAs with gradient-based techniques can, to some extent, increase their convergence rate. However, in the examples presented above, the gradient information relies on local and/or global surrogate models. For this, one major concern is how to build a high-fidelity surrogate model with the existing designs in the current population, since, their distribution in the design space can introduce some undesired bias in the surrogate model. Additionally, there are no rules for choosing the number of points for building the surrogate model, nor for defining the number of local searches to be performed. These parameters are empirically chosen.

4.3.3 *Comments on robust design optimization*

In aeronautical/aerospace engineering optimization, uncertainties in the environment must be taken into account. For example, the operating velocity of an aircraft may deviate from the normal condition during the flight. This change in velocity can be so high that it changes the Mach and/or Reynolds number for the flow. The variation of these parameters can substantially change the aerodynamic properties of the design. In this case, a robust optimal solution is desired, instead of the optimal solution found for ideal operating conditions. By robustness, it is meant in general that the performance of an optimal solution should be insensitive to small perturbations of the design variables or environmental parameters. In multiobjective optimization, the robustness of a solution can be an important factor for a decision maker in choosing the final solution. Search for robust solutions can be treated as a multiobjective task, i.e., to maximize the performance and the robustness simultaneously. These two tasks are very likely conflicting, and therefore, MOEAs can be employed to find a number of trade-off solutions.

Robust solutions can be achieved in evolutionary optimization in different ways. One simple approach is to add perturbations to the design variables or environmental parameters before the fitness is evaluated, which is known as implicit averaging [240]. An alternative to implicit averaging is explicit averaging, which means that the fitness value of a given design is averaged over a number of designs generated by adding random perturbations to the original design. One drawback of the explicit averaging method is the number of additional quality evaluations needed, which can turn the approach impractical. In order to tackle this problem, metamodeling techniques have been considered [176, 184]. A slightly different approach is to find the solution with the maximal allowed deviation given the allowed performance deterioration [153]. One potential advantage of this method is that no assumptions need to be made concerning the noise distribution (as needed in the averaging based approaches). Interested readers are referred to Jin and Branke [110] for a more detailed discussion on evolutionary search for robust solutions.

4.3.4 *Comments on multidisciplinary design optimization*

Multi-disciplinary design optimization (MDO) aims at incorporating optimization methods to solve design problems, considering not only one engineering discipline, but a set of them. This latter condition is frequently faced by aeronautical/aerospace engineering designs, where aerodynamics, structural dynamics, propulsion, acoustics, among others, are present. MDO allows designers to incorporate all relevant disciplines (through their interactions) simultaneously. The optimum of a multidisciplinary problem might be a compromise solution from the multiple disciplines involved. In this sense, multi-objective optimization is well suited for this type of problems, since it can exploit the interactions between the disciplines, and can help to find the trade-offs among them.

The increasing complexity of engineering systems has raised the interest in multidisciplinary optimization, as can be seen from the examples presented in this chapter. For this task, MOEAs facilitate the integration of several disciplines, since they do not require additional information other than the evaluation of the corresponding objective functions, which is usually done by each discipline and by the use of simulations. Additionally, an advantage of the use of MOEAs for MDO, is that they can easily manage any combination of variable types, coming from the involved disciplines i.e., from the aerodynamic discipline, the variables can be continuous, but for the structural optimization, it can happen that the variables are discrete. Kuhn et al. [134] presented an example of this condition for the multi-disciplinary design of an airship. However, one challenge in MDO is the increasing dimensionality attained in the design space, as the number of disciplines also increases.

4.3.5 *Comments on data mining and knowledge extraction*

So far we have presented some applications of MOEAs aeronautical/aerospace engineering design optimization. In the majority of the examples, the researchers' aim was to find trade-offs among the conflicting objectives. This information is valuable for the decision making process. Additionally, by obtaining Pareto optimal solutions of a multi-objective problem, one can obtain useful design information such as which objective functions are conflicting/independent, which design parameters are sensitive/insensitive to the objective functions, which design parameters are dependent/independent, etc. Such information is useful for designers because it helps them to design and develop real-world products. A multiobjective optimization usually results in tenths (sometimes hundreds) of Pareto optimal solutions, each of which has multiple objective function values, constraint function values, and design parameter values. Thanks to rapid improvement in computational speed, the number of Pareto-optimal solutions we can obtain is increasing. One important aspect in practical multi-objective shape optimization is, thus, how to understand the hundreds of Pareto optimal solutions obtained and how to extract useful knowledge from them.

Data mining tools, along with data visualization using graphical methods, can help to understand and extract information from the data contained in the Pareto optimal solutions found using any MOEA. However some graphical techniques are limited to a maximum number of objective functions. In this sense, Multi-Objective Design Exploration (MODE), proposed by Jeong et al. [105] is a framework to extract design knowledge from the obtained Pareto optimal solutions such as trade-off information between contradicting objectives and sensitivity of each design parameter to the objectives. In the framework of MODE, Pareto-optimal solutions are obtained by a MOEA and knowledge is extracted by analyzing the design parameter values and the objective function values of the obtained Pareto-optimal solutions using data mining approaches such as Self Organizing Maps (SOMs) and analysis of variance (ANOVA). They also propose to use rough sets theory to obtain rules from the Pareto optimal solutions.

4.4 REMARKS FOR MOEAS IN AERONAUTICAL/AEROSPACE ENGINEERING

We have reviewed the application of MOEAs used for solving a diversity of aeronautical/aerospace engineering problems. From this review we identified that many of the existing applications rely on the use of state-of-the-art MOEAs such as: *NSGA*, *NCGA*, *MOGA*, *SPEA2*, *NSGA-II*. Most of them are Pareto-based MOEAs which use genetic algorithms as their search engine. More recently, there have been attempts to use other types of MOEAs, based on different search operators such as evolution strategies, in algorithms such as: *GAME*, *($\mu + \mu$)-ES*, *($1 + 10$)-MODES*, and *HAPMOEA*; and particle swarm optimization *(MOPSO)*. Additionally, the use of indicator-based MOEAs such as the *SMS-EMOA* is also reported in the specialized literature. In the context of aerodynamic shape optimization problems, there are two MOEAs that were specifically designed for dealing with them: *ARMOGA* and $\epsilon\mu$-*ARMOGA*. These two latter algorithms are based on *MOGA*, and make use of range adaptation for the design variables in both cases, and a μ-*GA* and ϵ-dominance in the latter case. We also identified that the use of other metaheuristics with good convergence properties, such as *differential evolution*, has been scarce in this domain.

In spite of the considerable amount of research currently available on the use of MOEAs for solving these types of problems, there exists a continuous need to develop new MOEA techniques that can reduce the computational cost, measured in terms of the number of objective function evaluations, required for solving the complex type of problems commonly found in these disciplines. We conclude that the use of MOEAs in aeronautical/aerospace engineering optimization is a mature area and that the use of metaheuristics such as *differential evolution* is a promising research path in these engineering fields. This condition motivated us to design an algorithm based on this metaheuristic. Also, many of the applications reviewed, rely on the concept of Pareto dominance for its selection process, and crowding mechanisms for distribution of solutions along the Pareto front. These mechanisms has proven to be very effective, but as indicated by Goel et al. [241], due to the finite size of the population, some of the good solutions make way for the other solutions. If the lost solutions are the Pareto optimal solutions, the loss may not be repaired and sub-Pareto optimal solutions are obtained as the final solution. This problem is named as *Pareto drift*. In order to avoid this condition, in our approaches we explore the use of different selection mechanisms such as the use of scalarization functions.

5 OUR PROPOSED APPROACHES

5.1 INTRODUCTION

In this chapter, we present a novel MOEA called *MODE-LD+SS*, which adopts *differential evolution (DE)* [190] as its global search engine. Our main motivation to use DE was that MOEAs based on this search engine have been found to be very effective, outperforming those based on genetic algorithms [242]. Our proposed approach incorporates two additional mechanisms to those normally found in a MOEA. The first (local dominance) is used to improve the convergence rate towards the Pareto front, while the second (a selection mechanism based on a scalarization function) is used to find nondominated solutions covering the entire Pareto front.

Also, we have developed two different parallel schemes for improving the performance of the basic *MODE-LD+SS*. The first is designed for improving effectiveness (i.e., for better approximating the Pareto front), while the second is designed for improving efficiency (i.e., for reducing the execution time by using small population sizes in each sub-population). Either approach (or both) can be of interest in solving real-world engineering MOPs. The two proposed parallel schemes are based on the *island paradigm*, and use the multi-objective differential evolution algorithm *MODE-LD+SS* as their search engines. These parallel schemes will also be described in this section.

5.2 DIFFERENTIAL EVOLUTION

DE is a simple and powerful evolutionary algorithm that has been found to outperform genetic algorithms in a variety of numerical single-objective optimization problems [190]. DE encodes solutions as vectors and uses operations such as vector addition, scalar multiplication and exchange of components (crossover) to construct new solutions from the existing ones. DE operates as follows: a newly created solution, also called *candidate*, is compared to its parent. If the candidate is better than its parent, it replaces the parent in the population; otherwise, the candidate is discarded. Being a steady-state algorithm, it implicitly enforces *elitism*, i.e., no solution from the population can be deleted unless a better solution is created. DE was originally proposed to be used with real-numbers encoding.

The above indicated process is described as follows:

For each current vector $P_i \in \{P\}$, three parents (mutually different among them) $\vec{u}_1, \vec{u}_2, \vec{u}_3 \in \{P\}$ $(\vec{u}_1 \neq \vec{u}_2 \neq \vec{u}_3 \neq P_i)$ are randomly selected for creating a mutant vector $\vec{v}$ using the following mutation operation (Figure 20 graphically shows this process):

$$\vec{v} \leftarrow \vec{u}_1 + F \cdot (\vec{u}_2 - \vec{u}_3) \tag{16}$$

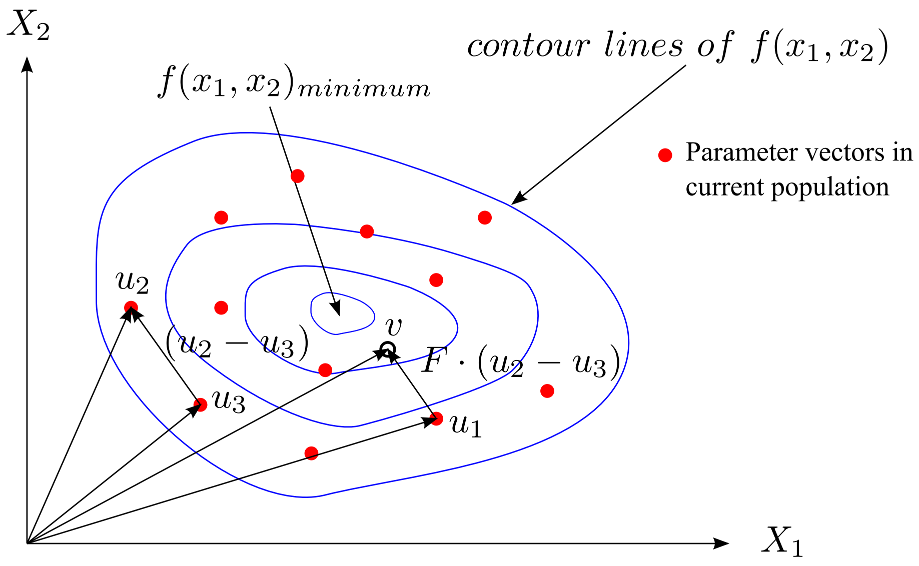


Figure 20: Differential evolution mutation process

$F > 0$, is a real constant called *scaling factor* which controls the amplification of the difference $(\vec{u}_2 - \vec{u}_3)$. Using this mutant vector, a new offspring $P_i^{'}$ (also called trial vector in DE) is created by crossing over the mutant vector $\vec{v}$ and the current solution P_i (cf. Figure 21), in accordance to:

$$P_j^{'} = \begin{cases} v_j & \text{if } (rand_j(0,1) \leqslant CR \text{ or } j = j_{rand} \\ P_j & \text{otherwise} \end{cases} \tag{17}$$

In the above expression, the index j refers to the jth component of the decision variables vectors. CR is a positive constant and j_{rand} is a randomly selected integer in the range $[1, \ldots, D]$ (where D is the dimension of the solution vectors) ensuring that the offspring is different at least in one component with respect to the current solution P_i. The above DE variant is known as $\mathrm{rand}/1/\mathrm{bin}$.

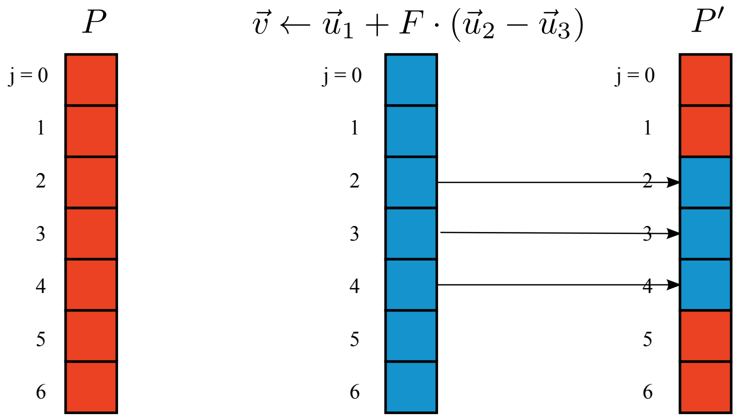

P
$\vec{v} \leftarrow \vec{u}_1 + F \cdot (\vec{u}_2 - \vec{u}_3)$
P'
j = 0
1
2
3
4
5
6
j = 0
1
2
3
4
5
6
j = 0
1
2
3
4
5
6

Figure 21: Differential evolution recombination, rand/1/bin scheme

5.3 MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION

DE has been adopted to solve MOPs in several ways. In the earlier approaches (PDE [1] and GDE [141]), only the concept of Pareto dominance was used to compare individuals. The parent was replaced only if it was dominated by the candidate, and it was discarded otherwise. Many subsequent approaches (PDEA [155], MODE [262], NSDE [94], GDE2 [135], DEMO [202], GDE3 [136] and NSDE-DCS [95]), use nondominated sorting and/or the crowding distance metric to evaluate the fitness of the individuals. Only recently, new algorithms that do not follow the environmental selection of NSGA-II were proposed, such as $\epsilon - MyDE$ [207], DEMORS [82], $\epsilon - ODEMO$ [31], and MOEA/D-(DE) [149]. The first three algorithms of this last group, make use of the ϵ-dominance concept, as proposed by Laumanns et al. [145]. ϵ-dominance is adopted for spreading the solutions in a uniform manner without resorting to any crowding distance metric. Finally the MOEA/D-(DE) algorithm [149] is based on the MOEA/D of Zhang and Li [267], but using the differential evolution operators. A comprehensive review of some of these multi-objective differential evolution approaches can be found in [161].

5.4 MODE-LD+SS

The MOEA we propose is called *MODE-LD+SS*, and adopts the evolutionary operators from differential evolution. Additionally, the proposed algorithm incorporates two mechanisms for improving both the convergence towards the Pareto front and the uniform distribution of nondominated solutions along the Pareto front. These mechanisms correspond to the concept of local dominance and the use of an environmental selection based on a scalar function. Below, we explain these two mechanisms in more detail. Algorithm 2 shows the description of our proposed *MODE-LD+SS*.

In Algorithm 2, the solution vectors $\vec{u}_1, \vec{u}_2, \vec{u}_3$, required for creating the trial vector $\vec{v}$ (in equation (16)), are selected from the current population, only if they are locally nondominated in their neighborhood $\aleph$. Local dominance is defined as follows:

Definition 6. Pareto local dominance Let $\vec{x}$ be a feasible solution, $\aleph(\vec{x})$ be a neighborhood structure for $\vec{x}$ in the decision space, and $\vec{f}(\vec{x})$ a vector of objective functions.

- We say that a solution $\vec{x}$ is locally nondominated with respect to $\aleph(\vec{x})$ if and only if there is no $\vec{x}'$ in the neighborhood of $\vec{x}$ such that $\vec{f}(\vec{x}') \prec \vec{f}(\vec{x})$

The neighborhood structure is defined as the NB closest individuals to a particular solution. Closeness is measured using the Euclidean distance between solutions. The major aim of using the local dominance concept, as defined above, is to exploit good individuals' genetic information in creating DE trial vectors, and the associated offspring, which might help to improve the MOEA convergence rate toward the Pareto front. Figures 22 and 23 show, in decision and objective function space, respectively, the solutions satisfying the local dominance criteria; in this case a neighborhood size of 4 is used. From this subset of solutions, the offspring will be created (cf. Figure 24).

From Algorithm 2, and Figures 22 to 24, it can be noted that this mechanism has a stronger effect during the earlier generations, when the portion of nondominated individuals is low in the global population, and progressively weakens, as the number of nondominated individuals grows during the evolutionary process. This mechanism is automatically switched off, once all the individuals in the population become nondominated, and has the possibility to be switched on, as some individuals become dominated. Additionally, the diversity of the created offspring can be controlled by the local dominance neighborhood size NB. Low values of NB will increase the diversity of offspring, and viceversa.

Algorithm 2 MODE-LD+SS

```
INPUT:
N = Population Size
F = Scaling factor
CR = Crossover Rate
$\lambda[1, \ldots, N]$ = Weight vectors
NB = Neighborhood Size
GMAX = Maximum number of generations
OUTPUT:
PF = Pareto front approximation
Begin
$g \leftarrow 0$
Randomly create $P_i^g$ , $i = 1, \ldots, N$
Evaluate $P_i^g$ , $i = 1, \ldots, N$
while $g < GMAX$ do
    $\{LND\} = \{\oslash\}$
    for $i = 1$ to N do
        DetermineLocalDominance($P_i^g$,NB)
        if $P_i^g$ is locally nondominated then
            $\{LND\} \leftarrow \{LND\} \cup P_i^g$
        end if
    end for
    for $i = 1$ to N do
        Randomly select $\vec{u}_1$, $\vec{u}_2$, and $\vec{u}_3$ from $\{LND\}$
        $\vec{v} \leftarrow$ CreateMutantVector($\vec{u}_1, \vec{u}_2, \vec{u}_3$)
        $P_i^{g+1} \leftarrow$ Crossover($P_i^g, v$)
        Evaluate $P_i^{g+1}$
    end for
    $Q \leftarrow P^g \cup P^{g+1}$
    Determine $z*$ for Q
    for $i = 1$ to N do
        $P_i^{g+1} \leftarrow$ MinimumTchebycheff($Q, \lambda^i, z*$)
        $Q \leftarrow Q \backslash P_i^{g+1}$
    end for
    $PF \leftarrow \{P\}^{g+1}$
end while
Return PF
End
```

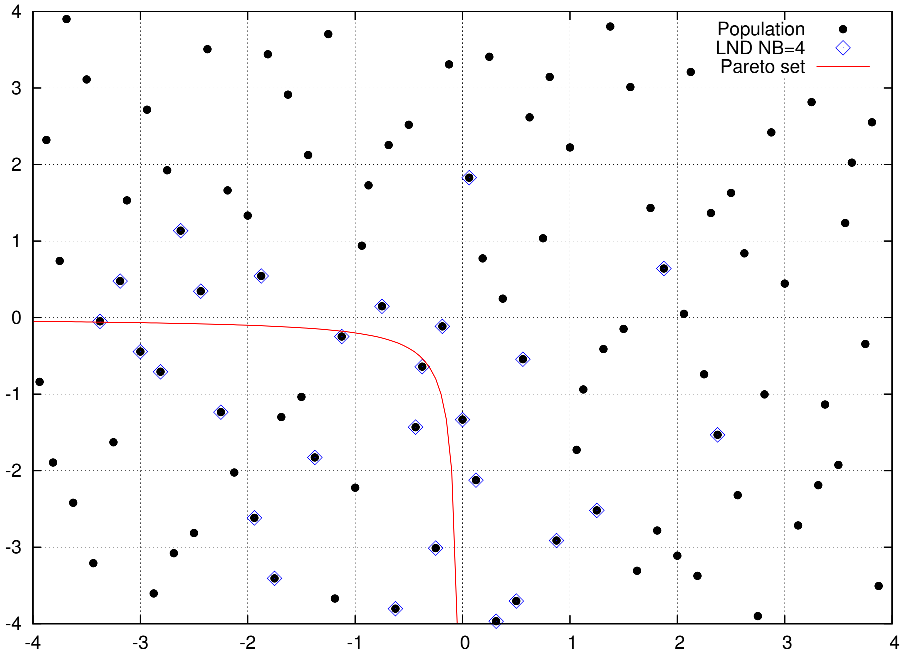


Figure 22: Locally nondominated solutions in design space

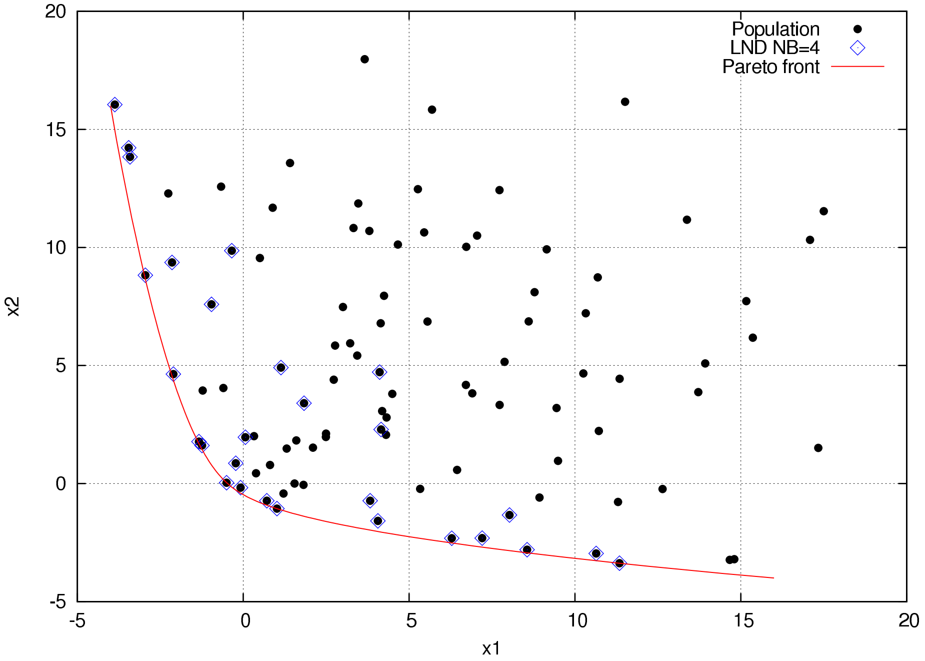


Figure 23: Locally nondominated solutions in objective space

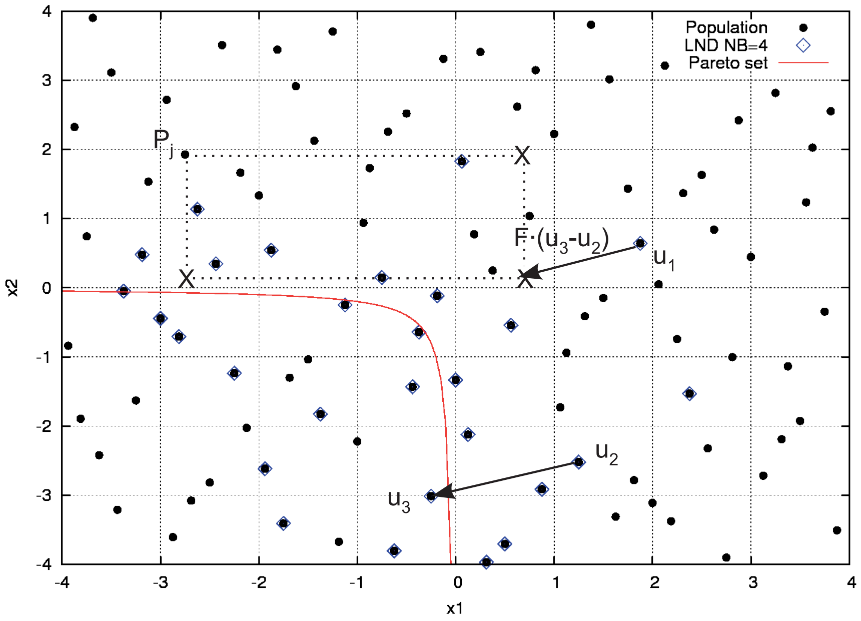


Figure 24: DE rand/1/bin recombination process using locally nondominated solutions

The second mechanism that we introduced is called *selection based on a scalar function*, and is based on the Tchebycheff scalarization function given by:

$$g(\vec{x}|\lambda^j, z^*) = \max_{1 \leqslant i \leqslant m} \{\lambda_i^j |f_i(x) - z_i^*|\} \tag{18}$$

In the above equation, $\lambda^j, j = 1, \ldots, N$ represents the set of weight vectors used to distribute the solutions along the entire Pareto front (see Figure 25). In this work, this set is calculated using the procedure described in Zhang and Li [267]. z^* corresponds to a reference point, defined in objective space and is determined with the minimum objective values of the combined population Q, consisting on the current parents and the created offspring. This reference point is updated at each generation, as the evolution progresses. The procedure *MinimumTchebycheff(*$Q, \lambda^i, z*$*)* finds, from the set Q, (the combined population consisting on the current parents and the created offspring), the solution vector that minimizes equation (18) for each weight vector λ^i and the reference point z^*. Figure 26 illustrates this process.

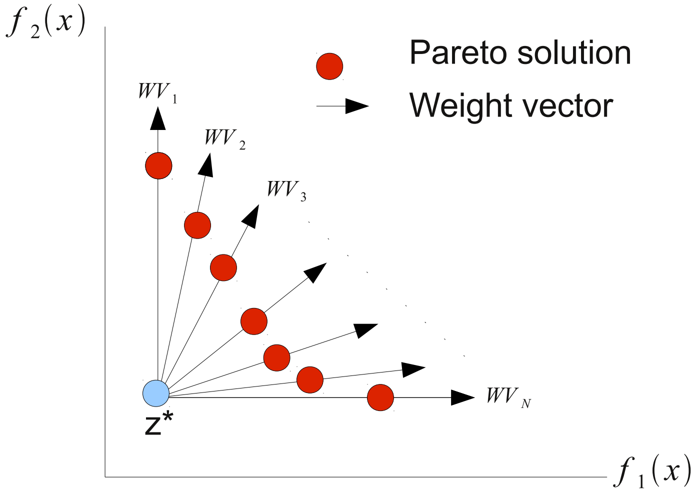


Figure 25: Distribution of the weight vectors

5.5 PMODE-LD+SS

Multi-objective evolutionary algorithms (MOEAs) have been found to be very suitable for solving a wide variety of engineering optimization problems, because of their generality, ease of use and relatively low susceptibility to the specific features of the search space of the problem to be solved [41]. Nonetheless, they are normally computationally expensive due to several reasons: (1) real-world optimization problems typically involve high-dimensional search spaces and/or a large number of objective functions, (2) they require finding a set of solutions instead of only one, often requiring, in consequence, large population sizes, and (3) frequently, the task of evaluating the objective functions demands high computational costs (e.g., complex computer simulations are required). All these factors decrease the utility of serial MOEAs when applied to real-world engineering Multi-objective Optimization Problems (MOPs). In order to reduce the execution time required to solve these problems two main types of approaches have been normally adopted[1]: (1) Enhance the MOEA's design, namely improving its convergence properties, so that the number of objective function evaluations can be reduced, and, (2) Use of parallel programming techniques, i.e., to adopt a parallel or distributed MOEA.

[1] Our discussion here is focused exclusively on MOEAs that use exact objective function values, but fitness approximation schemes and surrogate models can also be used to deal with expensive MOPs [109, 126].

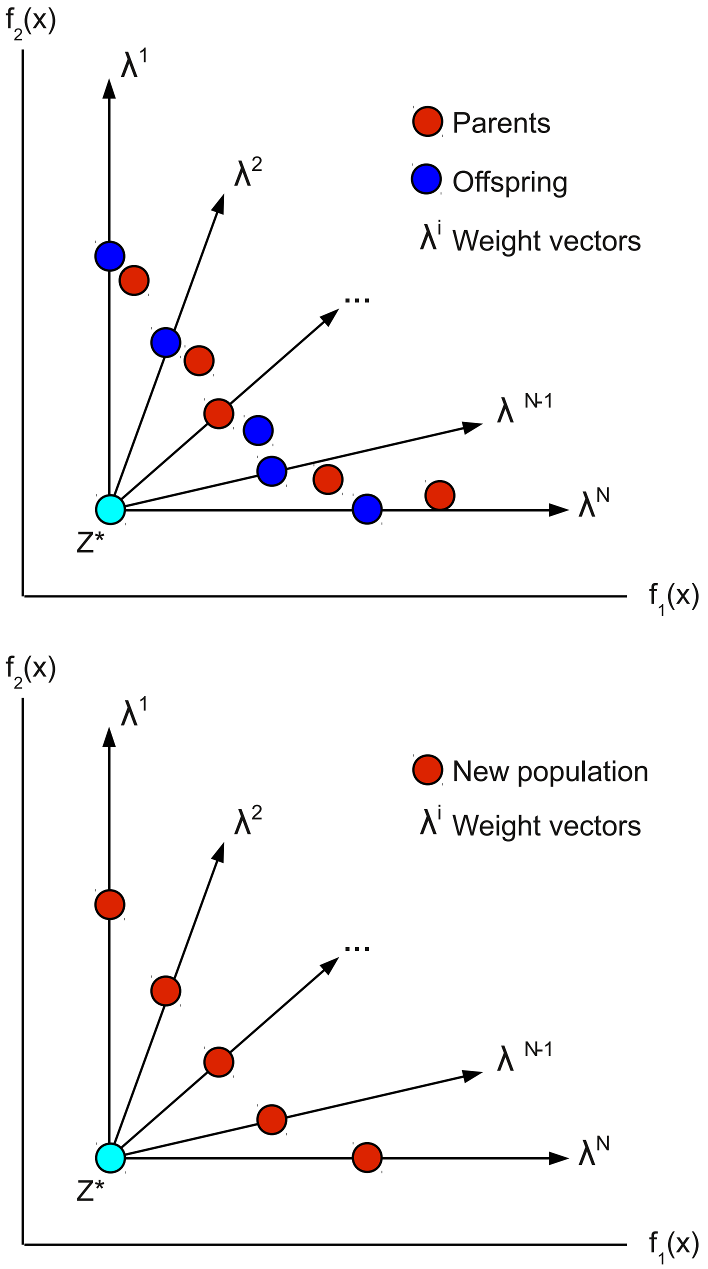


Figure 26: Selection process in *MODE-LD+SS*

Based on the above, two different parallel schemes were developed for improving the performance of the serial version of MODE-LD+SS. The first is designed for improving effectiveness (i.e., for better approximating the Pareto front), while the second is designed for improving efficiency (i.e., for reducing the execution time by using small population sizes in each sub-population). Either approach (or both) can be of interest in solving real-world engineering MOPs. The two proposed schemes are based on the *island paradigm*, and use the multi-objective differential evolution algorithm MODE-LD+SS as their search engine.

5.5.1 *MOEA parallelization*

MOEAs, being population-based approaches, are very suitable for parallelization because their main operations (i.e., crossover, mutation, and, particularly, objective function evaluation) can be applied independently on different individuals and/or groups of them. A pMOEA can be useful to solve problems faster, but also for generating novel and more efficient search schemes i.e., a pMOEA can be more effective than its sequential counterpart, even when is executed in a single processor machine [231]. From the specialized literature, four major pMOEA paradigms are commonly used [41, 246]. These are (i) master-slave, (ii) island, (iii) diffusion, and (iv) hierarchical or hybrid. A comprehensive review of these paradigms can be found in [41, 246]. It is important to note that each paradigm can be implemented in either a synchronous or in an asynchronous manner. Synchronous means that all communication processes are synchronized at previously-defined check points. On the other hand, asynchronous implementations consider inter-processor communications occurring at random and with no warranty of delivering messages to their intended destinations. We are interested here in the *island model*, which is detailed next.

5.5.2 *Island pMOEA model*

The island paradigm for pMOEAs is based on the phenomenon of natural populations evolving independently. At each island, a serial MOEA is executed for a predefined number of generations called *epoch*. At the end of an epoch, communication between neighboring islands is allowed. During this communication process, individuals (or copies of them in the case of pollination) can migrate from their current island to a different one according to a predefined *migration topology* (see Figure 27), which determines the migration path along

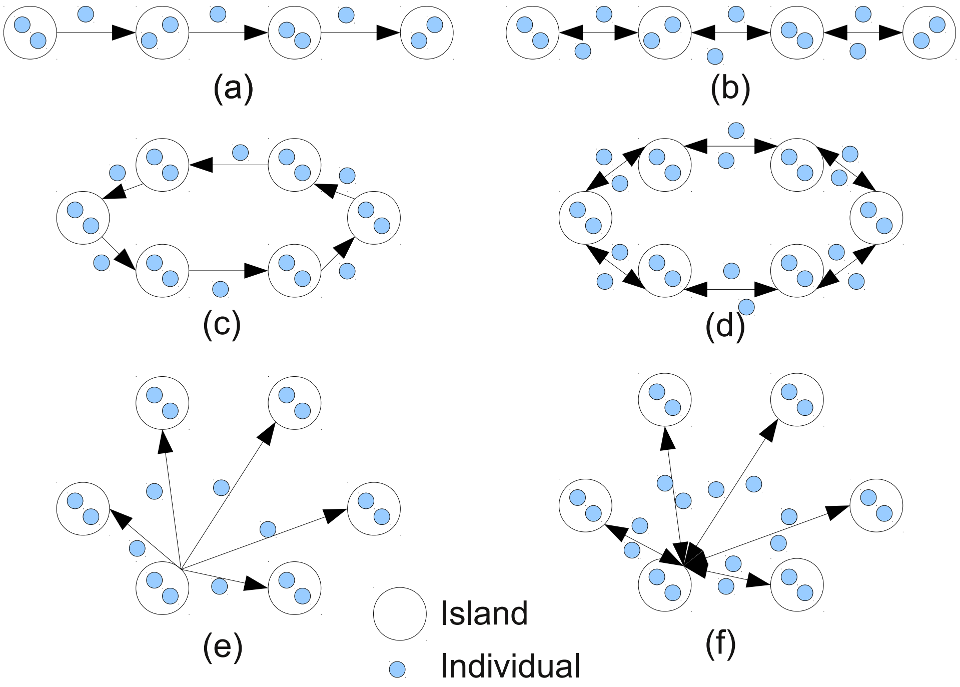


Figure 27: Examples of migration topologies. (a) linear/unidirectional, (b) linear/bidirectional, (c) ring/unidirectional, (d) ring/bidirectional, (e) random/unidirectional, (f) random/bidirectional

which individuals can move. For migration, not only one individual, but a set of them, can migrate; the number of migrants is defined by the *migration rate*. Additionally, the island model requires the definition of policies for selecting *migrants* (migration selection policy) as well as selecting the individuals in a subpopulation that will be replaced by migrants (replacement policy). The migration process allows gene mixing between different islands, which helps in maintaining global population diversity. Also, different evolutionary operators can be assigned to each island, allowing each subpopulation to search many different regions of the whole search space. This condition can be amplified if different random number generators and/or seeds are used in each island. Finally, for the island model, four basic pMOEAs variants can be considered, each one with a specific migration parameter set: (1) *Homogeneous*, where all islands execute identical MOEAs/parameters, (2) *Heterogeneous*, where all islands execute different MOEAs/parameters, (3) Each island evaluates different objective function subsets, and (4) each island represents a different region of the genotype or phenotype domains.

5.5.3 *Previous related work*

In this section, we review approaches that use the island paradigm for developing pMOEAs.

Kamiura et al. [115] presented a pMOEA called MOGADES (Multi-Objective Genetic Algorithm with Distributed Environment Scheme). In this pMOEA, the population is divided into M islands, and in each of them the MOP is converted into a scalar one, i.e., a different weight vector is assigned to each island. The aim of this algorithm is that each island can capture a different region of the Pareto front. One important aspect in this approach is that when migration occurs, the weights for each island are varied. A major drawback for this approach is that a good distribution of solutions cannot be guaranteed as it depends on the dynamics of the evolutionary system, i.e., of the weight vector variation.

Deb et al. [52] proposed a pMOEA based on the NSGA-II [51]. This approach attempts to distribute the task of finding the entire Pareto optimal front among the several participating islands. All islands search on the entire decision variable space, but for each of them, a different region of the Pareto optimal front is assigned. For guiding the search, the approach uses a guided domination concept defined by Branke et al. [26]. Additionally, the approach requires the definition of a set of *hyperplanes* for dividing the entire Pareto optimal front among the participating processes. A drawback of this approach is that the shape and/or continuity of the Pareto front must be known *a priori*, in order to define the search regions to be explored. Furthermore, the approach described in this paper can only be applied to convex Pareto fronts.

Streichert et al. [228] proposed a pMOEA, which combines an island model with the "divide and conquer" principle. This approach partitions the population using a clustering algorithm (k-means), with the aim of assigning to each island, the search task of a particular Pareto front region. In this approach, at each epoch, the sub-populations are gathered by a master process for performing the clustering/distributing process. The individuals in each island are kept within their assigned Pareto front region using zone constraints. The main drawback of this approach is that *a priori* knowledge of the Pareto front shape is needed to define the zone constraints.

Zahaire and Petcu [264] developed the *multi-population APDE* (APDE stands for Adaptive Pareto Differential Evolution). This approach consists of dividing the main population into sub-populations (islands), each of equal size. In each island, a serial version of the APDE is executed with its own set of randomly initialized adaptive parameters, and is evolved for an epoch. Afterwards, a migration process is started. This process is based on a random connection topology, i.e., each individual from each sub-population can be swapped (with a given migration probability) with a randomly selected individual from another randomly selected island.

5.5.4 *Our proposed approach*

Based on the serial version of MODE-LD+SS, previously described, we present here two parallelization schemes. The first is designed for improving effectiveness[2] and is called *pMODE-LD+SS*(A). The second is designed for improving efficiency, and is called *pMODE-LD+SS*(B). Both of them share the following characteristics:

- Use of a "random pair-islands" bidirectional migration scheme (cf. Figure 28). In this scheme, at each epoch, pairs of islands are randomly selected. Then, the communication is performed between each pair of islands. Migrants from one island are considered as immigrants in the receptor island, and viceversa.

- Use of a pollination scheme, i.e., copies of selected migrants are sent, while the original individuals are retained in their own population.

- The migration policy is based on randomly selected individuals.

- The replacement policy is based on the environmental selection mechanism adopted in the serial version running in each island. In this case, immigrants are added to the receptor island's population, and the environmental selection process is applied to this extended population.

The main difference between the two proposed approaches is on the weight vectors distribution used. The *pMODE-LD+SS*(A) approach can be seen as the serial version of MODE-LD+SS running in p processors and exchanging information among them. For this approach, the same weight vector distribution (see Figure 25) is used in each island. For maintaining diversity of the global population and to evolve each island in an independent manner, different seed values are used in the islands' random numbers generators. In the second case, for the *pMODE-LD+SS*(B) approach, each island is also instructed to search for the whole Pareto front, but in this case, using a reduced

[2] Here, effectiveness refers to the degree to which objectives are achieved and the extent to which targeted problems are solved. In contrast to efficiency, effectiveness is determined without reference to costs.

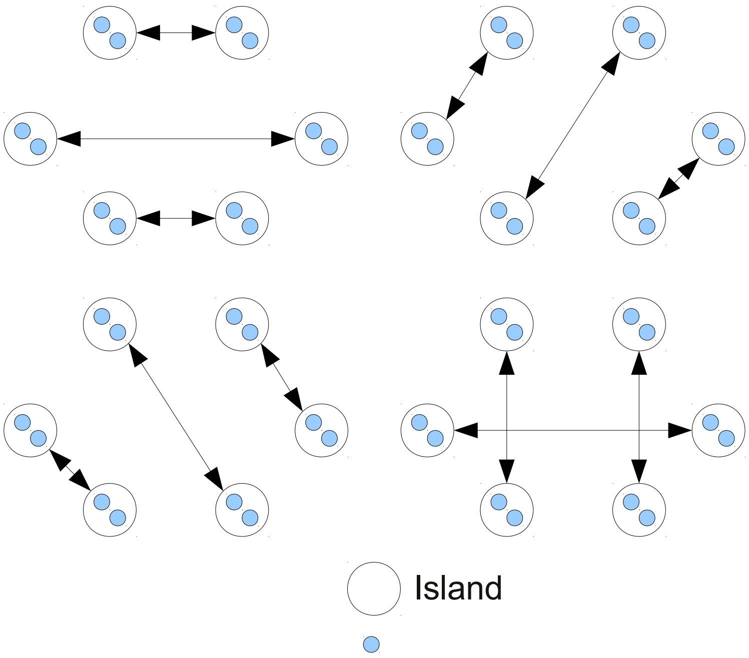


Figure 28: Examples of the migration topology adopted in *pMODE-LD+SS*

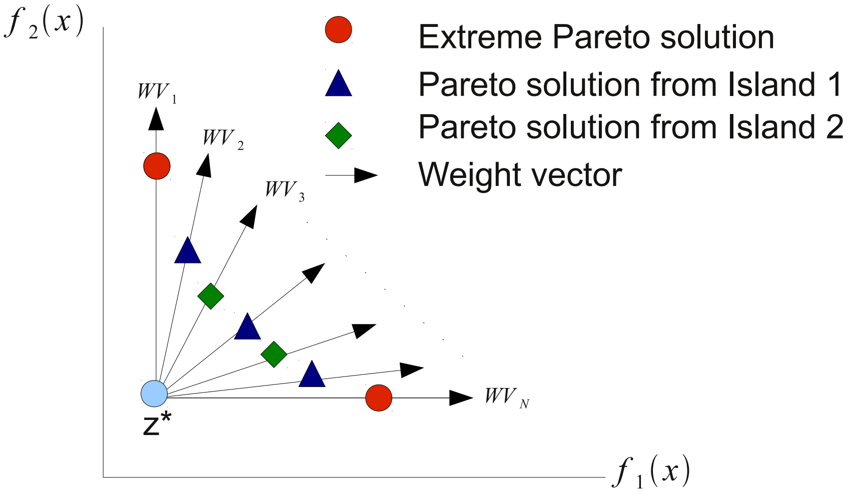


Figure 29: Weight vectors distribution for *pMODE-LD+SS*(B)

population and different weight vectors sets. It is important to note that all islands contain weight vectors for searching the extreme Pareto solutions. The main idea for the second parallel approach is that the combination of all islands' weight vectors covers the whole Pareto front region. Figure 29 illustrates this situation for the case of a bi-objective MOP with two islands participating in the pMOEA.

6 EXPERIMENTAL SETUP

6.1 INTRODUCTION

The major aim in designing any new MOEA is that it can be used for solving real-world multi-objective optimization problems with a high level of reliability and efficiency. Nonetheless, a commonly encountered situation is that real-world MOPs are very hard to solve and, in many cases, they require high computational efforts. Additionally, the nature and characteristics of their fitness landscapes are not totally understood, and even the optimal solutions are not known a priori. These two situations make it difficult to evaluate how well any MOEA has performed when applied to the solution of a particular real-world MOP. For this purpose, there is a clear need to develop and design test MOPs, or more generally, a test MOP suite, which might be helpful in assessing, comparing, classifying, and improving MOEA performance, in terms of their *effectiveness* and *efficiency*, before deciding to apply it to the solution of any particular real-world MOP.

The MOEA community has created a variety of test MOPs and suites. In the early days of MOEAs, several numeric MOP functions were adopted to assess MOEA's performance. For example, Schaffer's study introduced two MOPs, SCH1 and SCH2 [214, 215], being both single variable test problems, and having two objective functions each. Kursawe's test problem KUR [139], had a scalable number of decision variables and has two objective functions. The same conditions are present for Fonseca and Fleming's test problem FON [70]. Poloni et al.'s test problem POL [188] used only two decision variables and comprised two objective functions. Although the mathematical formulation of the problem in the latter case, is nonlinear, the resulting Pareto-optimal front corresponds to an almost linear relationship among the decision variables. Viennet's test problem VNT [252] has a discrete set of Pareto-optimal fronts, and was designed with three objective functions.

The solution of MOPs requires two main tasks: (1) to converge as close as possible to the *true* Pareto-optimal front, and (2) to obtain as diverse a set of solutions as possible, covering the entire Pareto-optimal front. In this sense any MOP should be able to test a MOEA's ability to overcome artificial difficulties or hurdles which might prevent and/or retard MOEAs' progress in converging towards the true Pareto-optimal front. This is achieved in numerical MOPs by placing, for example, some local Pareto-optimal attractors. Additionally, any MOP should also be able to test MOEA's ability to find diverse sets of solutions along the approximated Pareto front. For this sake, numerical MOPs are designed with varied Pareto-optimal front geometries such as, non-convex, disconnected, and having variable density of solutions along them.

More recently, research work has been conducted in designing test suites by using systematic procedures to obtain MOPs presenting the following desired features as indicated in [41, 53]:

1.- Controllable hindrance to converge to the true Pareto-optimal front and also in finding a widely distributed set of Pareto-optimal solutions.

2.- Scalability in terms of the number of decision variables.

3.- Scalability in terms of the number of objective functions.

4.- Simplicity and diversity for the construction of test MOPs.

5.- Pareto-optimal front with a shape and location exactly known. (e.g., Pareto fronts having a closed form)

6.- Test MOPs useful in practice, i.e., exhibiting difficulties similar to those that can be present in most real-world MOPs.

A number of MOP test suites and test function generators currently exist. The most commonly adopted in the specialized literature are the following:

- **Zitzler-Deb-Thiele (ZDT) test functions**: They were proposed by Zitzler et al. [273]. This suite comprises a set of six two-objective functions. These MOPs do not scale in the number of objectives. The ZDT MOPs are defined as follows:

$$\begin{aligned} &Minimize\ F = f(f_1(\vec{x}), f_2(\vec{x}))\ ,\ where \\ &f_1 = f(x_1, \ldots, x_m) \\ &f_2 = g(x_{m+1}, \ldots, x_N)h(f(x_1, \ldots, x_m), g(x_{m+1}, \ldots, x_N)) \end{aligned} \tag{19}$$

  where function f_1 is a function of ($m < N$) decision variables and f_2 a function of all N decision variables. The function g is one of $(N - m)$

decision variables which are not included in function f. The function h is directly a function of f and g values. The f and g functions are also restricted to positive values in the search space, i.e. $f > 0$, and $g > 0$. For this test suite, their authors proposed several functions for f_1, g, and h [273], which then, can be *mixed* and *matched* to create different MOPs, each one having different desired characteristics [41]. For this test suite the f function controls the vector representation uniformity along the Pareto front, g controls whether the MOP will be multifrontal or if it will contain a single global Pareto front. h controls the MOPs' characteristics such as convexity, disconnectedness, etc.

- **Deb-Thiele-Laummans-Zitzler (DTLZ) test functions**: This test suite was proposed by Deb et al. [49]. The MOPs generated in this case attempt to define generic MOPs that are scalable to a user defined number of objectives. Usually this set is used to test MOEAs in solving MOPs with three or more objective functions. In Deb et al. [49] three main methods are discussed for generating scalable MOPs. They are: *(i) Multiple single-objective approach*, *(ii) Bottom-up approach*, and *(iii) Constrained surface approach*. The first method, being the most intuitive has been used implicitly by early MOEA researchers [214, 215, 139, 70, 188, 252]. The DTLZ test suite follows the two latter approaches.

In the *bottom-up approach*, a mathematical function describing the Pareto-optimal front is assumed in the objective space, and an overall search space is constructed from this front to define the test problem. In an early study, Deb [50] suggested a generic MOP test problem generator in which, for M objective functions, with a complete decision variables vector partitioned into M non-overlapping groups

$$\vec{\mathbf{x}} \equiv (\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_{M-1}}, \mathbf{x_M})^T \tag{20}$$

the following function structure was suggested [53]

$$
\begin{aligned}
&\text{Minimize} \quad f_1(\mathbf{x_1})\\
&\text{Minimize} \quad f_2(\mathbf{x_2})\\
&\qquad\vdots\\
&\text{Minimize} \quad f_{M-1}(\mathbf{x_{M-1}})\\
&\text{Minimize} \quad f_M(\mathbf{x}) = g(\mathbf{x_M})h(f_1(\mathbf{x_1}), f_2(\mathbf{x_2}), \dots, f_{M-1}(\mathbf{x_{M-1}}), g(\mathbf{x_M}))\\
&\text{subject to } x_i \in \mathbb{R}^{|X_i|} \text{ for } i = 1, 2, \dots, M
\end{aligned}
\tag{21}
$$

Here, the Pareto-optimal front is described by solutions which are the global minimum of $g(\mathbf{X_M})$ (with g^*). Thus, the Pareto-optimal front is described as

$$f_M = g^* h(f_1, f_2, \dots, f_{M-1}) \tag{22}$$

Since g^* is a constant, the h function (with fixed $g = g^*$) describes the Pareto-optimal surface.

Similar to the ZDT suite, the f, g, and h functions control the MOPs characteristics (i.e., uniformity of solutions along the Pareto front, multifrontality and Pareto front's shape). The advantage of using the above defined *bottom-up approach* is that the exact form of the Pareto-optimal surface can be controlled by the developer. The number of objectives and the variability in density of solutions can all be controlled by choosing proper g and h functions. Since the search space is constructed from identical functionals, the search space is also structured

[49]. On the other hand, in the *constrained surface approach*, the MOP test generator begins by predefining a simple overall search space:

$$\begin{aligned}
&\text{Minimize} \quad f_1(x),\\
&\text{Minimize} \quad f_2(x),\\
&\qquad \vdots\\
&\text{Minimize} \quad f_M(x),\\
&\text{Subject to}: \; f_i^{(L)} \leqslant f_i(x) \leqslant f_i^{(U)} \text{ for } i = 1, 2, \ldots, M
\end{aligned} \tag{23}$$

It is intuitive, from the above problem definition, that the Pareto-optimal set has only one solution, i.e., the solution with the lower bound of each objective $[f_1^{(L)}, f_2^{(L)}, \ldots, f_M^{(L)},]^T$. The problem is then made more interesting by adding a series of constraints (linear or non-linear):

$$g_j(f_1, f_2, \ldots, f_M) \geqslant 0 \;\text{ for }\; j = 1, 2, \ldots, J \tag{24}$$

Each constraint eliminates some portion of the original rectangular region in a systematic way. We can observe that in the *constrained surface approach* for generating MOPs, the aim is to find the nondominated portion of the boundary of the feasible search space. Difficulties can be introduced by using varying density of solutions in the search space, i.e., by using non-linear functionals for f_i within the decision variables. The construction process in this latter approach is much simpler as compared to the *bottom-up approach*. Using the *constrained surface approach*, different shapes (convex, non-convex, or discrete) can be generated for the Pareto-optimal region [49].

The DTLZ test generator as proposed, is built using both the *bottom-up* and the *constrained surface* approaches, and contains 9 test families of MOP geometries, two of them being constrained MOPs.

- **Okabe's test functions**: This MOP test function generator was proposed by Okabe et al. [174], and is based on a different methodology which considers the mapping of probability density functions, from the decision variable space to the objective function space. The basic idea in generating a MOP in this test suite is to depart from a starting space, called S^2 between decision variable space and objective function space, and from there, the user constructs the corresponding decision variable space and objective function space by applying transformations to S^2. In the proposed suite, the authors use the inverse of generation operation, i.e., deformation, rotation and shift. For a detailed description of the construction of these test MOPs, the reader is referred to [174, 41].

- **Huband's test functions**:This MOP test function generator was proposed by Huband et al. [91]. In this test suite the authors propose a set of transformations which are sequentially applied to the decision variables. The major aim in this suite is that each transformation adds a different desired characteristic to the MOP. All the problems which are generated in this suite follow the format:

$$\begin{aligned}
\mathrm{Given}\ \mathbf{z} &= [z_1, \ldots, z_k, z_{k+1}, \ldots, z_n] \\
\mathrm{Minimize}\ f_{m=1:M}(\mathbf{x}) &= Dx_M + S_m h_m(x_1, \ldots, x_{M-1}) \\
\mathrm{where}\ \mathbf{x} &= [x_1, \ldots, x_M] \\
&= [\max(t^p_M, A_1)(t^p_1 - 0.5) + 0.5, \ldots, \\
&\quad \max(t^p_M, A_{M1})(t^p_{M1} - 0.5) + 0.5, t^p_M] \\
\mathbf{t}^p &= [t^p_1, \ldots, t^p_M] \mapsfrom \mathbf{t}^{p-1} \mapsfrom \ldots \mapsfrom \mathbf{t}^1 \mapsfrom \mathbf{z}_{[0,1]} \\
\mathbf{z}_{[0,1]} &= [z_{1,[0,1]}, \ldots, z_{n,[0,1]}] \\
&= [z_1/z_{1,\max}, \ldots, z_n/z_{n,\max}]
\end{aligned} \tag{25}$$

where $\mathbf{z}$ is the vector of decision variables with $0 \leqslant z_i \leqslant z_{i,\max}$, D, $A_{1:M-1}$ and $S_{1:M}$ are constants to modify the position and the scale of the Pareto front. The flexibility for this MOP test suite lies on the $h_{1:M}$ functions, and the transformations to obtain the transition vectors $\mathbf{t}^{1:p}$, which keep desired characteristics of the problem as separate design decisions. First, the $h_{1:M}$ functions define the shape of the Pareto front, which can be linear, convex, concave, mixed (convex and concave), and disconnected. For a detailed description of several functions proposed for $h_{1:M}$, the interested reader is referred to [91, 41].

Once the shape of the Pareto front is designed in the Huband's MOP test suite, the rest of the characteristics are added through a set of transformations. Huband et al. distinguish between three types of transformations,

> based on the characteristics they emphasize as being important when designing MOPs. A *bias transformation* produces a bias in the fitness landscape, and is used to produce polynomial bias, flat regions, or other type of bias depending of the values of another variables; *shift transformations* move the location of optimal values, and are used to apply linear shift, or to produce deceptive and multimodal problems; and *reduction transformations*, which combines the values of several decision variables into a single one. They are also used to produce inseparability of the problem (dependencies among the decision variables). For a detailed description of the proposed transformations proposed for this MOP test suite, the interested reader is referred to [91, 41]. In these references the definition of 9 test MOPs can also be found. The whole approach proposed by Huband et al. [91] is very versatile, because it can be easily used to design new test MOPs with desired properties and difficulties.

The simplicity of construction, scalability to any number of decision variables and objectives, knowledge of the shape and location of the resulting Pareto-optimal front, and introduction of controlled difficulties both, for converging to the true Pareto-optimal front and for maintaining a widely distributed set of solutions are the main features of the test problems generators previously described. Because of these features they are very useful in various MOEA research activities such as performance testing of a new MOEA design, comparisons among different MOEAs, and better understanding of the working principles of MOEAs.

The above MOP test functions generators were designed with characteristics that are expected to be present when solving real-world MOPs. However a MOEA that is able to solve any MOP test function or a set of them, has no guarantee of continued effectiveness and efficiency when applied to real-world MOPs. In this sense, and as suggested by Coello et al. [41], real-world applications should be considered for inclusion in any comprehensive MOEA test suite.

Many applications of MOEAs are presented in [41]. The surveyed applications can be numeric, non-numeric or both, and usually have more constraints than numeric MOP tests. It is important to note that many real-world applications, nowadays employ extensive computer simulation tools, i.e. fitness functions are computationally evaluated by simulation codes, for example using Computational Fluid Dynamics, (CFD), and Computational Structural Dynamics/Mechanics (CSD/M), in the aeronautical engineering field. In this thesis, we are interested in solving multi-objective aeronautical engineering problems with the use of MOEAs. We are also interested in designing MOEAs that are more efficient and effective than state-of-the-art MOEAs, when applied to this particular class of problems. For this reason, additional to selecting numerical test MOPs for evaluating our proposed MOEA approaches, we will also select some representative MOPs, which are commonly solved in the aeronautical engineering design research area. In the following sections we give a brief description of the MOP test problems, that were selected for the work reported in this thesis.

6.2 BENCHMARK ADOPTED FROM THE AVAILABLE TEST SUITES

From the different MOPs currently available, and for the purposes of evaluating the convergence characteristics and distribution of solutions, as attained by the different MOEAs proposed in this thesis, we adopted nine numerical MOPs. Five of them are two-objective MOPs, taken from the ZDT test suite, namely ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6. It is important to note that ZDT5, also from the ZDT test suite, is a binary MOP and is not considered here because we are interested in solving continuous MOPs, i.e., MOPs having real-valued decision variables. The other four MOPs contain three objectives and are taken from the DTLZ test suite, namely DTLZ1, DTLZ2, DTLZ3, and DTLZ4. These nine MOPs were selected because they encapsulate a wide variety of characteristics, such as multi-frontality, non-convexity, and discontinuities. All the mentioned characteristics are known to cause difficulties to MOEAs. Also, some problems possess a disconnected Pareto front which make it difficult for MOEAs to reach all regions of the true Pareto-optimal front. An additional reason for choosing these MOPs is because they are commonly adopted by the MOEA community. Next, we present a brief description of each of the numerical MOPs that were selected for the work reported here.

6.2.1 *Definition of the ZDT test problems*

- Test Problem **ZDT1**:

  This MOP is defined by:

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 \\ f_2(\mathbf{x},\mathbf{g}) &= g(\mathbf{x}) \cdot \left(1 - \sqrt{\tfrac{f_1(x)}{g(x)}}\right) \\ g(\mathbf{x}) &= 1 + \tfrac{9}{(n-1)} \cdot \sum_{i=2}^{n} x_i \end{aligned} \tag{26}$$

  The above MOP has a convex Pareto-optimal front and the number of decision variables used is $n = 30$. Their ranges are $x_i \in [0,1], i = 1,\dots,30$. The true Pareto front corresponds to $g(x) = 1$. Figure 30 shows the shape of the true Pareto front for this test problem.

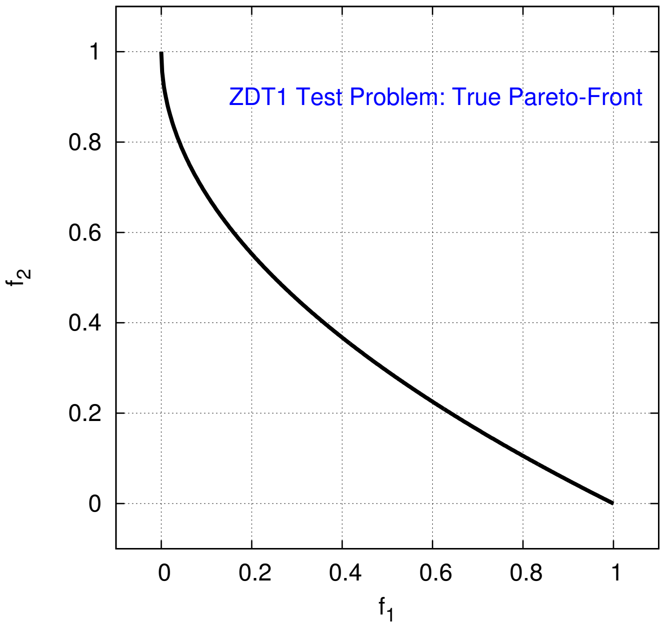


Figure 30: Shape of the true Pareto front for ZDT1.

- Test Problem **ZDT2**:

  This MOP is defined by:

$$
\begin{aligned}
f_1(\mathbf{x}) &= x_1 \\
f_2(\mathbf{x},\mathbf{g}) &= g(\mathbf{x}) \cdot \left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right) \\
g(\mathbf{x}) &= 1 + \frac{9}{(n-1)} \cdot \sum_{i=2}^{n} x_i
\end{aligned}
\tag{27}
$$

  The above MOP has a nonconvex Pareto-optimal front and the number of decision variables used is $n = 30$. Their ranges are $x_i \in [0,1], i = 1,\ldots,30$. The true Pareto front corresponds to $g(x) = 1$. Figure 31 shows the shape of the true Pareto front for this test problem.

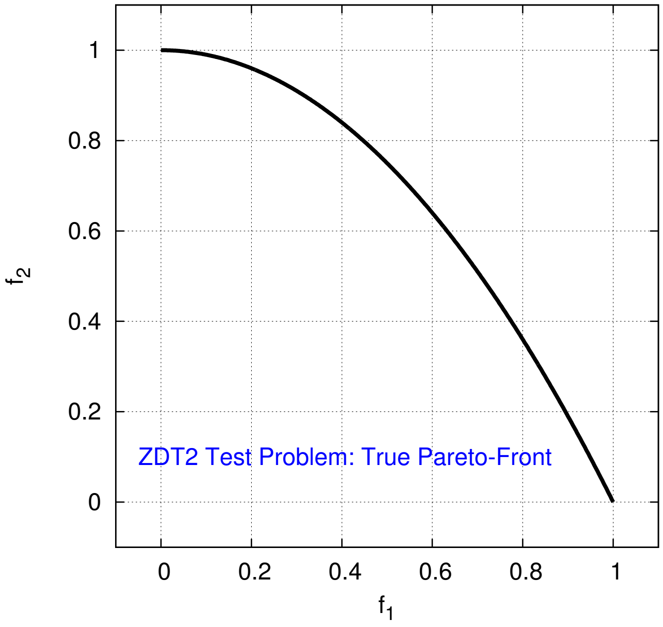


Figure 31: Shape of the true Pareto front for ZDT2.

- Test Problem **ZDT3**:

  This MOP is defined by:

$$
\begin{aligned}
f_1(\mathbf{x}) &= x_1 \\
f_2(\mathbf{x},\mathbf{g}) &= g(\mathbf{x}) \cdot \left(1 - \sqrt{\tfrac{f_1(x)}{g(x)}} - \tfrac{f_1(x)}{g(x)} \cdot \sin(10\pi x_1)\right) \\
g(\mathbf{x}) &= 1 + \tfrac{9}{(n-1)} \cdot \sum_{i=2}^{n} x_i
\end{aligned}
\tag{28}
$$

  This MOP has a Pareto-optimal front that is convex but disconnected, and having five segments. The number of decision variables adopted is $n = 30$, and their ranges are $x_i \in [0,1], i = 1,\ldots,30$. The true Pareto front corresponds to $g(x) = 1$. It is important to note that for this MOP, the $\sin()$ function causes the disconnectedness for the Pareto-front in the objective space, but such a discontinuity in the decision space is not present. Figure 32 shows the shape of the true Pareto front for this test problem.

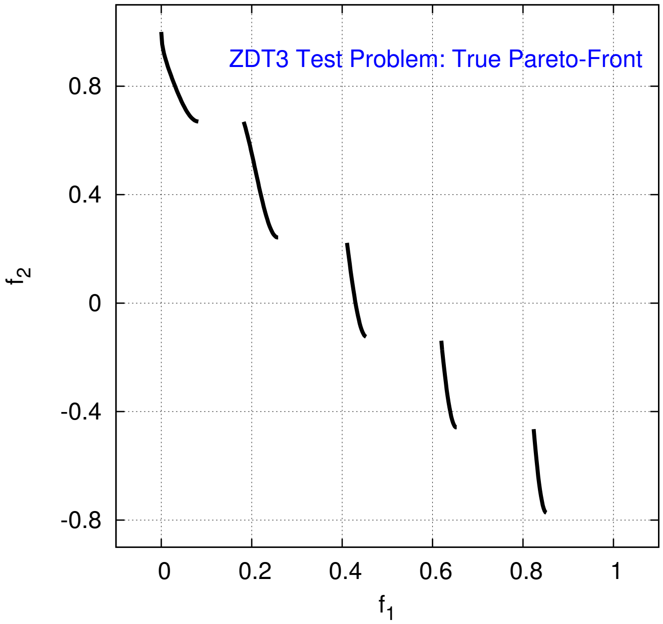


Figure 32: Shape of the true Pareto front for ZDT3.

- Test Problem **ZDT4**:

  This MOP is defined by:

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 \\ f_2(\mathbf{x},\mathbf{g}) &= g(\mathbf{x}) \cdot \left(1 - \sqrt{\frac{f_1(x)}{g(x)}}\right) \\ g(\mathbf{x}) &= 1 + 10 \cdot (n-1) + \sum_{i=2}^{n} (x_i^2 - 10 \cdot \cos(4\pi x_i)) \end{aligned} \tag{29}$$

This MOP has a Pareto-optimal front that is convex in shape an identical to that of **ZDT1**. However, this MOP contains 21^9 local Pareto fronts and therefore, is intended to test a MOEA's ability to deal with multifrontality. The number of decision variables used for this test MOP is $n = 10$, and their ranges are $x_1 \in [0, 1]$, and $x_2, \ldots, x_n \in [-5, 5]$. The global true Pareto-optimal front corresponds to $g(x) = 1$, while the best local Pareto front corresponds to $g(x) = 1.25$. Figure 33 shows the shape of the global true Pareto front for this test problem.

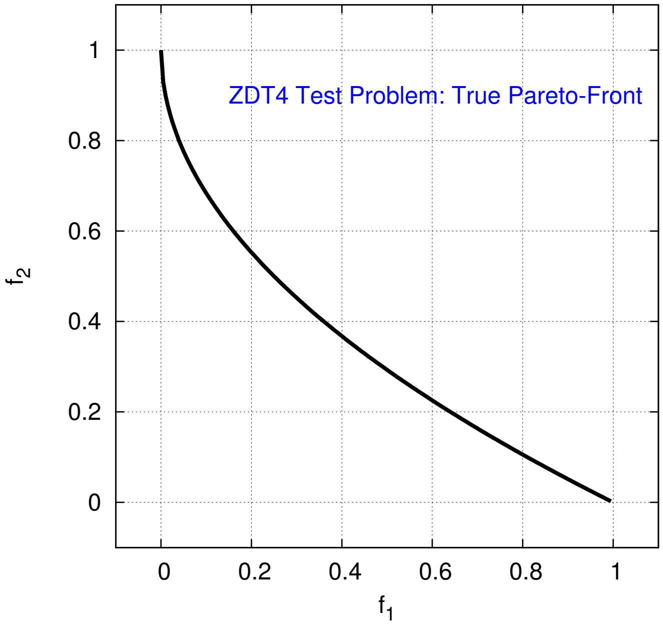


Figure 33: Shape of the true Pareto front for ZDT4.

- Test Problem **ZDT6**:

  This MOP is defined by:

$$\begin{aligned} f_1(\mathbf{x}) &= 1 - \exp(-4x_1) \cdot \sin^6(6\pi x_1) \\ f_2(\mathbf{x}, \mathbf{g}) &= g(\mathbf{x}) \cdot \left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right) \\ g(\mathbf{x}) &= 1 + 9 \cdot \left[\frac{1}{9}\left(\sum_{i=2}^{n} x_i\right)\right]^{0.25} \end{aligned} \tag{30}$$

  This MOP has a Pareto-optimal front that is nonconvex. It includes two major difficulties caused by some defined nonuniformities in the search space. The first difficulty comprises a bias of solutions towards the value of one in f_1. The second difficulty, comprises a difference in the density of solutions in the function fitness landscape, being the lowest near the true Pareto front and highest away from the front. The number of decision variables used is $n = 10$, and their ranges are $x_i \in [0, 1], i = 1, \ldots, 10$. The true Pareto front corresponds to $g(x) = 1$. Figure 34 shows the shape of the true Pareto front for this test problem.

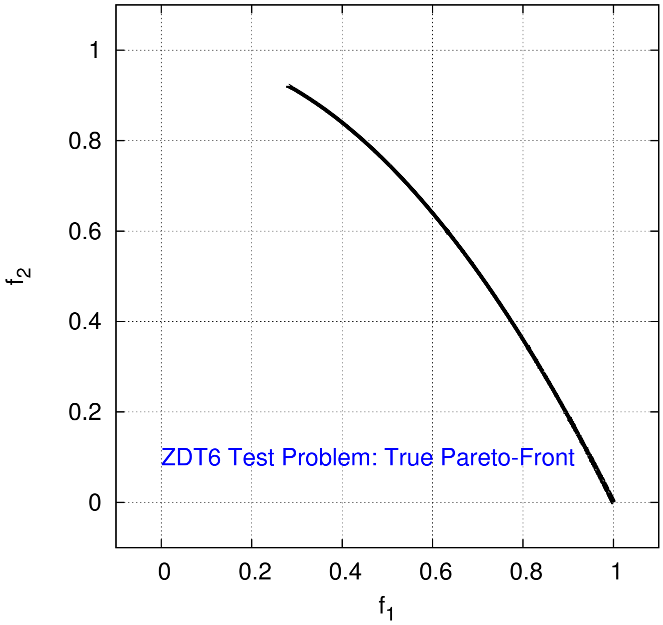


Figure 34: Shape of the true Pareto front for ZDT6.

In table 4, we summarize the definition of the ZDT test problems.

Table 4: Test problems selected from the ZDT test suite [273]

MOP	n	Ranges of the decision variables	Objective function*	Pareto front Characteristics
ZDT1	30	$x = (x_1, \dots, x_n)^T \in [0,1]^n$	$f_1(x) = x_1$ $f_2(x,g) = g(x)\left[1 - \sqrt{\frac{f_1(x)}{g(x)}}\right]$ $g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{(n-1)}$	Convex
ZDT2	30	$x = (x_1, \dots, x_n)^T \in [0,1]^n$	$f_1(x) = x_1$ $f_2(x,g) = g(x)\left[1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right]$ $g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{(n-1)}$	Concave
ZDT3	30	$x = (x_1, \dots, x_n)^T \in [0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)} \sin(10\pi x_1)\right]$ $g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{(n-1)}$	Convex/Disconnected
ZDT4	10	$x = (x_1, \dots, x_n)^T \in [0,1] \times [-5,5]^{(n-1)}$	$f_1(x) = x_1$ $f_2(x,g) = g(x)\left[1 - \sqrt{\frac{f_1(x)}{g(x)}}\right]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right]$	Multifrontal (21^9 local fronts)
ZDT6	10	$x = (x_1, \dots, x_n)^T \in [0,1]^n$	$f_1(x) = 1 - \exp(-4x_1) \cdot \sin^6(6\pi x_1)$ $f_2(x,g) = g(x)\left[1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right]$ $g(x) = 1 + 9 \cdot \left[\frac{\sum_{i=2}^{n} x_i}{9}\right]^{(0.25)}$	Concave/nonuniform

* All functions are intended to be minimized

6.2.2 *Definition of the DTLZs test problems*

The DTLZ test suite is generated by a systematic approach as defined in the above paragraphs. For the following DTLZ test MOPs, we will present the general definition and thereafter the particular MOP instantiation, considering a three objective MOP instantiation.

- Test Problem **DTLZ1**:

  This test MOP is generally defined by

$$\begin{array}{rcl}
\text{Minimize } f_1(x) & = & \frac{1}{2}x_1x_2\ldots x_{M-1}(1+g(x_M)), \\
\text{Minimize } f_2(x) & = & \frac{1}{2}x_1x_2\ldots(1-x_{M-1})(1+g(x_M)), \\
\vdots & & \\
\text{Minimize } f_{M-1}(x) & = & \frac{1}{2}x_1(1-x_2)(1+g(x_M)), \\
\text{Minimize } f_M(x) & = & \frac{1}{2}(1-x_1)(1+g(x_M)), \\
\text{subject to} & & 0 \leqslant x_i \leqslant 1, \text{ for } i = 1,2,\ldots,n
\end{array} \tag{31}$$

  where

$$g(x_M) = 100 \cdot \left(|x_M| + \sum_{x_i \in X_M} \left[(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right] \right) \tag{32}$$

  The functional $g(x_M)$ requires $|x_M| = k$ variables, and the number of variables in the MOP is related to the number of objectives M and k by $n = M + k - 1$. For a three objective MOP, $M = 3$, the instantiation used as test MOP corresponds to

$$\begin{array}{rcl}
f_1(x) & = & \frac{1}{2}x_1x_2(1+g(x)), \\
f_2(x) & = & \frac{1}{2}(1+g(x))x_1(1-x_2), \\
f_3(x) & = & \frac{1}{2}(1+g(x))(1-x_1)
\end{array} \tag{33}$$

and

$$g(x) = 100 \cdot \left(5 + \sum_{i=3}^{n} \left[(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right]\right) \quad (34)$$

The Pareto-optimal solution for this MOP corresponds to the following condition $x_i^* = 0.5$ $(x_i \in X_M)$ and the objective function values lie on the linear hyperplane $\sum_{m=1}^{M} f_m^* = 0.5$. Figure 35 shows the shape of the true Pareto front for the instantiation of DTLZ1 using three objectives, $M = 3$, and $n = 7$ decision variables (a value $k = 5$ has been used). According to the definition of the $g(x)$ functional, this test MOP challenges any MOEA to converge to the true Pareto front, since the search space contains $(11^k - 1)$ local Pareto fronts.

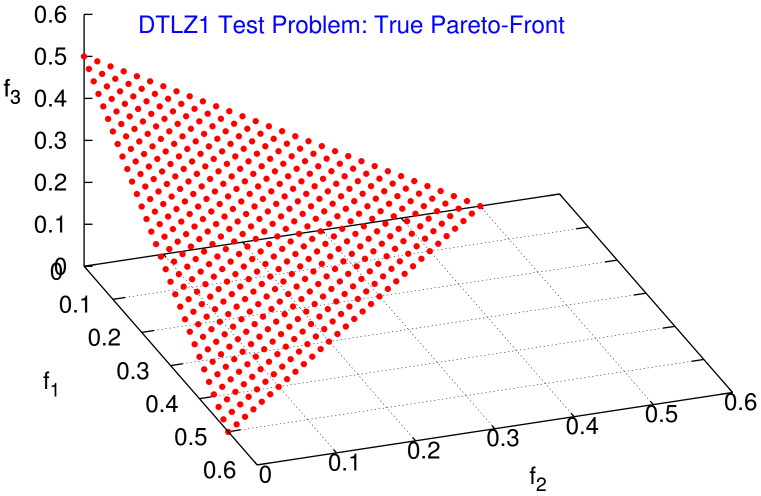


Figure 35: Shape of the true Pareto front for DTLZ1.

- Test Problem **DTLZ2**:

This MOP is generally defined by

$$\begin{aligned}
\text{Minimize } f_1(x) &= (1+g(X_M))\cos(x_1\tfrac{\pi}{2})\ldots\cos(x_{M-1}\tfrac{\pi}{2}),\\
\text{Minimize } f_2(x) &= (1+g(X_M))\cos(x_1\tfrac{\pi}{2})\ldots\sin(x_{M-1}\tfrac{\pi}{2}),\\
&\vdots\\
\text{Minimize } f_M(x) &= (1+g(X_M))\sin(x_1\tfrac{\pi}{2}),\\
\text{Subject to} &\quad 0 \leqslant x_i \leqslant 1,\ \text{for } i=1,2,\ldots,n,\\
\text{where} &\quad g(X_M) = \sum_{x_i \in X_M}(x_i-0.5)^2
\end{aligned} \tag{35}$$

Also for this MOP the functional $g(x)$ requires $|x_M| = k$ variables, and the number of variables in it is related to the number of objectives M by $n = M+k-1$. For a three-objective MOP, $M = 3$, the DTLZ2 instantiation used as test MOP corresponds to

$$\begin{aligned}
f_1(x) &= \cos(x_1\tfrac{\pi}{2})\cos(x_2\tfrac{\pi}{2})(1+g(X_M)),\\
f_2(x) &= \cos(x_1\tfrac{\pi}{2})\sin(x_2\tfrac{\pi}{2})(1+g(X_M)),\\
f_3(x) &= \sin(x_1\tfrac{\pi}{2})(1+g(X_M))
\end{aligned} \tag{36}$$

and

$$g(X_M) = \sum_{i=3}^{n}(x_i-0.5)^2 \tag{37}$$

In this test MOP a value of $|X_M| = k = 10$ is used; therefore, the number of decision variables corresponds to $n = 12$. The Pareto-optimal solution for this test MOP corresponds to the following condition $x_i^* = 0.5,\ (x_i^* \in X_M)$ and all objective function values must satisfy the condition $\sum_{m=1}^{M}(f_m^*)^2 = 1$ which corresponds to a Pareto-optimal front being spherical in shape. Figure 36 shows the shape of the true Pareto front for the instantiation of the DTLZ2 test MOP using $M = 3$ objectives and $n = 12$ decision variables.

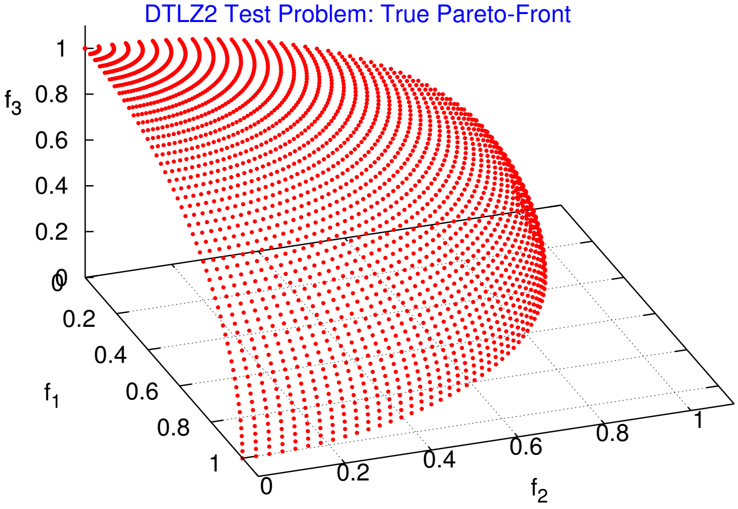


Figure 36: Shape of the true Pareto front for DTLZ2.

- Test Problem **DTLZ3**:

  The definition of this test MOP is the same as that of DTLZ2, but using a different $g(X_M)$ functional. The corresponding functional used in this case is defined by

$$g(X_M) = 100 \cdot \left(10 + \sum_{i=3}^{n} \left[(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right] \right) \quad (38)$$

  The reader might note that this functional resembles the one used in DTLZ1. In this test MOP, a value for $k = 10$ is used, and, therefore, the number of decision variables corresponds to $n = 12$. Analogous to DTLZ1, the defined functional introduces multifrontality in the fitness landscape. The defined DTLZ3 test MOP contains $3k - 1$ local Pareto fronts, which are parallel to the global true Pareto front, and challenges any MOEA to deal with a multifrontality characteristic. The

Pareto-optimal solutions for this test MOP correspond to the following condition $x_i^* = 0.5, \ (x_i^* \in X_M)$ and all objective function values must satisfy $\sum\limits_{m=1}^{M} (f_m^*)^2 = 1$. The true Pareto-optimal front for this test MOP is identical to that of DTLZ2, whose shape is given in Figure 36.

- Test Problem **DTLZ4**:

  The definition of this test MOP is the same as that of DTLZ2, but now, a modified meta-variable mapping is used: $x_i \mapsto x_i^\alpha, \ \alpha > 0$. The parameter suggested in Deb et al. [49] is $\alpha = 100$. This modification allows a dense set of solutions to exist near the $f_M - f_1$ plane. Thus, this test MOP is designed to validate the ability of any MOEA to maintain a good distribution of solutions along the Pareto front approximation. The true Pareto front shape is similar to that given for DTLZ2 and presented in Figure 36.

In Table 5, we summarize the definition of the DTLZ problems MOPs selected here.

Table 5: Test problems selected from the DTLZ test suite [49]

MOP	n	Ranges of the decision variables	Objective function*	Pareto front Characteristics
DTLZ1	7	$x = (x_1, \ldots, x_n)^T \in [0,1]^n$	$f_1(x) = \frac{1}{2} x_1 x_2 (1 + g(x))$ $f_2(x) = \frac{1}{2}(1 + g(x)) x_1 (1 - x_2)$ $f_3(x) = \frac{1}{2}(1 + g(x))(1 - x_1)$ $g(x) = 100(n-2) + 100 \sum_{i=3}^{n} \{(x_i - 0.5)^2 - \cos[20\pi(x_i - 0.5)]\}$	Linear
DTLZ2	12	$x = (x_1, \ldots, x_n)^T \in [0,1]^2 \times [-1,1]^{(n-2)}$	$f_1(x) = (1 + g(x)) \cos\left(\frac{x_1 \pi}{2}\right) \cos\left(\frac{x_2 \pi}{2}\right)$ $f_2(x) = (1 + g(x)) \cos\left(\frac{x_1 \pi}{2}\right) \sin\left(\frac{x_2 \pi}{2}\right)$ $f_3(x) = (1 + g(x)) \sin\left(\frac{x_1 \pi}{2}\right)$ $g(x) = \sum_{i=3}^{n} (x_i - 0.5)^2$	Concave
DTLZ3	12	$x = (x_1, \ldots, x_n)^T \in [0,1]^n$	$f_1(x) = (1 + g(x)) \cos\left(\frac{x_1 \pi}{2}\right) \cos\left(\frac{x_2 \pi}{2}\right)$ $f_2(x) = (1 + g(x)) \cos\left(\frac{x_1 \pi}{2}\right) \sin\left(\frac{x_2 \pi}{2}\right)$ $f_3(x) = (1 + g(x)) \sin\left(\frac{x_1 \pi}{2}\right)$ $g(x) = 100(n-2) + 100 \sum_{i=3}^{n} \{(x_i - 0.5)^2 - \cos[20\pi(x_i - 0.5)]\}$	Concave
DTLZ4	12	$x = (x_1, \ldots, x_n)^T \in [0,1]^n$	$f_1(x) = (1 + g(x)) \cos\left(\frac{x_1^\alpha \pi}{2}\right) \cos\left(\frac{x_2^\alpha \pi}{2}\right)$ $f_2(x) = (1 + g(x)) \cos\left(\frac{x_1^\alpha \pi}{2}\right) \sin\left(\frac{x_2^\alpha \pi}{2}\right)$ $f_3(x) = (1 + g(x)) \sin\left(\frac{x_1^\alpha \pi}{2}\right)$ $g(x) = \sum_{i=3}^{n} (x_i - 0.5)^2$	Concave

* All functions are intended to be minimized

6.3 AERONAUTICAL ENGINEERING PROBLEMS

6.3.1 *Introduction*

In the previous section, we presented the selected numerical MOPs that are to be used to benchmark and to assess the performance of the MOEA approaches proposed in this thesis. As has been pointed out and recommended, for the evaluation and assessment of any new MOEA, some real-world test MOPs should also be included [41]. In this section, we present and describe some real-world test MOPs, found in aeronautical engineering applications. In chapter 4, we have surveyed some research works in the aeronautical engineering design area, where MOEAs have been successfully applied. It is important to remark that the number of applications in this engineering discipline is by far extense. Many more applications of MOEAs are also highlighted in a related publication [10]. In this regard, from the universe of possible real-world applications in aeronautical engineering, and for the purposes of this thesis, we have defined some multi-objective aerodynamic optimization problems, namely multi-objective Aerodynamic Shape Optimization (ASO) problems, dealing with the multi-objective optimization of airfoil shape geometries.

Aerodynamics is the science that deals with the interactions of fluid flows and objects. This interaction is governed by three basic conservation laws: continuity, momentum and energy. These conservation laws are mathematically expressed by means of the *Navier-Stokes* equations [6, 5, 85], which comprise a set of partial differential equations, being unsteady, nonlinear and coupled among them. Aerodynamicists are interested in the effects of this interaction, in terms of the aerodynamic forces and moments, which are the result of integrating the pressure and shear stresses distributions that the flow excerses over the object with which it is interacting. In its early days, aerodynamic designs were done by extensive use of experimental facilities. Nowadays, the use of CFD technology to simulate the flow of complete aircraft configurations, has made it possible to obtain very impressive results with the help of high performance computers and fast numerical algorithms. At the same time, experimental verifications are carried out in scaled flight tests, avoiding many of the inherent disadvantages and extremely high costs of wind tunnel technology. Therefore, we can consider aerodynamics as a mature engineering science.

Current aerodynamic research focuses on finding new designs and/or improving current ones, by using numerical optimization techniques. For this, an automated computer-based search for optimal solutions with respect to a given set of objective functions, is performed in the case of multi-objective optimization. In these cases, the objective functions are defined in terms of aerodynamic coefficients and flow conditions. Additionally, design constraints are included to render the solutions practical or realizable in terms of manufacturing and/or operating conditions. Optimization is accomplished by means of a more or less systematic variation of the design variables which parameterize the shape to be optimized. A variety of optimization algorithms, ranging from gradient-based methods to stochastic approaches with highly sophisticated schemes for the adaptation of the individual mutation step sizes, are currently available. From them, MOEAs have been found to be a powerful but easy-to-use choice [7, 10, 132, 133].

Shape optimization is considered across a wide range of engineering disciplines. Modifications to existing or new geometries are performed iteratively and in a systematic way, with the aim of obtaining the best performing shape for given multiple- objectives and/or requirements. In the aerospace industry, the process of aerodynamic shape optimization is critical during all phases of design [199, 25, 238]. The recurring strategy in aerodynamic shape optimization [172, 72, 77, 101, 102, 97] corresponds to the integration of three distinct tasks:

1. Definition/selection of the geometry parameterization method,
2. Selection of the computational method for the objective function evaluation,
3. Selection and application of an efficient search engine.

Of particular importance to aerodynamic design, are bodies for which, the force parallel and opposite to the direction of motion (drag) is significantly smaller in magnitude to the force component acting normal to the direction of motion (lift). A discipline which has benefited greatly from optimization theory in the recent past is airfoil shape design [120, 119]. Airfoils can be defined as the cross-section of any three-dimensional lifting surface, such as the main wing or a turbine engine blade. Figure 37 shows different airfoil geometries. These geometries were selected to represent a wide range of applications, ranging from low-speed to high-speed aircrafts, each one being efficient for a different flight condition.

Aerodynamic forces and moments for airfoil sections arise from surface pressure and shear stress distributions. Pressure distribution is originated by the flow velocity variation as it traverses the airfoil surface. Shear stresses are produced by fluid viscosity and by the velocity gradients that are generated by

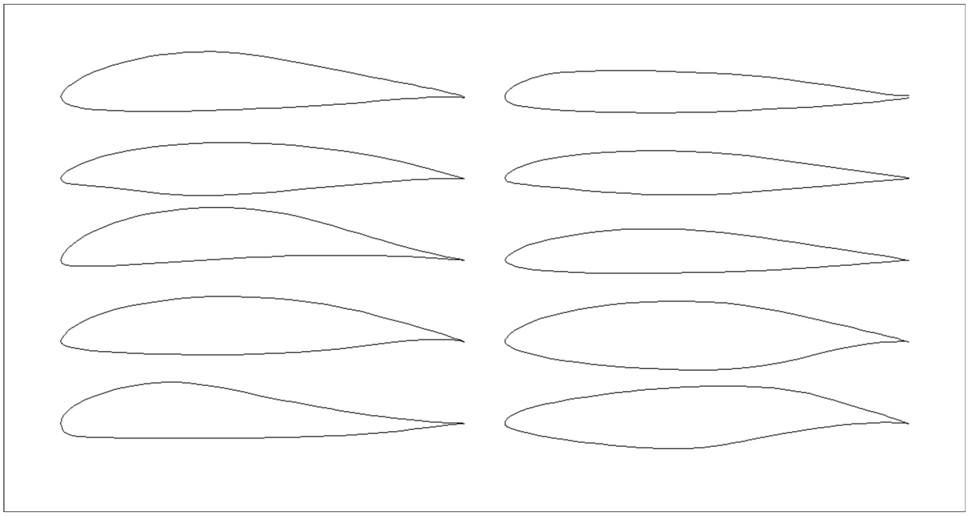


Figure 37: Shape of the airfoils for different aeronautical applications.

the flow retard as it contacts the airfoil surface. In Figure 38, the aerodynamic lift force (L) is mainly the result of integrating the flow pressure distribution acting over the airfoil surface, in a direction normal to the fluid flow velocity (V). Also in Figure 38, the aerodynamic drag force (D) is the result of integrating, both the pressure and shear stress distributions, acting over the airfoil surface, in a direction parallel to the fluid flow velocity (V). We can observe that the pressure is not uniform over the the airfoil surface. Thus, additional to the aerodynamic forces of lift and drag, an aerodynamic moment (M) will also be present as a result of the aerodynamic interaction of the flow with the airfoil shape. This aerodynamic moment governs the pitching motion of the airfoil. The magnitude of the aerodynamic forces and moment are proportional to the flow properties such as density, viscosity, velocity and angle of attack (orientation angle of the flow velocity to the airfoil geometry).

For a given operating condition, variations in the airfoil geometry will result in different flow pressure and shear stress distributions, which in turn will modify the aerodynamic forces and moment. From an aerodynamic point of view, an algorithm in airfoil shape optimization is used to determine the optimal shape in order to maximize/minimize the desired forces and/or objective function. For a general aerodynamic problem, we can infer that one objective will be to maximize the lift force. This condition will allow the aeronautical system to increase its payload capacity. One second objective will be to reduce the drag force, which in turn will reduce the required power for the aeronautical system. Finally, a third objective will be to minimize the aerodynamic

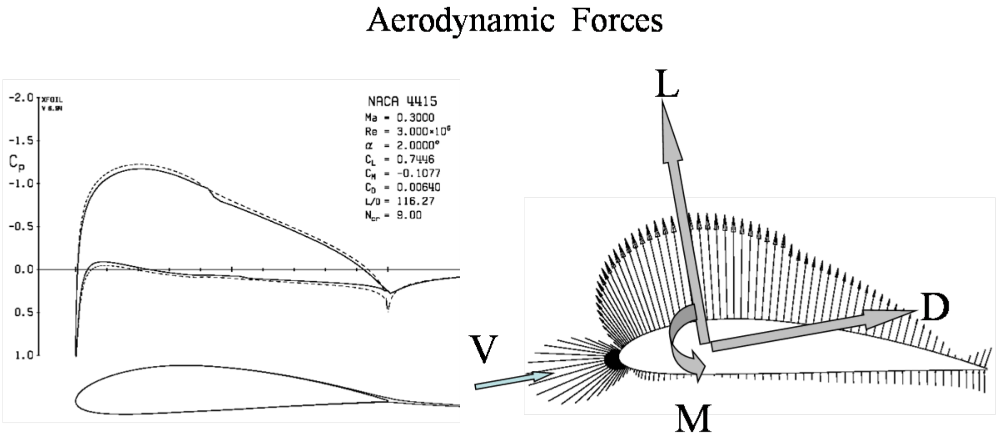


Figure 38: Aerodynamic forces and moments present in an airfoil shape.

moment, which will improve the longitudinal stability for the aeronautical system. In aerodynamic design and analysis it is common to express these forces and moment as scalar coefficients:

$$\text{Coefficient of lift force } = C_L = \frac{L}{\frac{1}{2}\rho V^2 S} \tag{39}$$

$$\text{Coefficient of drag force } = C_D = \frac{D}{\frac{1}{2}\rho V^2 S} \tag{40}$$

$$\text{Coefficient of moment } = C_M = \frac{M}{\frac{1}{2}\rho V^2 S c} \tag{41}$$

where V is the flow velocity, ρ is the fluid density, c is a characteristic length, which in this case corresponds to the airfoil length (or chord), and S is a characteristic surface, which in this case is the airfoil length (or chord) times a unity length ($S = c \cdot 1$). It follows from aerodynamic theory that, for a geometrically similar airfoil shape, at a given flow incidence angle (α), the lift (C_L), drag

(C_D) and moment (C_M) coefficients are a function of the Reynolds number (Re) and Mach number (M).

$$[\ C_l\ ,\ C_d\ ,\ C_m\ ] = f(\alpha, Re, M) \tag{42}$$

$$Re = \frac{\rho \cdot V \cdot L}{\mu} \tag{43}$$

$$M = \frac{V}{a} \tag{44}$$

where μ is the fluid dynamic viscosity, and a is the speed of sound in the fluid. The Reynold number (Re) is the ratio of the inertial to the viscous forces, and quantifies their relative importance in the flow and the given operating condition. The Mach number (M) is a measure of the ratio of the flow velocity with respect to the speed of sound in the fluid, and gives a measure of the compressibility effects in gas fluids such as air.

If we take an airplane as an example of an aeronautical system, for which we are trying to optimize the airfoil shape that will be used as a wing's cross section, depending on the mission phase, either the aerodynamic forces or the aerodynamic moment, or a combination of them, can be used as objectives to optimize. For example, during cruise, designers will try to reduce the drag coefficient at a prescribed lift coefficient in order to optimize fuel economy. For airplanes that are to have a maximum range, the ratio of lift to drag coefficient (C_L/C_D is defined as the aerodynamic efficiency) needs to be maximized [203, 238, 220, 25, 158]. However there are other types of airplane missions such as reconnaissance for which the designer wants to maximize the endurance (endurance is the time an airplane can fly for a given payload and fuel capacity); in this case the ratio C_L^3/C_D^2 (defined as the aerodynamic power efficiency) needs to be maximized [203, 238, 220, 25, 158]. Additional to these objectives, designers will also try to maximize airplane stability, for which, the optimization of the aerodynamic moment coefficient C_M needs to be considered. Finally, all these optimizations must be done at different operating conditions, posing some additional conditions and/or constraints on the operating Reynolds (Re) and Mach (M) numbers.

A design is normally optimized for one flight condition, but good designs are those exhibiting optimal aerodynamic performances over a wide range of flight conditions and different mission segments. This latter condition calls for the application of a multi-objective optimization approach to evaluate the trade-off among the different competing objectives and/or to find the best trade-off design.

At this point we recall the reader that we are trying to define some real-world MOPs, similar to the ones found in the aeronautical engineering discipline. The main purpose of these test MOPs is to benchmark the MOEA approaches proposed as part of the research done in this thesis. Numerical MOPs are easy to construct since only analytical functions need to be considered and evaluated. In the case of real-world MOPs these conditions are quite different. Therefore, aerodynamic shape MOPs are defined similarly to those that were solved in references [229, 17, 61, 167, 165, 166, 179, 157, 90, 197, 198, 191, 46, 256]. An effort has been made to present a diverse set of MOPs containing different Pareto-front shapes, as well as to present different fitness landscapes. For this reason, various combinations of parameters were selected. Among them, we consider variations in the airfoil angle of attack (α), the operating flow conditions in terms of Reynolds number (Re) and Mach number (M), as well as a combination of the aerodynamic coefficients and relations of them, defined as objective functions. For this benchmark, we have designated the test MOPs as *ASO-MOP1* through *ASO-MOP7*. From the seven *ASO-MOPs* defined, four of them have two objective functions, and the other three contain three objective functions. It is worth nothing that, two *ASO-MOPs* include constraints in the operating conditions in terms of aerodynamic coefficients (i.e. aerodynamic coefficient constraints).

Since the test ASO-MOPs deal with airfoil shape geometries, next we describe the geometry parameterization used in them, which in turn defines the dimensionality of the search space. We also describe the CFD simulation used to evaluate the objective functions in these ASO-MOPs. Finally, we give a brief description of the ASO-MOPs definition, and present an approximation of the shape of the Pareto front after a fixed number of objective function evaluations using a MOEA proposed in this thesis. This situation contrasts with numerical test MOPs where the true Pareto front shape is known. In this case, since we do not know a priori the shape of the true Pareto front, we only present an approximation of it.

6.3.2 *Geometry parameterization*

Finding an optimum representation scheme for aerodynamic shape optimization problems is an important step for a successful aerodynamic optimization task. Several options can be used for airfoil shape parameterization. Techniques commonly employed in this area are: Bezier curves or splines [65], Hicks-Henne bump functions [84], and PARSEC airfoil representation [223]. A few general criteria should be considered for an appropriate geometric representation:

(a) The representation used needs to be flexible to describe any general airfoil shape.

(b) The representation also needs to be efficient, so that the parameterization can be achieved with a minimum number of parameters. Inefficient representations may result in an unnecessarily large design space which, in consequence, can reduce the search efficiency of an evolutionary algorithm.

(c) The representation should allow the use of any optimization algorithm to perform local search. This requirement is important for refining the solutions obtained by the global search engine in a more efficient way.

In this case, the PARSEC airfoil representation [223] is used. Figure 39 illustrates the 11 basic parameters used for this representation:

- r_{le} leading edge radius,
- X_{up} location of maximum thickness for the upper surface (extrados),
- X_{lo} location of maximum thickness for lower surface (intrados),
- Z_{up} maximum thickness for upper surface,
- Z_{lo} maximum thickness for lower surface,
- Z_{xxup} curvature for upper surface, at maximum thickness location,
- Z_{xxlo} curvature for lower surfaces, at maximum thickness location,
- Z_{te} trailing edge coordinate,
- ΔZ_{te} trailing edge thickness,

- α_{te} trailing edge direction, and
- β_{te} trailing edge wedge angle.

For the present case, the modified PARSEC geometry representation adopted allows us to define independently the leading edge radius, both for upper and lower surfaces. Thus, 12 variables in total are used. Their allowable ranges are defined in Table 6.

Parameter	Lower bound	Upper bound
r_{leup}	0.0085	0.0126
r_{lelo}	0.0020	0.0040
α_{te}	7.0	10.0
β_{te}	10.0	14.0
Z_{te}	-0.0060	-0.0030
ΔZ_{te}	0.0025	0.0050
X_{up}	0.4100	0.4600
Z_{up}	0.1100	0.1300
Z_{xxup}	-0.90	-0.70
X_{lo}	0.20	0.26
Z_{lo}	-0.0230	-0.0150
Z_{xxlo}	0.05	0.20

Table 6: Parameter ranges for modified PARSEC airfoil representation

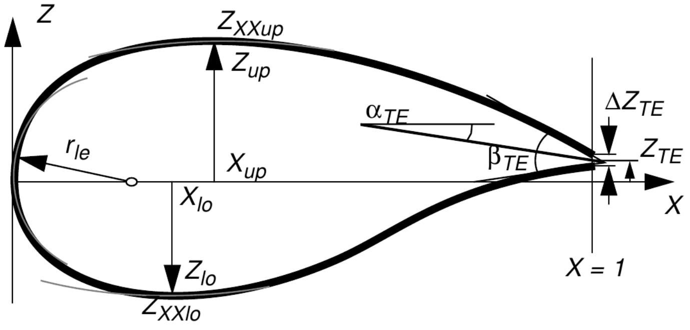


Figure 39: PARSEC airfoil parameterization

The PARSEC airfoil geometry representation uses a linear combination of shape functions for defining the upper and lower surfaces. These linear combinations are given by:

$$Z_{upper} = \sum_{n=1}^{6} a_n x^{\frac{n-1}{2}} \tag{45}$$

$$Z_{lower} = \sum_{n=1}^{6} b_n x^{\frac{n-1}{2}} \tag{46}$$

In the above equations, the coefficients a_n, and b_n are determined as functions of the 12 described geometric parameters, by solving the following two systems of linear equations:

Upper surface:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ X_{up}^{1/2} & X_{up}^{3/2} & X_{up}^{5/2} & X_{up}^{7/2} & X_{up}^{9/2} & X_{up}^{11/2} \\ 1/2 & 3/2 & 5/2 & 7/2 & 9/2 & 11/2 \\ \frac{1}{2}X_{up}^{-1/2} & \frac{3}{2}X_{up}^{1/2} & \frac{5}{2}X_{up}^{3/2} & \frac{7}{2}X_{up}^{5/2} & \frac{9}{2}X_{up}^{7/2} & \frac{11}{2}X_{up}^{9/2} \\ -\frac{1}{4}X_{up}^{-3/2} & \frac{3}{4}X_{up}^{-1/2} & \frac{15}{4}X_{up}^{1/2} & \frac{35}{4}X_{up}^{3/2} & \frac{63}{4}X_{up}^{5/2} & \frac{99}{4}X_{up}^{7/2} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} Z_{te} + \frac{1}{2}\Delta Z_{te} \\ Z_{up} \\ \tan((2\alpha_{te} - \beta_{te})/2) \\ 0 \\ Z_{xxup} \\ \sqrt{r_{leup}} \end{bmatrix} \tag{47}$$

Lower surface:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ X_{lo}^{1/2} & X_{lo}^{3/2} & X_{lo}^{5/2} & X_{lo}^{7/2} & X_{lo}^{9/2} & X_{lo}^{11/2} \\ 1/2 & 3/2 & 5/2 & 7/2 & 9/2 & 11/2 \\ \frac{1}{2}X_{lo}^{-1/2} & \frac{3}{2}X_{lo}^{1/2} & \frac{5}{2}X_{lo}^{3/2} & \frac{7}{2}X_{lo}^{5/2} & \frac{9}{2}X_{lo}^{7/2} & \frac{11}{2}X_{lo}^{9/2} \\ -\frac{1}{4}X_{lo}^{-3/2} & \frac{3}{4}X_{lo}^{-1/2} & \frac{15}{4}X_{lo}^{1/2} & \frac{35}{4}X_{lo}^{3/2} & \frac{63}{4}X_{lo}^{5/2} & \frac{99}{4}X_{lo}^{7/2} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \begin{bmatrix} Z_{te} - \frac{1}{2}\Delta Z_{te} \\ Z_{lo} \\ \tan((2\alpha_{te} + \beta_{te})/2) \\ 0 \\ Z_{xxlo} \\ -\sqrt{r_{lelo}} \end{bmatrix} \tag{48}$$

It is important to note that the geometric parameters r_{leup}/r_{lelo}, X_{up}/X_{lo}, Z_{up}/Z_{lo}, Z_{xxup}/Z_{xxlo}, Z_{te}, ΔZ_{te}, α_{te}, and β_{te} are the actual design variables in the optimization process, and that the coefficients a_n, b_n serve as intermediate variables for interpolating the airfoil's coordinates, which are used by the CFD solver (we used the Xfoil CFD code [56]) for its discretization process.

6.3.3 *Flow solver*

The flow solver adopted in this case was the Xfoil CFD code[1] [56] developed at MIT by Mark Drela. It uses a steady potential panel method coupled with a Karman-Tsien model of the viscous boundary layer. Both, a forced or a free transition of the boundary layer may be specified, as well as varying Reynolds and Mach numbers. The viscous flow solution proceeds by means of an iterative Newton-like procedure that solves a linear system of equations in each step. The size of the system is proportional to the discretization used to represent the upper and lower surfaces of the airfoil[2]. In this case a total of 201 points (comprising both lower and upper airfoil's surfaces) are used to define each design candidate. The CFD flow solution for this tailored airfoil simulation code results in an orders-of-magnitude better performance than typical CFD. The complete evaluation of a single airfoil, involving the solution of 3 flow states for a test MOP with three objectives, strongly depends on the airfoil shape and the flow conditions, but typically requires less than 10 seconds on a single processor. Under reasonably mild flow conditions (i.e., a flow around the airfoil without laminar separation bubbles or large region of turbulent boundary layer separation towards the trailing edge of the airfoil), the accuracy of the computed airfoil characteristics (lift, drag and moment) using the XFOIL CFD code, can compete with more sophisticated CFD codes, or even tunnel measurements. These conditions are usually fulfilled when optimizing for drag, because a well-behaved boundary layer is needed to achieve small drag.

It is important to note that XFOIL solves the flow equations in an iterative manner. As a parameter, the user can define the maximum number of iterations to try to find a CFD flow solution, given an airfoil shape that is being evaluated. For this parameter, and in order not to use too much computational time without having a guarantee of attaining a converged CFD flow solution, we have defined a maximum of 500 iterations to be executed in the XFOIL

[1] In this work, the source code of XFOIL, written in FORTRAN, was adapted for its execution as a direct function call, from the main optimization process.

[2] For the aerodynamic simulation process, XFOIL, uses a discretized airfoil shape. In this work, this shape is provided, by a prescribed distribution of points, both, in the upper and lower airfoil's surfaces, i.e. by evaluating equations (45) and (46) in a uniform distribution of abcissae in the range $x \in [0, 1]$

simulation process. If the CFD code is able to solve the flow equations within this number of prescribed iterations, the aerodynamic forces and moment as reported by the CFD code, are used for evaluating the corresponding objective functions defined in the test ASO-MOPs. It happens, with an certain rate that the XFOIL CFD code is not able to solve the flow equations. In these cases the aerodynamic forces and moment, as reported by the XFOIL CFD code, will dramatically affect and mislead the search process (this can be considered as a situation with a noisy environment) in the optimization. Therefore, we have to check whether the CFD code has attained a converged solution, an we will reject the evaluation for the airfoil shapes not satisfying a converged CFD solution, by penalizing with a high value their corresponding objective functions. The reader might realize that the simulation has to be performed before rejecting and penalizing any design. This situation presents a waste of objective function evaluations for the MOEA.

6.3.4 *Definition of the ASO-MOP test problems*

- Test Problem **ASO-MOP1**:

  ASO-MOP1 considers two objective functions: (i) to minimize the drag force coefficient, and (ii) to maximize the lift force coefficient. Both objectives are evaluated at a fixed angle of attack (α), and at the same flow conditions in terms of flow velocity, given by the Reynolds (Re) and Mach (M) numbers. In this case, only one call to the CFD simulation code is needed to evaluate both objectives. Note that all ASO-MOPs will be mathematically defined as minimization MOPs (i.e., all objectives are to be minimized, therefore the second objective function in this test case, which is the maximization of the lift force coefficient, is reformulated as a minimization problem, by substracting it from a constant[3]. This constant can be selected as the maximum lift coefficient expected for this type of aerodynamic problems. In this case, a value of 2.0 has been selected). The mathematical definition for the ASO-MOP1 is:

  $$\min(C_d) \ @ \ \alpha = 0.0^o, \mathrm{Re} = 4.0 \times 10^6, M = 0.2$$

  $$\min(2.0 - C_l) \ @ \ \alpha = 0.0^o, \mathrm{Re} = 4.0 \times 10^6, M = 0.2$$

[3] Other option could be to multiply the value of the lift coefficient by −1, but in this case, the objective function could have negative values. We have adopted a convention in the ASO-MOPs definition by which the objective functions take only positive values

Figure 40 shows an approximation of the Pareto front obtained by the *MODE-LD+SS* MOEA proposed in this thesis, at a level in which 5,000 Objective Function Evaluations (OFEs) have been performed[4]. From this figure, we can observe that the shape of the Pareto front has a mixed convex-concave shape. Also and different to the case of the numerical MOPs, where the shape of the Pareto front is continuous, in this ASO-MOP, we can anticipate that the shape of the Pareto front is not continuous, Nonetheless, we have used a line to connect every solution in the approximation of the Pareto front to better appreciate its shape [5], but this should not be interpreted as the Pareto front being continuous.

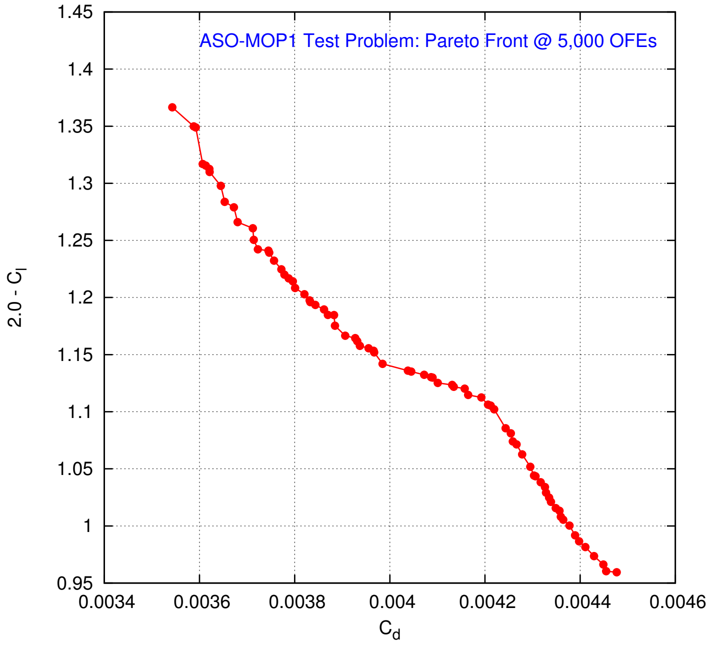


Figure 40: Shape of the approximated Pareto front for ASO-MOP1.

- Test Problem **ASO-MOP2**:

[4] It is important to remark that this number of OFEs is quite high for the type of real-world problem considered here, since in most cases, on the order of 1,000 and up to 2,000 OFEs are allowed in real industrial scenarios, where more sophisticated CFD simulation codes are used, requiring in consequence, computational efforts that are several orders of magnitude higher. An example of this situation is reported in [15], where on the order of 2000 hrs. of computing time are used for the multi-objective re-design of a transonic turbine rotor blade, in a four-processors workstation, using 2000 OFEs.

[5] This also applies for the other bi-objective ASO-MOPs described in this section.

ASO-MOP2 considers two objective functions: (i) to maximize the aerodynamic efficiency, i.e., the lift to drag forces ratio C_l/C_d, and (ii) to minimize the absolute value for the aerodynamic moment coefficient. Both objectives are evaluated at a fixed angle of attack (α), and at the same flow conditions in terms of flow velocity, given by the Reynolds (Re) and Mach (M) numbers. In this case, only one call to the CFD simulation code is needed to evaluate both objectives. Note also that for this ASO-MOP, the first objective is reformulated to a minimization case, by considering the inverse of the lift to drag forces ratio, i.e. by minimizing the C_d/C_l ratio. Regarding the aerodynamic moment coefficient, for some airfoil designs, being evaluated, it can happen that its value could be negative. In this case, we consider the moment coefficient value squared as the objective to minimize. The mathematical definition for the ASO-MOP2 is:

$\min(C_d/C_l)$ @ $\alpha = 4.0^o$, $Re = 2.0 \times 10^6$, $M = 0.1$

$\min(C_m^2)$ @ $\alpha = 4.0^o$, $Re = 2.0 \times 10^6$, $M = 0.1$

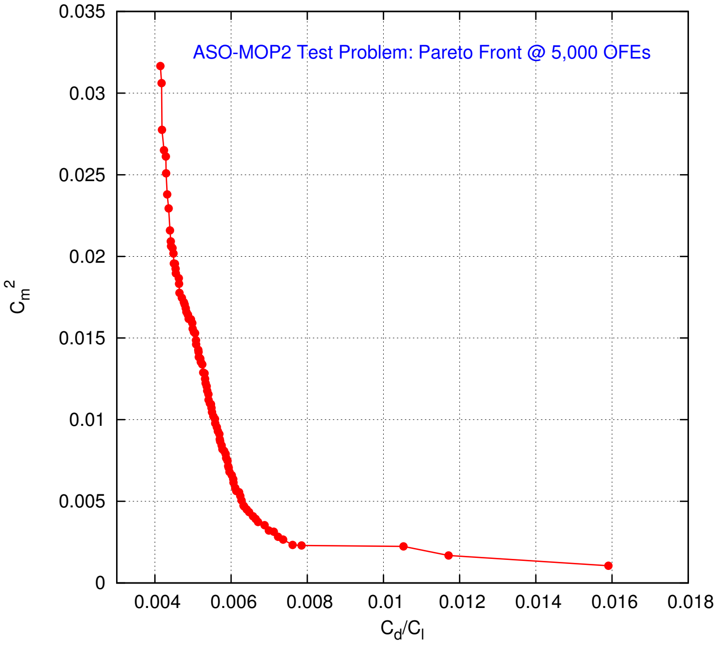


Figure 41: Shape of the approximated Pareto front for ASO-MOP2.

Figure 41 shows an approximation of the Pareto front obtained by the *MODE-LD+SS* MOEA proposed in this thesis, at a level in which 5,000

OFEs have been performed. From this figure, we can observe that the shape of the Pareto front is convex. We can also anticipate that the shape of the Pareto front is not continuous towards its lower right part. Nonetheless, we have used a line to connect every solution in the approximation of the Pareto front to better appreciate its shape.

- Test Problem **ASO-MOP3**:

  ASO-MOP3 considers two objective functions: (i) to maximize the aerodynamic efficiency, i.e., the lift to drag forces ratio C_l/C_d, and (ii) to maximize aerodynamic power efficiency, i.e., C_l^3/C_d^2. In this case each objective is evaluated at different operating flow conditions, in terms of angle of attack (α), and flow velocity, given by the Reynolds (Re) and Mach (M) numbers. The aim is to reflect a real case in which we try to obtain the trade-off in this ASO-MOP at two different design conditions. In consequence, for this case we will require, for each airfoil design candidate, to do two calls to the CFD simulation code, duplicating the computational effort for the objective function evaluations. Note also that for this ASO-MOP, both objective are reformulated to a minimization case, by considering the inverse of the aerodynamic forces ratios. i.e., by minimizing the C_d/C_l ratio, and by minimizing the C_d^2/C_l^3 ratio. The mathematical definition for the ASO-MOP3 is:

  $\min(C_d/C_l)$ @ $\alpha = 1.0^o$, $Re = 3.0 \times 10^6$, $M = 0.3$

  $\min(C_d^2/C_l^3)$ @ $\alpha = 5.0^o$, $Re = 1.5 \times 10^6$, $M = 0.15$

  Figure 42 shows an approximation of the Pareto front obtained by the MODE-LD+SS MOEA proposed in this thesis, at a level in which 5,000 OFEs have been performed. It is important to note that in this case, the number of OFEs should be interpreted as the number of different airfoil designs being evaluated, since each objective function requires a call to the CFD simulation code. From this figure, we can observe that the shape of the Pareto front has a general convex shape, but there are some small regions where the shape can be considered as concave. We can also anticipate that the shape of the Pareto front is not continuous. Nonetheless, we have used a line to connect every solution in the approximation of the Pareto front to better appreciate its shape.

- Test Problem **ASO-MOP4**:

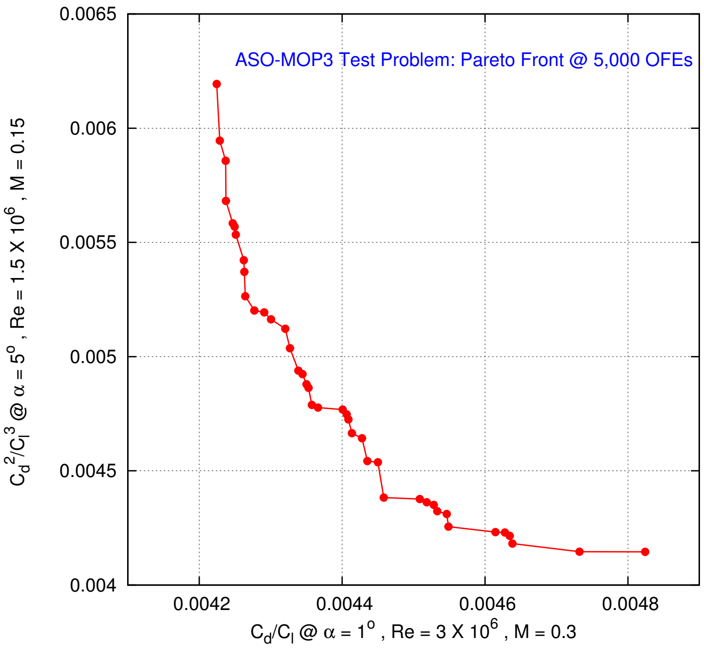


Figure 42: Shape of the approximated Pareto front for ASO-MOP3.

ASO-MOP4 considers three objective functions: (i) to minimize the drag force coefficient, (ii) to maximize the lift force coefficient, and (iii) to minimize the absolute value for aerodynamic moment coefficient. All three objectives are evaluated at a fixed angle of attack (α), and at the same flow conditions in terms of flow velocity, given by the Reynolds (Re) and Mach (M) numbers. In this case, only one call to the CFD simulation code is needed to evaluate the three objectives. Similar to the cases for the maximization of the lift force coefficient in ASO-MOP1, and the maximization of the absolute value for the aerodynamic moment coefficient in ASO-MOP2, these objectives are reformulated in a similar manner. The mathematical definition for ASO-MOP4 is:

$$\min(C_d) \text{ @ } \alpha = 4.0^o, Re = 3.0 \times 10^6, M = 0.3$$

$$\min(2.0 - C_l) \text{ @ } \alpha = 4.0^o, Re = 3.0 \times 10^6, M = 0.3$$

$$\min(C_m^2) \text{ @ } \alpha = 4.0^o, Re = 3.0 \times 10^6, M = 0.3$$

Figure 43 shows an approximation of the Pareto front obtained by the *MODE-LD+SS* MOEA proposed in this thesis, at a level in which 12,000 OFEs have been performed. From this figure, we can observe that the shape of the Pareto front is convex.

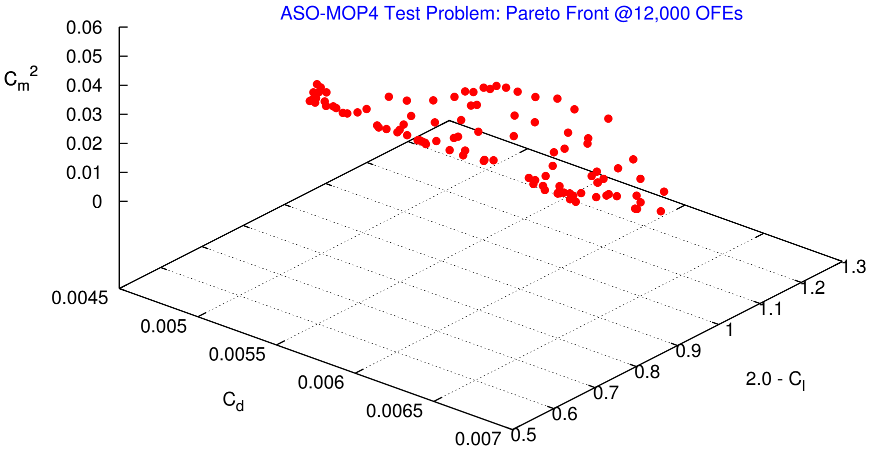


Figure 43: Shape of the approximated Pareto front for ASO-MOP4.

- Test Problem **ASO-MOP5**:

ASO-MOP5 also considers three objective functions: (i) to maximize the aerodynamic efficiency, i.e., to maximize the C_l/C_d ratio, (ii) to maximize aerodynamic power efficiency, i.e., C_l^3/C_d^2 at a first operating condition, and (iii) to maximize aerodynamic power efficiency, i.e., C_l^3/C_d^2 at a second operating condition. In this case, each objective is evaluated at different operating flow conditions, in terms of angle of attack (α), and flow velocity, given by the Reynolds (Re) and Mach (M) numbers. The aim is to reflect a real case in which we try to obtain the trade-off in this ASO-MOP at three different design conditions. In consequence, for this case we will require, for each airfoil design candidate, to do three calls to the CFD simulation code, thus multiplying by three the computational effort. Note also that for this ASO-MOP, the three objective are reformulated to a minimization case, by considering the inverse of the aerodynamic forces ratios. i.e. by minimizing the C_d/C_l ratio for the first objective, and by minimizing the C_d^2/C_l^3 ratio fro the second and third objectives. The mathematical definition for ASO-MOP5 is:

$\min(C_d/C_l)$ @ $\alpha = 1.0^o$, $Re = 4.0 \times 10^6$, $M = 0.3$

$$\min(C_d^2/C_l^3) \text{ @ } \alpha = 3.0^o, \mathrm{Re} = 3.0 \times 10^6, M = 0.3$$

$$\min(C_d^2/C_l^3) \text{ @ } \alpha = 5.0^o, \mathrm{Re} = 2.0 \times 10^6, M = 0.3$$

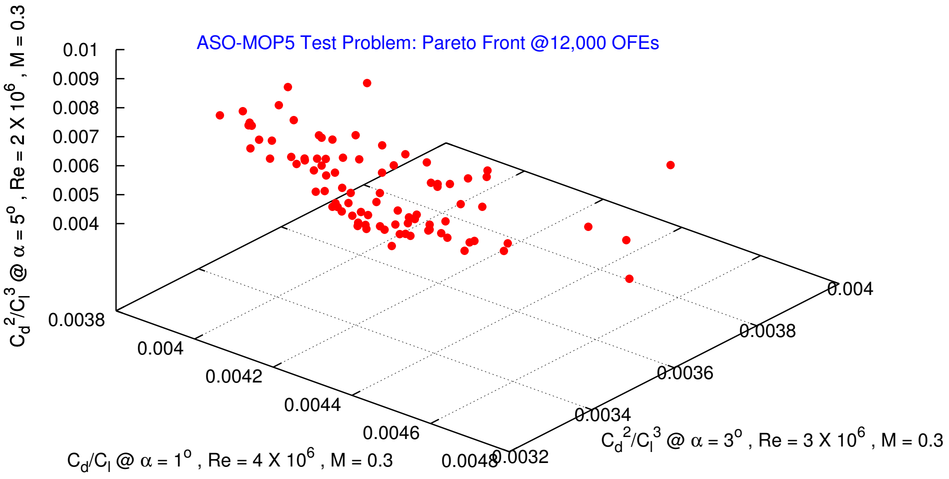


Figure 44: Shape of the approximated Pareto front for ASO-MOP5.

Figure 44 shows an approximation of the Pareto front obtained by the MODE-LD+SS MOEA proposed in this thesis, at a level in which 12,000 OFEs have been performed. From this figure, we can observe that the shape of the Pareto front is convex.

- Test Problem **ASO-MOP6**:

ASO-MOP6 considers two objective functions: (i) to maximize the aerodynamic efficiency, i.e., to maximize the C_l/C_d ratio at given operating condition and subject to a specified lift force coefficient, and (ii) to maximize the aerodynamic power efficiency, i.e., the C_l^3/C_d^2 ratio at a second operating condition, and subject also to a specified lift force coefficient. In this case, each objective is evaluated at different operating flow conditions, in terms of flow velocity, given by the Reynolds (Re) and Mach (M) numbers. Unlike to the other ASO-MOPs previously defined, in which the angle of attack has been defined a priori, in this case, instead, a lift force coefficient is defined for each operating condition and objective, and then, the corresponding angle of attack needs to be calculated. This latter condition indicates that this is a constrained problem. For this, the flow solver, given the design candidate geometry, solves the flow equations with a constraint on the C_l value, i.e., it additionally determines the operating angle of attack α. The aim of this ASO-MOP is to better reflect a real case in which we try to obtain the trade-off in this ASO-MOP at two different design conditions with constraints. In consequence, we will require, for each airfoil design candidate, to do two calls to the CFD simulation code, thus duplicating the computational effort for the objective function evaluations, but also increasing the computational cost, since the flow solver treats the constraints on C_l by using an iterative process. Note also that for this ASO-MOP, the two objectives are reformulated to a minimization case, by considering the inverse of the aerodynamic forces ratios. i.e., by minimizing the C_d/C_l ratio for the first objective, and by minimizing the C_d^2/C_l^3 ratio for the second objective. The mathematical definition for the ASO-MOP6 is:

$\min\,(C_d/C_l)$ @ $Re = 2.04 \cdot 10^6$, $M = 0.12$ and subject to $C_l = 0.63$,

$\min\,(C_d^2/C_l^3$ @ $Re = 1.29 \cdot 10^6$, $M = 0.08$ and subject to $C_l = 1.05$

Figure 45 shows an approximation of the Pareto front obtained by the *MODE-LD+SS* MOEA proposed in this thesis, at a level in which 5,000 OFEs have been performed. It is important to note that similar to ASO-MOP3, the number of OFEs should be interpreted as the number of different airfoil designs being evaluated, since each objective function requires a call to the CFD simulation code. From this figure, we can observe that the shape of the Pareto front has a general convex shape, with small concave portions, and a steepest tendency in the upper left part of the approximated Pareto front. Note that we have used a line to connect every solution in the approximation of the Pareto front to better appreciate its shape.

- Test Problem **ASO-MOP7**:

  ASO-MOP7 considers three-objective functions: (i) to maximize the aerodynamic efficiency, i.e., to maximize the C_l/C_d ratio at a given operating condition and subject to a specified lift force coefficient, (ii) to maximize the aerodynamic efficiency, i.e., the C_l/C_d ratio a at a second operating condition, and subject also to a specified lift force coefficient, and (iii) to maximize the aerodynamic power efficiency, i.e., the C_l^3/C_d^2

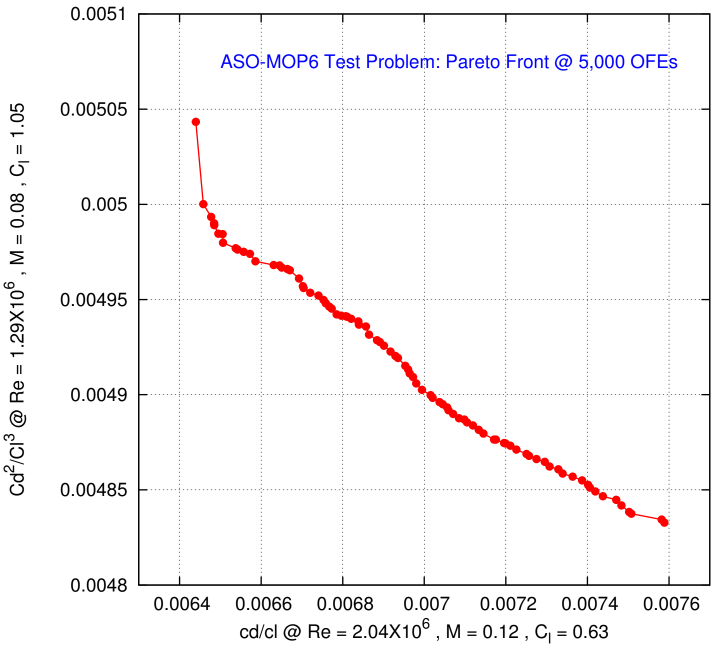


Figure 45: Shape of the approximated Pareto front for ASO-MOP6.

ratio at a third operating condition, and subject also to a specified lift force coefficient. In this case, also each objective is evaluated at different operating flow conditions, in terms of flow velocity, given by the Reynolds (Re) and Mach (M) numbers. This is also a constrained problem The aim of this ASO-MOP is to better reflect a real case in which we try to obtain the trade-off in this ASO-MOP at three different design conditions with constraints. In consequence, we will require, for each airfoil design candidate, to do three calls to the CFD simulation code, multiplying by three the computational effort for the objective function evaluations, and also increasing the computational cost, since the flow solver treats the constraints on C_l by using an iterative process for each objective function. Note also that for this ASO-MOP, the three objectives are reformulated to a minimization case, by considering the inverse of the aerodynamic forces ratios. i.e. by minimizing the C_d/C_l ratio for the first objective, and by minimizing the C_d^2/C_l^3 ratio for the second and third objectives. The mathematical definition for the ASO-MOP7 is:

$$\min\ (C_d/C_l)\ @\ Re = 2.04 \cdot 10^6,\ M = 0.12 \text{ and subject to } C_l = 0.63,$$

$$\min\ C_d/C_l\ @\ Re = 1.63 \cdot 10^6,\ M = 0.10 \text{ and subject to } C_l = 0.86,$$

$$\min (C_d^2/C_l^3 \ @ \ \mathrm{Re} = 1.29 \cdot 10^6, \mathrm{M} = 0.08 \text{ and subject to } C_l = 1.05$$

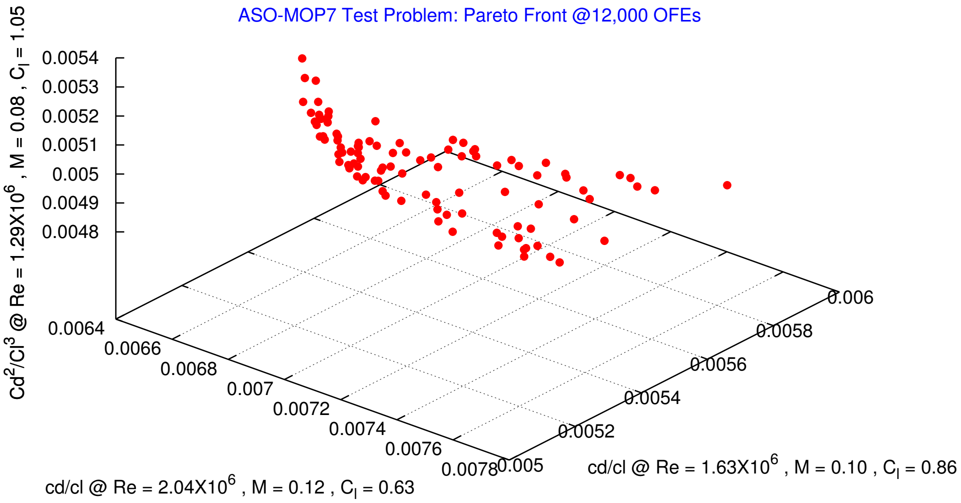


Figure 46: Shape of the approximated Pareto front for ASO-MOP7.

Figure 46 shows an approximation of the Pareto front obtained by the *MODE-LD+SS* MOEA proposed in this thesis, at a level in which 12,000 OFEs have been performed. From this figure, we can observe that the shape of the Pareto front is convex.

6.4 PERFORMANCE MEASURES

In order to perform a quantitative comparison of results between different MOEAs, there are mainly two goals we want to measure: (1) MOEA's ability to obtain solutions as close to the Pareto optimal solutions as possible (closeness to the true Pareto front) and (2) MOEA's ability to obtain solutions as diverse as possible along the Pareto front (good distribution of the solutions). We can consider that these goals are independent from each other, and use different performance measures to deal with one or both of the goals. In general, it does not exist a single performance measure that can indicate the superiority of one MOEA over another one in these two aspects. In consequence, there is a clear need to have at least two performance measures for adequately evaluate both goals (convergence and diversity) of a MOEA.

In this work, two performance measures were adopted in order to assess our results: *Hypervolume (Hv)* and *Two Set Coverage (C-Metric)*. This choice was driven, mainly, because in the ASO-MOPs we do not know, in advance, the exact geometry and position for the true Pareto front, i.e., we lack of a reference Pareto set to compare to, and this condition precludes, for the ASO-MOPs, the use of the *generational distance (GD)*, or the *inverse generational distance (IGD)* measures, which are used commonly by the MOEA community, to assess and compare MOEA results. A brief description of the selected measures is presented next.

6.4.1 *Hypervolume (Hv):*

Given a Pareto approximation set PF_{known}, and a reference point in objective space z_{ref}, this performance measure estimates the *Hypervolume* attained by it. Such hypervolume corresponds to the non-overlapping volume of all the hypercubes formed by the reference point (z_{ref}) and every vector in the Pareto set approximation. This is mathematically defined as:

$$HV = \{\cup_i vol_i | vec_i \in PF_{known}\}$$

vec_i is a nondominated vector from the Pareto set approximation, and vol_i is the volume for the hypercube formed by the reference point and the nondominated vector vec_i. Here, the reference point (z_{ref}) in objective space for the 2-objective MOPs was set to (1.05,1.05); for DTLZ1 it was set to (0.6,0.6,0.6), and to (1.05,1.05,1.05) for DTLZ2, DTLZ3 and DTLZ4. This performance measure is Pareto compliant [275, 271], and is used to assess both convergence and spread of the solutions along the approximated Pareto front. High values indicate that the solutions are closer to the true Pareto front and that they cover a wider extension of it.

6.4.2 *Two Set Coverage (C-Metric):*

This performance measure is also Pareto compliant, and estimates the coverage proportion, in terms of percentage of dominated solutions, between two sets.

Given the sets A and B, both containing only nondominated solutions, the C-Metric is mathematically defined as:

$$C(A,B) = \frac{|\{u \in B | \exists v \in A : v \, dominates \, u\}|}{|B|}$$

This performance measure indicates the portion of vectors in B being dominated by any vector in A. In the present work this measure is used in two different ways. In the first, the set A is the true Pareto front, which is known for the ZDT and DTLZ test functions used; therefore, the C-Metric can be considered as a measure for the ability of the algorithm to find solutions that are nondominated with respect to the Pareto optimal set (i.e., solutions that also belong to the Pareto optimal set). In the second way, sets A and B correspond to two different Pareto approximations, as obtained by two different algorithms. Therefore, the C-Metric is used for pairwise comparisons between the two algorithms used. This performance measure is used for comparing all algorithms used for solving the ZDTs, DTLZs, as well as the aerodynamic shape optimization problems, in which the true Pareto front is unknown.

7 ASSESSMENT OF THE PROPOSED APPROACHES

In this chapter, we present the results of applying the two MOEAs proposed in this thesis, namely *MODE-LD+SS* and *pMODE-LD+SS*, for solving the selected test MOPs, defined in Chapter 6. Since in this chapter we are interested in assessing the general performance of the proposed MOEAs, in all cases 32 independent runs were executed, from which statistics were gathered for the performance measures selected. Additionally, and for comparison purposes, other state-of-the-art MOEAs were used to compare their performance against that of our proposed approaches.

7.1 ASSESSMENT OF MODE-LD+SS WITH NUMERICAL PROBLEMS

In this section we present the application of *MODE-LD+SS* to the nine selected numerical MOPs; ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ1, DTLZ2, DTLZ3, and DTLZ4. In this case we also present the comparisons with respect to the results obtained by *NSGA-II*, *SPEA2*, and *MOEA/D* algorithms. These three latter algorithms are considered as state-of-the-art MOEAs, and their main characteristics were described in Section 3.2.2 (in page 29) for *NSGA-II* and *SPEA2*, and in Section 3.3 (in page 37) for *MOEA/D*. The first two algorithms correspond to Pareto-based MOEAs while the third is based on decomposition. All of them have been successfully used to solve the numerical test MOPs indicated before.

7.1.1 *Parameter settings*

The parameters used in the experiments for the different algorithms adopted were set as follows. The common parameters for all algorithms comprise the population size N and maximum number of generations $GMAX$. These were set to $N = 100$ for all the bi-objective MOPs and $N = 300$ for all the MOPs having three objectives. We adopted $GMAX = 150$ for all MOPs, except for ZDT4 and DTLZ3, in which we used $GMAX = 200$. As for specific parameters of each algorithm, for both, the *NSGA-II* and the *MOEA/D* algorithms, some common parameters used were: *(a) crossover probability* $p_c = 1.0$; *(b) mutation probability* $p_m = 1/NVARS$ (where $NVARS$ corresponds to the number of decision variables in each numerical MOP as defined in previous chapter); and, *(c) distribution index for mutation* $\eta_m = 20$. For the *NSGA-II*, the distribution index for crossover $\eta_c = 15$; while for the MOEA/D, the distribution index for crossover $\eta_c = 20$. The previous choices are based on values reported in corresponding publications where the MOEAs were presented. *SPEA2* was taken from *PISA* [21, 22], and was used with the parameters defined therein:

Parameter	Value
individual_mutation_probability =	1.0
individual_recombination_probability =	1.0
variable_mutation_probability =	1/NVARS
variable_swap_probability =	0.5
variable_recombination_probability =	0.5
distribution index for crossover η_c =	15
distribution index for mutation η_m =	20
use_symmetric_recombination =	0

For our *MODE-LD+SS*, the parameter values to be used for F, CR, and NB, were tuned by experiments with the ZDTs and DTLZs MOPs. For each MOP, 32 different runs were executed for the following combination of parameter ranges: $F \in [0.1, 0.9]$, $CR \in [0.1, 0.9]$, in both cases increments of 0.1 were used, $NB = 1, 3, 5, 10, 20$, $NP = 50, 100$ and $NP = 210, 300$, for two-objective and three MOPs respectively. Due to space limitations, results from these experiments are not shown. The associated parameters that were selected are the following[1]: Scaling factor $F = 0.5$ for all MOPs; crossover rate $CR = 0.5$ for all MOPs, except for ZDT4 and DTLZ3, where we adopted $CR = 0.3$; Neighborhood size $NB = 5$ for all MOPs, except for ZDT4, where $NB = 1$ was used.

7.1.2 *Results and discussion*

Table 7 shows the results obtained for the hypervolume (Hv) measure for all numerical MOPs, and for the four algorithms compared in this thesis. The statistics presented for the hypervolume were obtained as average values from 32

[1] This combination of parameters showed good performance for the hypervolume measure for most of the ZDT and DTLZ MOPs.

Test Function	ALGORITHM			
	NSGA	SPEA2	MOEA/D	MODE-LD+SS
	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)
ZDT1	0.757357 (0.000928)	0.761644 (0.000556)	0.749964 (0.009777)	**0.763442** (0.000112)
ZDT2	0.422221 (0.001263)	0.321971 (0.171286)	0.387237 (0.061361)	**0.430358** (0.000141)
ZDT3	0.611480 (0.008038)	0.615533 (0.000416)	0.608377 (0.015638)	**0.616381** (0.000150)
ZDT4	0.217626 (0.192914)	0.287359 (0.188726)	**0.745887** (0.009983)	0.741770 (0.058697)
ZDT6	0.345949 (0.008772)	0.392697 (0.002336)	0.397720 (0.002886)	**0.411054** (0.000003)
DTLZ1	0.165918 (0.026090)	**0.191437** (0.000248)	0.188726 (0.000371)	0.187445 (0.000347)
DTLZ2	0.571146 (0.001942)	**0.590833** (0.000900)	0.578679 (0.001460)	0.581028 (0.001193)
DTLZ3	0.000000 (0.000000)	0.467163 (0.148867)	0.568895 (0.007220)	**0.581129** (0.003303)
DTLZ4	0.572327 (0.002537)	**0.590942** (0.000978)	0.579301 (0.001550)	0.578038 (0.001840)

Table 7: Comparison of the Hypervolume Metric (Hv) for all the algorithms used for solving numerical MOPs

independent runs for each MOP and for each algorithm. The reference point (z_{ref}) in objective space for computing the Hv performance measure was set to *(1.05,1.05)* for all the 2-objective MOPs. For DTLZ1 it was set to *(0.6,0.6,0.6)*, and to *(1.05,1.05,1.05)* for DTLZ2, DTLZ3 and DTLZ4. High values of the hypervolume indicate that the solutions are closer to the true Pareto front and that they cover a wider extension of it. From Table 7, it can be observed that, with respect to the Hv performance measure, *MODE-LD+SS* outperforms *NSGA-II* and *SPEA2*, in all the bi-objective MOPs. It also outperforms *MOEA/D* in four (ZDT1, ZDT2, ZDT3, and ZDT6) of five bi-objective MOPs. In the case of the 3-objective MOPs, *SPEA2* attains the best results for the Hv measure in three MOPs (DTLZ1, DTLZ2, and DTLZ4), while *MODE-LD+SS* attains the best result in DTLZ3. However, our proposed *MODE-LD+SS* obtained values very close to those of *SPEA2* in DTLZ1, DTLZ2 and DTLZ4 and better values in DTLZ2 and DTLZ3, as compared to those of *MOEA/D*. In all cases, our proposed approach significantly outperforms *NSGA-II*.

As previously indicated, one important aspect of numerical MOPs is that the shape and location of the true Pareto-front is known in advance. This condition can be used to better evaluate the convergence of any MOEA solving numerical MOPs. For evaluating the convergence of the MOEAs used in solving the test MOPs proposed, we use the *C-Metric* performance measure in two different ways. In the first way we measure the convergence towards the true Pareto-front, i.e., C-Metric($\mathsf{PF}_{\mathsf{true}}$,Algorithm). In this way we are measuring the ability of any MOEA to generate Pareto optimal solutions which are non-dominated with respect to the true Pareto front, and statistics are presented based on the 32 independent runs executed. The second way of using the *C-Metric* performance measure is for comparing the convergence performance between pairs of MOEAs, i.e., C-Metric(Algorithm_A,Algorithm_B). In this case, we are measuring the ability of any MOEA to generate Pareto optimal solutions that are non-dominated by the solutions of the other MOEA we are comparing against. For this latter case, statistics are obtained as average values of the comparison of all the independent runs from the first algorithm with respect to all the independent runs from the second algorithm (i.e., statistics are based on $32^2 = 1024$ comparisons).

Tables 8 to 16 show the comparison matrices for the *C-Metric* values obtained with the different algorithms and for all the MOPs used in the experiments. The diagonal values of each matrix correspond to the *C-Metric* for each algorithm, as evaluated with respect to the true Pareto front (i.e., C-Metric($\mathsf{PF}_{\mathsf{true}}$,Algorithm)). The off-diagonal elements correspond to the comparisons between each pair of algorithms, in both ways (i.e., C-Metric(Algorithm_A,Algorithm_B), and C-Metric(Algorithm_B,Algorithm_A)).

From these tables, it can be observed that *MODE-LD+SS* significantly outperforms all other algorithms in terms of convergence. *MODE-LD+SS* is able to converge closer to the true Pareto front in all MOPs, and to generate Pareto front approximations, having fewer solutions (none in many cases) being dominated by those generated by the other algorithms. Additionally, our proposed *MODE-LS+SS* generated more solutions that dominate those generated by the other algorithms. It is also important to note that for ZDT6, our proposed *MODE-LD+SS*, was able to reach the true Pareto front in the 32 independent runs performed.

For the case of DTLZ1 and DTLZ2, and regarding the *C-Metric* values presented in Tables 13 and 14, it can be observed that *MODE-LD+SS* is able to converge very close to the true Pareto front as indicated by the corresponding *C-Metric* measure. These results contrast with the Hv measure obtained by *SPEA2* for these same MOPs. The differences can be explained by the fact that

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	0.968750	0.000771	0.033167	0.000000
SPEA2	0.378115	0.895000	0.106198	0.000000
MOEA/D	0.299833	0.047767	0.883930	0.000000
MODE-LD+SS	0.589893	0.214844	0.274901	**0.374333**

Table 8: C-Metric(A,B) for ZDT1 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	1.000000	0.000303	0.026203	0.000000
SPEA2	0.362813	0.985938	0.041712	0.004331
MOEA/D	0.450922	0.154067	0.393976	0.057031
MODE-LD+SS	0.702266	0.242832	0.110288	**0.381057**

Table 9: C-Metric(A,B) for ZDT2 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	0.656875	0.002246	0.064717	0.000000
SPEA2	0.339297	0.389375	0.142818	0.000067
MOEA/D	0.221500	0.082778	0.389439	0.023824
MODE-LD+SS	0.377051	0.171533	0.299007	**0.199554**

Table 10: C-Metric(A,B) for ZDT3 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	1.000000	0.301200	0.002571	0.000166
SPEA2	0.546084	1.000000	0.003022	0.000566
MOEA/D	0.977144	0.938814	0.952296	0.164757
MODE-LD+SS	0.988408	0.976602	0.689329	**0.220064**

Table 11: C-Metric(A,B) for ZDT4 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	0.986873	0.000000	0.001372	0.000000
SPEA2	1.000000	0.990000	0.040134	0.000000
MOEA/D	0.986999	0.615444	0.990552	0.000000
MODE-LD+SS	0.992119	0.990000	0.976816	**0.000000**

Table 12: C-Metric(A,B) for ZDT6 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	0.655461	0.001915	0.000095	0.000000
SPEA2	0.707633	0.258360	0.012861	0.000000
MOEA/D	0.377986	0.019929	0.163130	0.0005383
MODE-LD+SS	0.611632	0.045080	0.263892	**0.008116**

Table 13: C-Metric(A,B) for DTLZ1 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	0.354375	0.027106	0.000000	0.000000
SPEA2	0.044411	0.806858	0.000000	0.000000
MOEA/D	0.722926	0.071016	0.142447	0.005816
MODE-LD+SS	0.082272	0.078098	0.008309	**0.074566**

Table 14: C-Metric(A,B) for DTLZ2 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	1.000000	0.000221	0.000000	0.000000
SPEA2	0.877437	0.798140	0.010756	0.001108
MOEA/D	0.418284	0.380625	0.639418	0.008286
MODE-LD+SS	0.977820	0.535140	0.370058	**0.339882**

Table 15: C-Metric(A,B) for DTLZ3 using different MOEAs

C-Metric(A,B)	NSGA-II	SPEA2	MOEA/D	MODE-LD+SS
	Mean	Mean	Mean	Mean
NSGA-II	0.361563	0.026370	0.000008	0.000000
SPEA2	0.043145	0.746696	0.000000	0.000000
MOEA/D	0.076018	0.067166	0.124613	0.009343
MODE-LD+SS	0.077891	0.077581	0.001519	**0.107422**

Table 16: C-Metric(A,B) for DTLZ4 using different MOEAs

SPEA2 obtained a better distribution of solutions. Thus, in this case, one algorithm provided better convergence (*MODE-LD+SS*), while the other provided a better spread of solutions (*SPEA2*) (see Figures 52 and 53). Figures 47 to 55 show the comparison of the obtained Pareto fronts by the four MOEAs and for all the MOPs adopted in our study. From these figures we can observe that our *MODE-LD+SS* approach obtains in general, for all numerical test MOPs, Pareto-optimal solutions covering the entire true Pareto front region with a good distribution along it.

7.2 ASSESSMENT OF PMODE-LD+SS WITH NUMERICAL PROBLEMS

In this section, we present the results obtained by the proposed parallel approach *pMODE-LD+SS*. As a first step, the serial version of *MODE-LD+SS* is compared with respect to *NSGA-II*, *MOEA/D*, and *MOEA/D-DE* [2]. In this case, using different parameter settings to those presented in the last section. Then, the *number of islands*, value of an *epoch* and *migration rate* adopted in the parallel approach, are tuned by means of an empirical study, using the *ZDT1* test MOP. Finally, the results obtained with the parallel approach are presented and compared to those of the serial version of *MODE-LD+SS*. These comparisons are based on the average results from 32 independent runs executed by each algorithm and for each MOP.

[2] We have included *MOEA/D-DE* for comparison since this MOEA was published during the research process of this thesis, and additional to the decomposition mechanisms as used in its predecessor *MOEA/D*, it incorporates the Differential Evolution operators, similar to our proposed *MODE-LD+SS* algorithm. Also, *MOEA/D-DE* was successfully tested in solving complex MOPs having difficult Pareto-optimal sets [149].

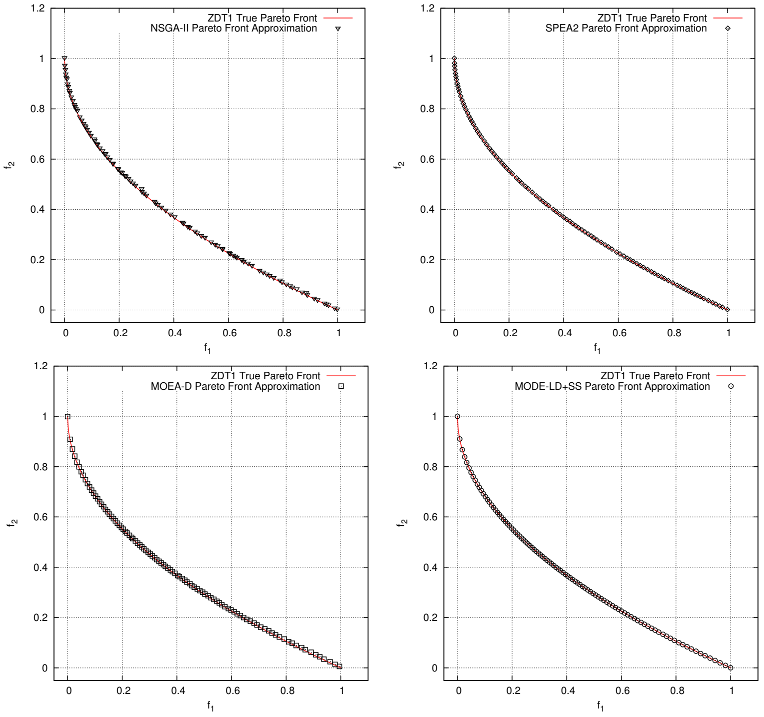


Figure 47: Pareto front approximation for ZDT1 using different MOEAs

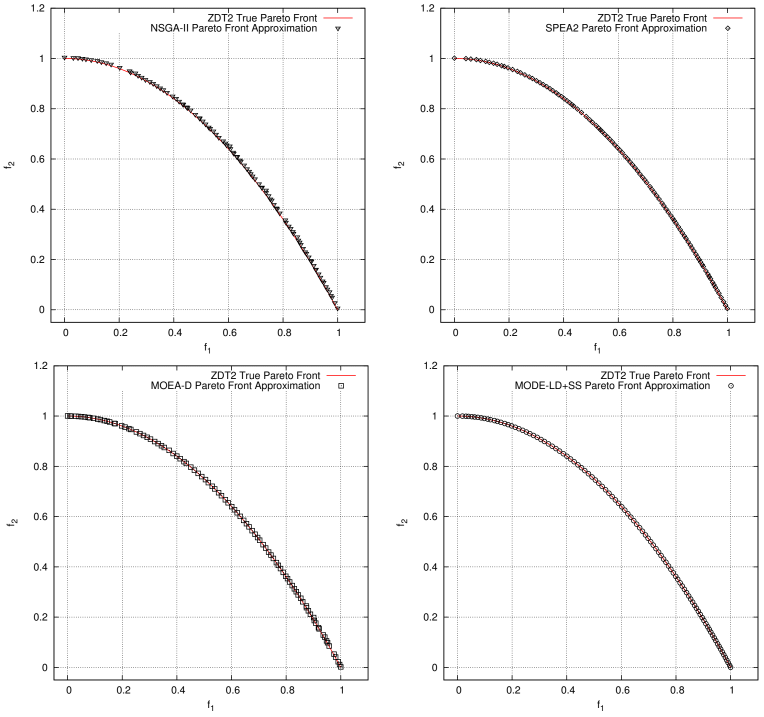


Figure 48: Pareto front approximation for ZDT2 using different MOEAs

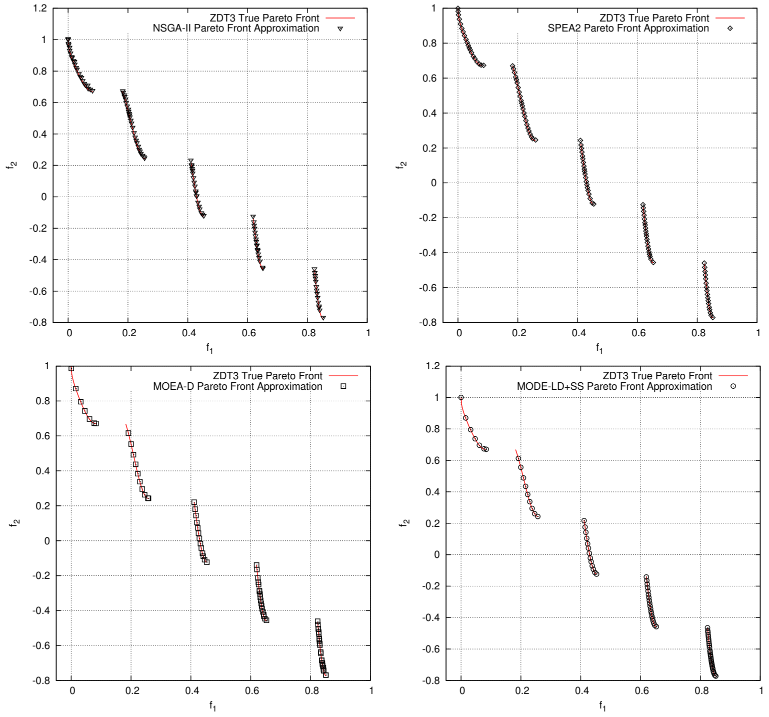


Figure 49: Pareto front approximation for ZDT3 using different MOEAs

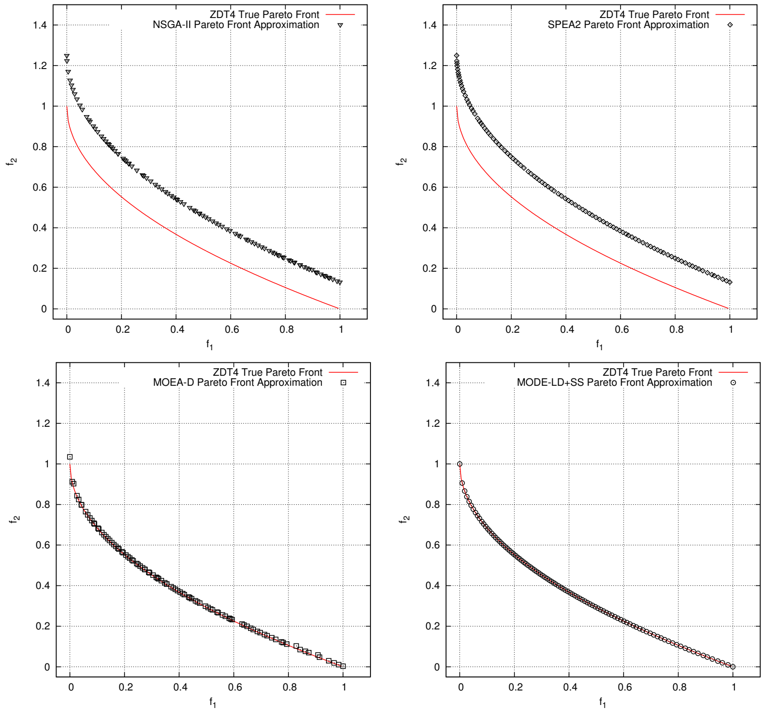


Figure 50: Pareto front approximation for ZDT4 using different MOEAs

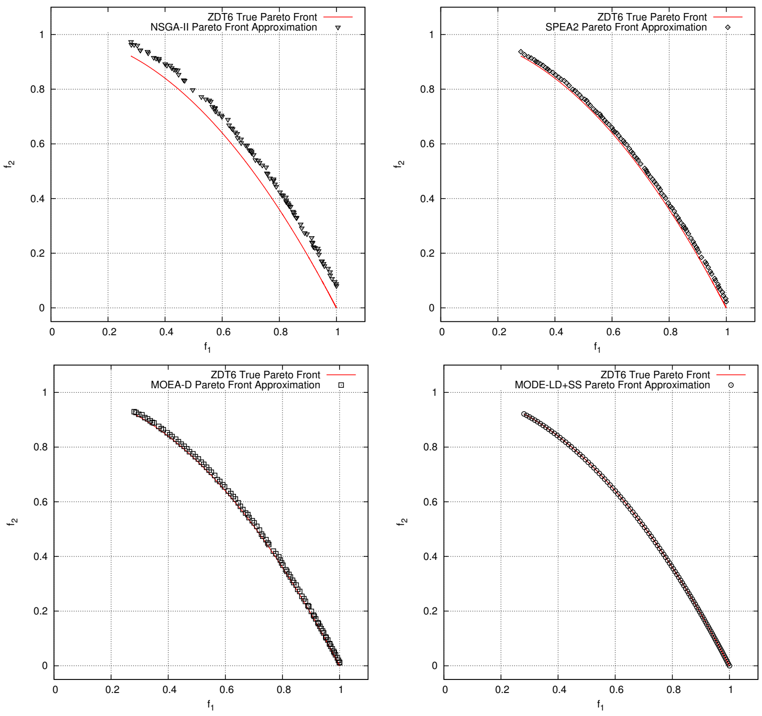


Figure 51: Pareto front approximation for ZDT6 using different MOEAs

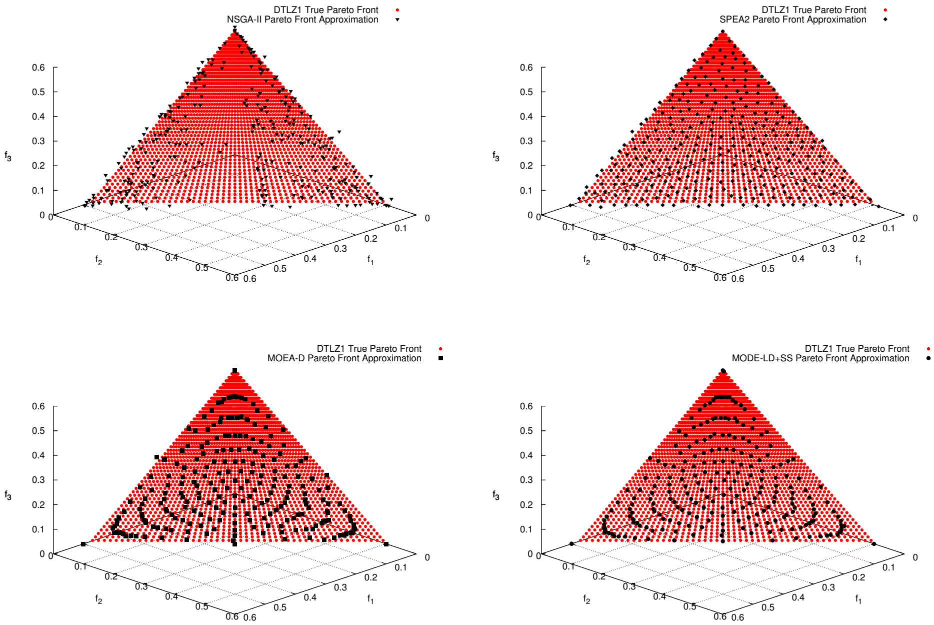


Figure 52: Pareto front approximation for DTLZ1 using different MOEAs

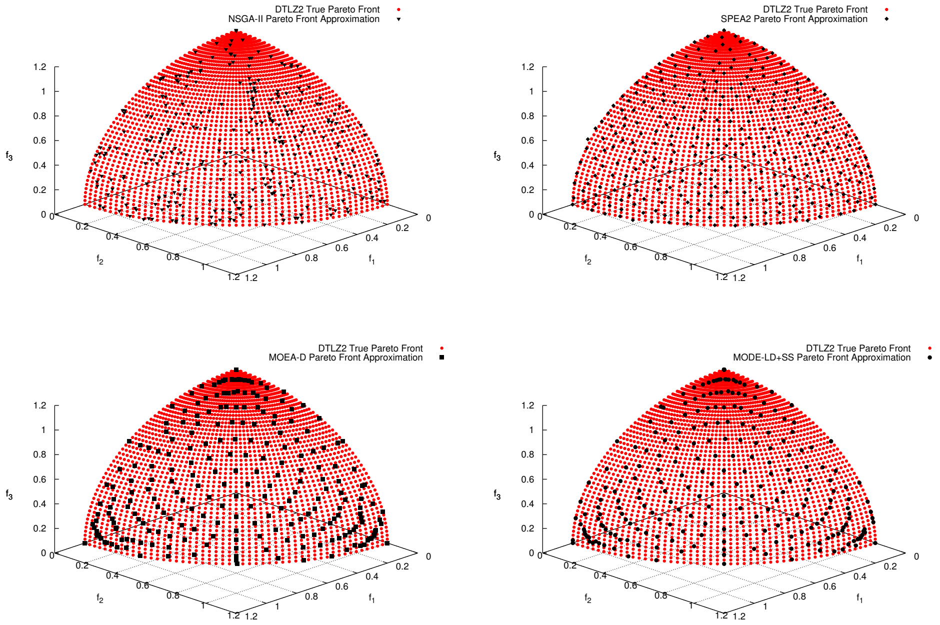


Figure 53: Pareto front approximation for DTLZ2 using different MOEAs

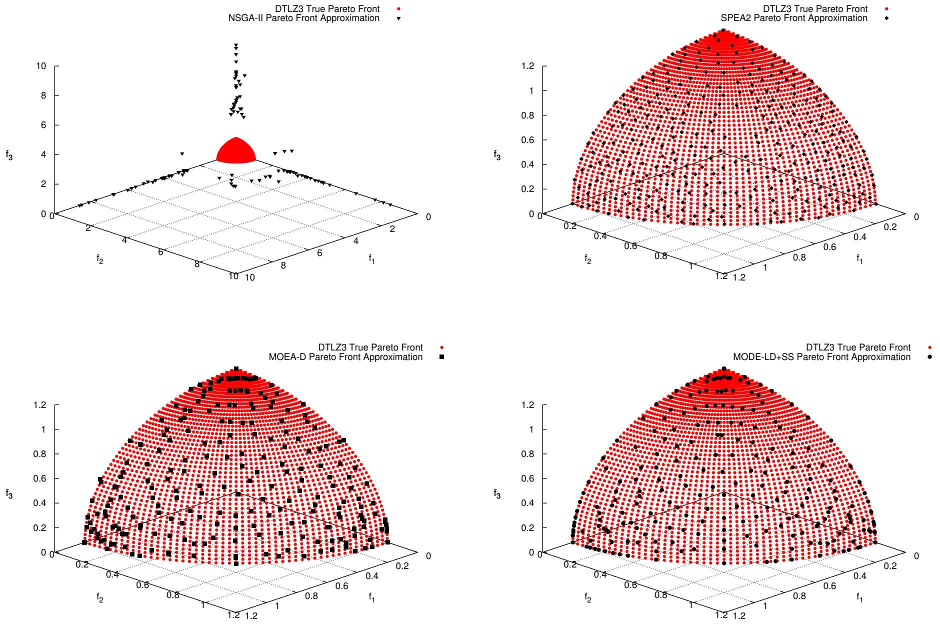


Figure 54: Pareto front approximation for DTLZ3 using different MOEAs

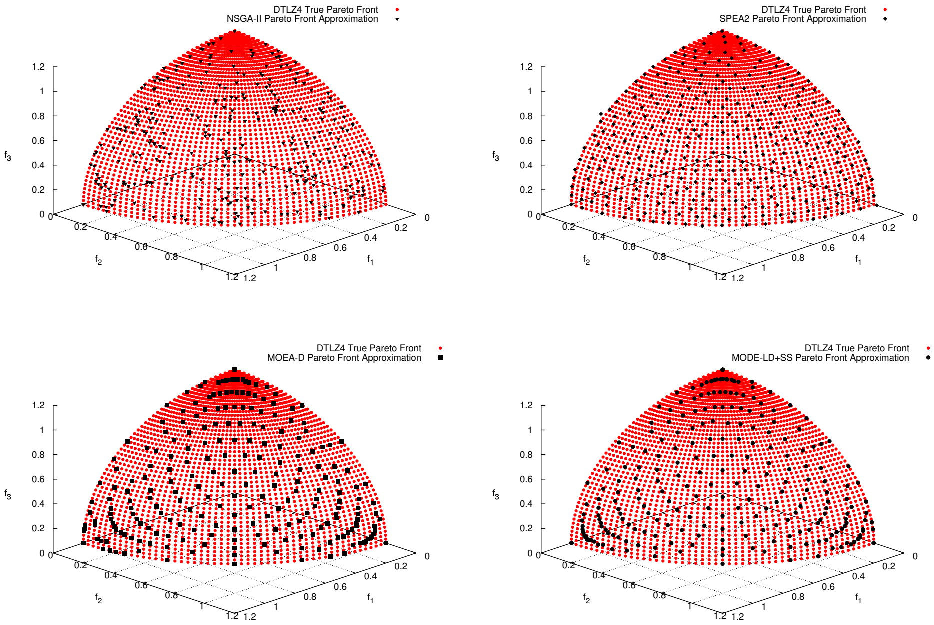


Figure 55: Pareto front approximation for DTLZ4 using different MOEAs

7.2.1 *Parameter settings for the MOEAs compared*

The parameters used in the experiments for the different algorithms adopted in this section are described next. The common parameters for all algorithms comprise the population size N and maximum number of generations $GMAX$. In this case they were set to $N = 50$ for all the bi-objective MOPs and $N = 153$ for all the MOPs having three objectives [3]. The maximum number of generations was set to $GMAX = 150$ for all MOPs, except for *ZDT4* and *DTLZ3*, in which we used $GMAX = 300$. As for specific parameters of each algorithm, the common parameters for *NSGA-II*, *MOEA/D* and *MOEA/D-DE* were set to: *(a).-Crossover probability* $p_c = 1.0$; *(b).- mutation probability* $p_m = 1/NVARS$ (where $NVARS$ corresponds to the number of variables

[3] The reader should note that in this case we are using smaller population sizes as compared to the ones used in the previous section, when we evaluated our proposed *MODE-LD+SS*. Roughly, we are halving the population sizes in this case. The motivation for this reduction in population size is twofold. On the one hand, we are willing to evaluate the performance of our proposed *MODE-LD+SS*, using small population sizes, and on the other hand, our interest is also in solving aerodynamic shape optimization problems having a high computational cost. In this sense, we are interested in testing our approach in a real scenario where a reduced number of objective functions will be allowed, to keep at a low level the computational cost.

in each numerical MOP); *(c).- distribution index for crossover* $\eta_c = 15$; and *(d).- distribution index for mutation* $\eta_m = 20$. For *MOEA/D* and *MOEA/D-DE* the replacing neighborhood size was set as indicated in [267] and [149], respectively. For the *MODE-LD+SS* algorithm, we used: $F = 0.5$ for all MOPs; $CR = 0.5$ for all MOPs except for ZDT4, where $CR = 0.3$ was used; Neighborhood size $NB = 5$ for all MOPs except for ZDT4, where we used $NB = 1$.

With these parameter settings, in Table 17 are presented the Hv performance measures for the four MOEAs compared in this section, namely *NSGA-II*, *MOEA/D*, *MOEA/D-DE*, and *MODE-LD+SS*. The reference point (z_{ref}) in objective space for computing the Hv performance measure was set to *(1.05,1.05)* for all the 2-objective MOPs. In the case of the three-objective MOPs DTLZ1 to DTLZ4, the reference point was set to *(5.0,5.0,5.0)*. This indicated reference point was chosen due to the fact that the NSGA-II presented, for some runs, Pareto front approximations very far from the true Pareto front. As previously indicated, high values in the Hv performance measure indicate that the solutions are closer to the true Pareto front and that they cover a wider extension of it. From the results presented in Table 17, it can be observed that our proposed *MODE-LD+SS* using a reduced population size, obtains the best results in six of nine MOPs for the Hv performance measure. MOEA/D was the algorithm that best performed in DTLZ1 and DTLZ4. However, taking a closer look, the reader can observe that the results for these two MOPs, attained by *MODE-LD+SS*, are very close to those attained by *MOEA/D*. Next, in Table 18 are presented the C-Metric performance measures for all the four MOEAs compared in this section. The C-Metric indicated in Table 18 was computed with respect to the true Pareto front, i.e., C-Metric(PF_{true},Algorithm). From this table we can observe that *MODE-LD+SS* obtains the best results in eight of nine MOPs when we consider the C-Metric performance measure. The results obtained for both performance measures compared, clearly indicates that our proposed *MODE-LD+SS*, in general, is able to converge faster and closer towards the true Pareto front, as compared to the other MOEA algorithms, when a reduced population size is used. Due to space limitations, we do not show figures of the obtained Pareto front approximations by the *MODE-LD+SS* with these parameter settings, but the corresponding analysis also indicates that our *MODE-LD+SS* approach, using a reduced population size, obtains, in general, and for all numerical test MOPS, Pareto optimal solutions covering the entire true Pareto front with a good spread along it.

Test Function	Algorithm			
	NSGA-II	MOEA/D	MOEA/D-DE	MODE-LD+SS
	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)
ZDT1	0.740382 (0.003323)	0.716729 (0.024506)	0.583847 (0.076507)	**0.757395** (0.000397)
ZDT2	0.377348 (0.070194)	0.176615 (0.079320)	0.082341 (0.115367)	**0.424895** (0.000331)
ZDT3	0.604214 (0.003199)	0.585094 (0.023488)	0.277813 (0.111381)	**0.613846** (0.000307)
ZDT4	0.073098 (0.122631)	**0.730980** (0.016966)	0.450990 (0.215977)	0.349325 (0.285549)
ZDT6	0.292164 (0.020894)	0.375312 (0.007755)	0.239793 (0.084688)	**0.407638** (0.000009)
DTLZ1	124.139600 (1.113898)	**124.969600** (0.000768)	119.402900 (7.771898)	124.967700 (0.000383)
DTLZ2	123.972600 (0.124088)	124.397400 (0.001778)	124.353700 (0.027743)	**124.397600** (0.003356)
DTLZ3	80.131930 (39.091680)	124.338100 (0.250190)	85.976200 (54.287600)	**124.396900** (0.003004)
DTLZ4	123.934300 (0.125475)	**124.400100** (0.002818)	124.387900 (0.003452)	124.393900 (0.002684)

Table 17: Comparison of Hv performance measure for different MOEAs using a reduced population size.

7.2.2 *Parameter setting for the pMODE-LD+SS approach*

For any *pMOEA* approach based on the island model, additional to the parameters required by its serial counterpart, in terms of population size, maximum number of generations, and crossover and mutation related parameters, we have to define the *number of islands*, *migration rate*, and *epoch* period. The choice of these parameters has a great influence in the performance of the *pMOEA* and is problem dependent. For selecting a set of parameters to be used in the present section, *ZDT1* was selected to conduct an experimental study for assessing how the parameters affected performance with respect to the serial version. For this study, the following set of parameters was used:

- *Number of Islands (NI)* = 4, 6, and 8.
- *Migration Rate (MR)* = 0.1, 0.2, 0.3 and 0.5.
- *Epoch* = 10, 20, and 50 generations.

Test Function	**Algorithm**			
	NSGA-II	MOEA-D	MOEA-D-DE	MODE-LD+SS
	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)
ZDT1	0.994591 (0.008785)	0.997234 (0.007463)	1.000000 (0.000000)	**0.748125** (0.153569)
ZDT2	1.000000 (0.000000)	**0.208557** (0.140626)	1.000000 (0.000000)	0.586492 (0.100261)
ZDT3	0.931490 (0.047844)	0.813861 (0.121330)	1.000000 (0.000000)	**0.384729** 0.092223
ZDT4	1.000000 (0.000000)	0.975157 (0.088624)	1.000000 (0.000000)	**0.845625** 0.364496
ZDT6	0.975723 (0.008476)	0.978242 (0.001639)	0.989831 (0.016529)	**0.000625** (0.003536)
DTLZ1	0.535550 (0.134412)	0.340389 (0.234715)	0.807088 (0.113710)	**0.021434** (0.014189)
DTLZ2	0.447368 (0.035370)	0.211798 (0.040208)	0.678562 (0.053395)	**0.171215** 0.009018
DTLZ3	1.000000 (0.000000)	0.725727 (0.179974)	0.972366 (0.063160)	**0.160711** (0.007169)
DTLZ4	0.453536 (0.058747)	0.205555 (0.036333)	0.554462 (0.051949)	**0.156578** (0.008628)

Table 18: Comparison of C-Metric performance measure for different MOEAs using a reduced population size.

All the combinations were tested. The parameters for maximum number of generations $GMAX$, F, CR, and NB were set, for all the islands, to the same values selected in the serial version previously described. However, it is important to note that as the number of islands increases, the population size N in each island is reduced accordingly, in order to have a global population similar to that of the serial version of *MODE-LD+SS*.

Table 19 presents the results of the study regarding the C-Metric as evaluated with respect to the true Pareto-front, i.e. C-Metric(PF_{true},Algorithm). Metric improvement in this table is expressed as a percentage variation with respect to the serial version values. Negative values represent a degradation in the performance, which is explained by the algorithm's reduction of its exploration capabilities, due to the use of smaller populations, as the number of participating islands grows. From the results of this study, and regarding the C-Metric, it was observed that high migration rates with shorter epoch periods produce the best improvements with respect to the serial version. However, this can lead to higher communication costs. From the study, the final set of parameters selected corresponds to the following:

- *Number of Islands (NI)* = 6.
- *Migration Rate (MR)* = 0.4.
- *Epoch* = 10 generations.

which will be used in assessing the parallel approach proposed here.

N Islands	4			6			8		
MR/E	10	20	50	10	20	50	10	20	50
0.1	5.01	2.67	-6.18	1.84	6.27	-5.18	2.84	2.84	-5.68
0.2	9.19	6.18	-1.92	11.86	5.60	-2.09	4.76	10.36	-1.84
0.3	10.36	-0.58	-0.75	16.29	2.76	0.17	5.76	3.68	-0.58
0.5	17.71	10.19	3.68	22.47	16.04	1.92	24.64	16.29	2.59

Table 19: C-Metric improvement for ZDT1 study, varying migration rate, epoch and number of islands

7.2.3 *Results and discussion*

Our proposed *pMODE-LD+SS* is based on the island *pMOEA* paradigm. In this sense we can consider *pMODE-LD+SS* as having a given number of islands, whose individuals (a certain proportion) will migrate to solve the MOP in a collaborative manner. For the island paradigm and for our proposed *pMODE-LD+SS*, we can consider two different parallel operating approaches. The first one will be named *pMODE-LD+SS for effectiveness improvement* (*pMODE-LD+SS(A)* for short). In this approach, we consider each island as being a serial version of the *MODE-LD+SS*. Thus, the collaboration and exchange of their individuals will help to solve the MOP more *effectively*, i.e., a better MOP solution is expected due to the synergistic manner this approach behaves. The second operating approach will be named *pMODE-LD+SS for efficiency improvement* (*pMODE-LD+SS(B)* for short). In this second operating approach, the population of each island corresponds to a reduced one. By collaborating and exchanging good individuals from each island, it will help to solve the MOP more *efficiently*, i.e. a MOP solution is expected to be done in less time than compared to the serial version of the MOEA. Next we present the results for these two different operating *pMODE-LD+SS* approaches.

pMODE-LD+SS for effectiveness improvement

Table 20 shows the results of the two proposed parallel operating approaches compared to the results obtained with the serial version of *MODE-LD+SS*. From this table, it can be observed that the approach designed for effectiveness improvement *pMODE-LD+SS(A)*, produced better Hv values in 4 of the 9 MOPs (ZDT1, ZDT3, ZDT4, and ZDT6), and at the same time improves the C-Metric in 7 of the 9 MOPs (ZDT1, ZDT4, ZDT6, DTLZ1, DTLZ2, DTLZ3, DTLZ4). Both metrics were compared with respect to the serial version of *MODE-LD+SS*. One important result to remark from this parallel operating approach, is its ability to reach the true Pareto front of ZDT4 and ZDT6 in the 32 runs performed, as indicated by the mean and standard deviations for the C-Metric in these two MOPs.

pMODE-LD+SS for efficiency improvement

For this second operating approach, each island uses a reduced population size of $N = 10$ for the 2-objective MOPs and of $N = 28$ for the 3-objective MOPs. Since we used 6 islands, the global population consists of 60 individuals for the 2-objective MOPs, and of 168 individuals for the 3-objective MOPs. Considering that the global population size grows, the maximum number of generations used in *pMODE-LD+SS(B)* was reduced accordingly to obtain an equivalent number of objective function evaluations as in the serial version. It is remarked that once the islands' populations are gathered and a global environmental selection is performed, the maximum population size reported for this approach is of 50 solutions for the 2-objective MOPs, and 153 for the 3-objective MOPs.

This latter condition is due to the fact that each island searches for the Pareto extreme solutions (there are redundant solutions which are filtered out). The parameters for F, and CR were set the same as in the serial version for all islands. However, due to the reduction in island population size, the parameter NB was set to 1 in all MOPs. In Table 20, the estimated average parallel speedup measure is reported for all the MOPs used. Also from this table, it can be seen that the approach designed for efficiency improvement produced better Hv values in three (DTLZ1, DTLZ2, and DTLZ4) of the nine MOPs adopted. By taking a closer look to the results for the Hv metric for ZDT1, ZDT2, ZDT3 and ZDT6, it can be seen that *pMODE-LD+SS(B)* obtained values very close to those of the serial version (MODE-LD+SS), even when each island was using a small population size. These conditions confirm the expected behavior for this operating approach, i.e., similar results are obtained with a reduced computational time.

7.3 ASSESSMENT OF MODE-LD+SS IN AERONAUTICAL ENGINEERING PROBLEMS

In this section we present the application of *MODE-LD+SS* for the seven selected aerodynamic shape optimization problems *ASO-MOP1* to *ASO-MOP7*. In this case, we also present the comparisons with respect to the results obtained by the following MOEAs: *MOEA/D*, *MOEA/D-DE*, *SMS-EMOA*, *ARMOGA*, and $\epsilon\mu$*ARMOGA*. All these MOEAs are considered as state-of-the-art. The consideration of the three latter MOEAs, namely *SMS-EMOA*, *ARMOGA*, and $\epsilon\mu$*ARMOGA* is because they have been successfully used to solve aerodynamic shape optimization problems similar to those we have defined as benchmarks in this thesis. For completeness, we provide next a brief description of the main characteristics for these three MOEAs:

- *SMS-EMOA* [17] is a steady-state algorithm based on two basic characteristics: (1) non-dominated sorting is used as its ranking criterion and (2) the hypervolume is applied as its selection criterion to discard that individual, which contributes the least hypervolume to the worst-ranked front. The basic algorithm starts with an initial population of μ individuals, and then a new individual is generated by means of randomized variation operators. Its authors adopted simulated binary crossover (SBX) and polynomial-based mutation. The new individual will become a member of the next population, if replacing another individual leads to a higher quality of the population with respect to the hypervolume. In this way, *SMS-EMOA* keeps those individuals which maximize the population's S-Metric value, which implies that the covered hypervolume of a population cannot decrease as evolution progresses.

- *ARMOGA* [208] uses real-numbers encoding, Fonseca's Pareto ranking [68], fitness sharing, stochastic universal selection, Simulated Binary Crossover (SBX), and polynomial-based mutation. ARMOGA incorporates two archiving techniques: a global archive, which stores all the best solutions obtained so far, and a recent archive, which stores the best solutions of the past previous P generations. Solutions from the second archive participate in the parent selection process. Also, a distinguishing characteristic of ARMOGA, is that it adapts the decision variables ranges, based on the statistics of the current nondominated solutions found so far. The range adaptation in ARMOGA takes place at every M generations. When performing this operation, a new range for decision variables is defined, based on the statistics of a set of design solutions. This set is formed with solutions taken from the archives, and the current generation.

- $\epsilon\mu$*ARMOGA* [229] called also *multi-objective micro-genetic algorithm with range adaptation, based on* ϵ*-dominance*. As implied by its name, is inspired on ARMOGA [208]. Additional to the characteristics inherited from it, $\epsilon\mu$ARMOGA introduces two additional mechanisms. The first corresponds to the use of a small population size (i.e. the use of a micro-genetic algorithm as in [131, 40]), coupled with the use of an external file or archive for storing the nondominated solutions obtained so far. The second mechanism corresponds to the use of the concept of ϵ-dominance [145], which is a relaxed form of Pareto dominance that has been used as an archiving strategy that allows to regulate convergence. The main population is reinitialized at every certain number of generations, based on the average and standard deviation of the decision variables. No mutation operator is used (the reinitialization procedure is considered to be the source of diversity in this case).

7.3.1 *Parameter Settings*

The parameters used in the experiments for the different algorithms adopted, were set as follows. The common parameters for most of the algorithms comprise the population size N and maximum number of generations $GMAX$. These were set to $N = 50$ for all the bi-objective MOPs and $N = 105$ for all the MOPs having three objectives. We adopted $GMAX = 100$ for all MOPs[4]. For the case of $\epsilon\mu$*ARMOGA* the population size is set to 4, and the archive size is set to 50 for all the bi-objective MOPs and 105 for all the MOPs having three objectives. For $\epsilon\mu$*ARMOGA*, the number of generations is adjusted to have an equivalent number of objective functions evaluations for each MOP, as compared to the other MOEAs. Most of the algorithms compared use SBX and Polynomial-based mutation; their specific parameter settings are summarized in Table 21.

For *MOEA/D* and *MOEA/D-DE*, a common parameter is the replacing niche used in the decomposition approach adopted in both MOEAs. The size of this niche was set, for both MOEAs, to 10 for the ASO-MOPs with two objectives

[4] With these population sizes, the number of objective function evaluations will be 5,000 for the ASO-MOPs with two objectives, and 10,500 for the ASO-MOPs with three objectives. This number of objective function evaluations can be considered low for a common MOEA practice. However, for real-world applications, this number can be considered high, due to the computational cost associated to the calls of the CFD simulation code.

and to 20 for the ASO-MOPs with three objectives. *MOEA/D-DE* uses two additional parameters, one is the probability for selecting the mating parents from the defined niche or from the whole population, and the second is the maximum number of solutions to be replaced in the niche once a better solution is found in the decomposition approach. Both parameters were used as the values indicated in the source code, and are 0.9 and 2 respectively.

For *ARMOGA*, range adaptation is executed every 5 generations, while for $\epsilon\mu$*ARMOGA*, reinitialization occurs at every generation. In both algorithms, the associated parameters for range adaptation, were used as those indicated in the source code and in the configuration files provided by the authors. Finally for the $\epsilon\mu$*ARMOGA*, the ϵ-dominance factor used, was that given in the source code. Finally, for our proposed *MODE-LD+SS*, the associated parameters were the following: Scaling factor $F = 0.5$ for all MOPs; crossover rate $CR = 0.5$ for all MOPs; Neighborhood size $NB = 2$ for all MOPs with two objectives, while $NB = 5$ for all MOPs having three objectives.

7.3.2 *Results and discussion*

Table 23 summarizes the results obtained for the Hv performance measure and for all the algorithms used in this section. In these results, the reference point (z_{ref}) in objective space for computing the Hv performance measure in each ASO-MOP, was obtained by considering the attained region in objective space for all the 32 independent runs of each MOEA compared. The specific values are summarized in Table 22. From this table we can observe that our proposed *MODE-LD+SS* approach attains the best performance in five of seven ASO-MOPs (ASO-MOP1, ASO-MOP3, ASO-MOP5, ASO-MOP6, and ASO-MOP7). *MOEA/D-DE* attains the best performance in only one case (ASO-MOP2), while *SMS-EMOA* attains the best performance in only one problem (ASO-MOP4). From the above results presented, it is important to remark that *SMS-EMOA* uses the hypervolume contribution in its selection process; in consequence we would expect *SMS-EMOA* to be the best for this performance measure. Nonetheless, our proposed *MODE-LD+SS* is able, in most ASO-MOPs, to perform better with respect to this performance measure. For the two ASO-MOPs in which our approach was not the best performer, we can observe that its Hv value is very close to that of the winner in these ASO-MOPs, and the difference can be considered very small, on the order of 0.7% for ASO-MOP2, and 0.12% percent for ASO-MOP4. In summary, we claim that our proposed

MODE-LD+SS has very good convergence properties for solving this type of aerodynamic shape MOPs.

Figures 56 to 62 show the approximated Pareto fronts obtained by all algorithms used in this section, and for all the ASO-MOPs solved. Since these ASO-MOPs correspond to real-world MOPs for which we do not know in advance the shape and position of the true Pareto front, neither we have any knowledge regarding their fitness landscapes, we have decided to plot for each ASO-MOP and for each MOEA, the whole set of approximated Pareto fronts, i.e., the 32 obtained approximations. From these figures, we can observe that for most of the ASO-MOPs, their fitness landscapes present some degree of roughness, as indicated by the dispersion of solutions obtained by the different MOEAs. Also, we can observe that our proposed *MODE-LD+SS* is the algorithm having the lowest dispersion of solutions in almost all ASO-MOPs, confirming the good convergence properties of our approach.

As it was previously indicated, from the parameter settings section, we are using a reduced number of objective function evaluations, compared to the standard MOEA practice, but we are using a high number of objective function evaluations, as compared to a real-world MOP solution scenario. In order to better assess the convergence properties of our proposed *MODE-LD+SS*, in Figures 63 and 64 we present the convergence history for the Hv performance measure, normalized with respect to the highest value attained in each ASO-MOP, for the different MOEAs. From these figures we can observe that our proposed approach shows in general, very good convergence properties, outperforming the corresponding convergence rate of the other MOEAs compared. If a low number of objective function evaluations is allowed, it still obtains good Hv values. For example, if we consider a budget of $1,000$ objective function evaluations, we can expect to attain in four ASO-MOPs (ASO-MOP1 to ASO-MOP4) more than 90% of the Hv performance measure, as compared to the results using 5000 objective function evaluations in the ASO-MOPs with two-objectives, and those using $10,500$ objective function evaluations for the three-objective ASO-MOPs.

One important aspect to highlight in ASO-MOP2 is the condition attained by $\epsilon\mu$*ARMOGA*. In this case, we can observe that on the order of 300 objective function evaluations were required for $\epsilon\mu$*ARMOGA* to attain 95% of the Hv performance measure, as compared to performing $5,000$ objective function evaluations. We can conclude that for this particular ASO-MOP, the mechanisms implemented in $\epsilon\mu$*ARMOGA* are very useful.

Finally, in Tables 24 to 30 we present the C-Metric comparisons for the Pareto front approximations obtained by *MODE-LD+SS*, *SMS-EMOA*, and

MOEA/D-DE. Only these three algorithms were selected to be compared from the whole set used in this section, since they are ranked from first to third (in the order mentioned) with respect to the Hv performance measure presented in Table 23. From these tables and regarding the C-Metric between approximations obtained by the three selected algorithms, we can observe that for all ASO-MOPs, our proposed *MODE-LD+SS* approach outperforms by an ample margin the corresponding results attained by *SMS-EMOA* and *MOEA/D-DE*. *MODE-LD+SS* is able to converge closer to the true Pareto front in all MOPs, and to generate Pareto front approximations, having very few solutions being dominated by those generated by the other algorithms. Additionally, our proposed *MODE-LS+SS* generated more solutions that dominate those generated by the other algorithms.

7.4 ASSESSMENT OF PMODE-LD+SS IN AERONAUTICAL ENGINEERING PROBLEMS

In this section, we present the results obtained by our proposed parallel approach *pMODE-LD+SS* when applied to solve the benchmark proposed for aerodynamic shape optimization problems. The results obtained with it are presented and compared to those of the serial version of *MODE-LD+SS*. These comparisons are based on the average results from 32 independent runs executed by each algorithm and for each MOP.

7.4.1 *Parameter settings for the pMODE-LD+SS approach*

In Section 7.2, we have presented the tuning for the parameters used in our island-based parallel approach, when it was used to solve the ZDTs and DTLZs test problems. In this case, we will use the same parameters previously adopted for the *number of islands*, *migration rate*, and *epoch*. These parameter values are summarized next:

- *Number of Islands (NI)* = 6.
- *Migration Rate (MR)* = 0.4.
- *Epoch* = 10 generations.

As for specific parameters used by the *pMODE-LD+SS* approach, namely population size per island N, maximum number of generations $GMAX$, scale

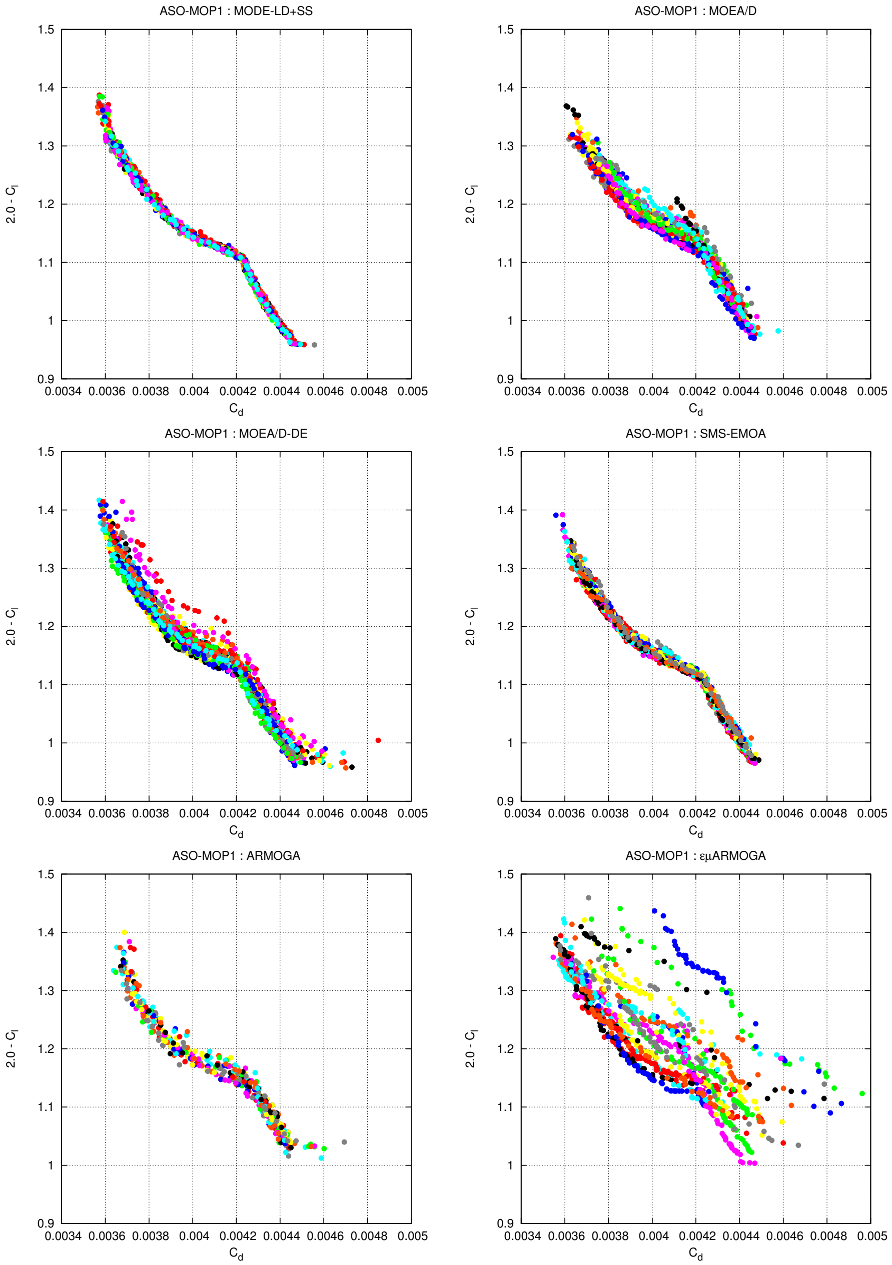


Figure 56: Pareto front approximation for ASO-MOP1 by different MOEAs

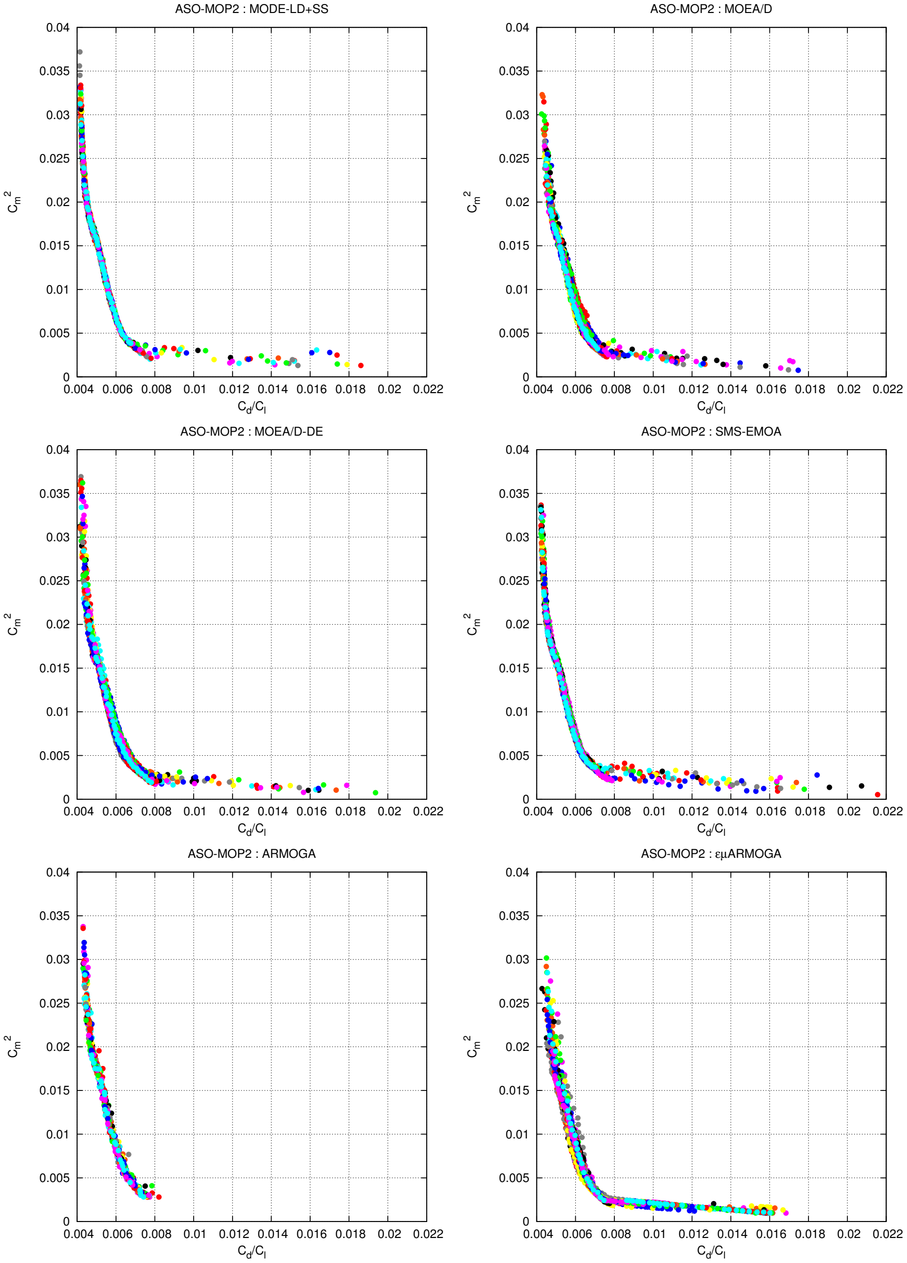


Figure 57: Pareto front approximation for ASO-MOP2 by different MOEAs

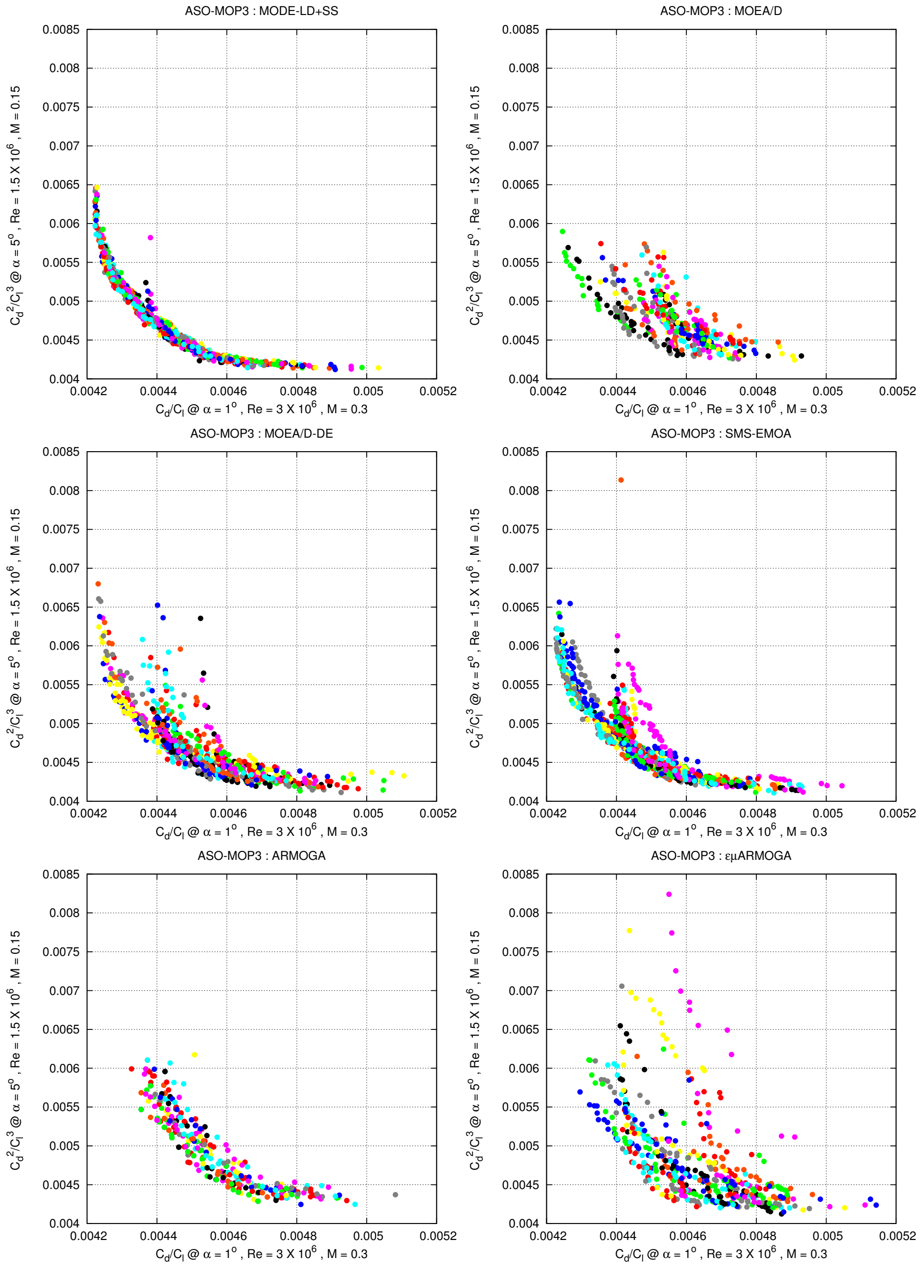
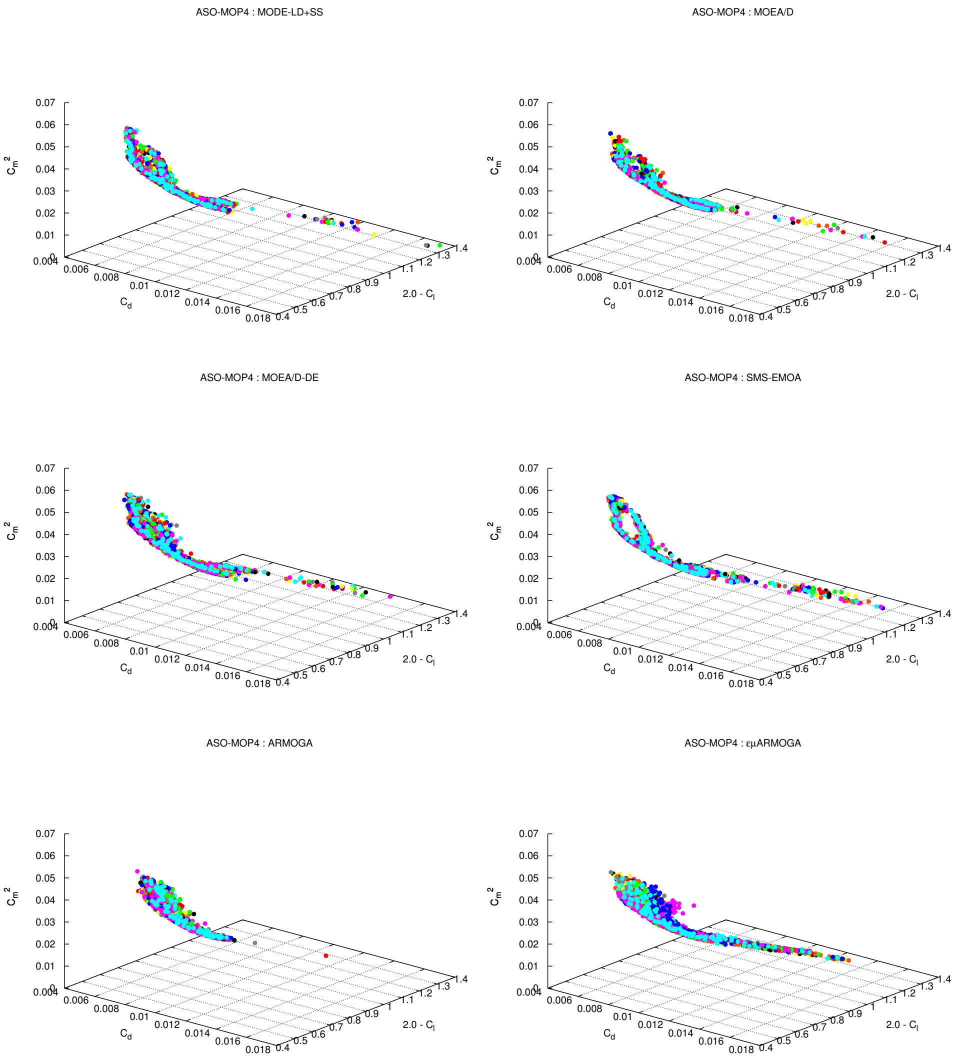


Figure 58: Pareto front approximation for ASO-MOP3 by different MOEAs



ASO-MOP4 : MODE-LD+SS

ASO-MOP4 : MOEA/D

ASO-MOP4 : MOEA/D-DE

ASO-MOP4 : SMS-EMOA

ASO-MOP4 : ARMOGA

ASO-MOP4 : εμARMOGA

Figure 59: Pareto front approximation for ASO-MOP4 by different MOEAs

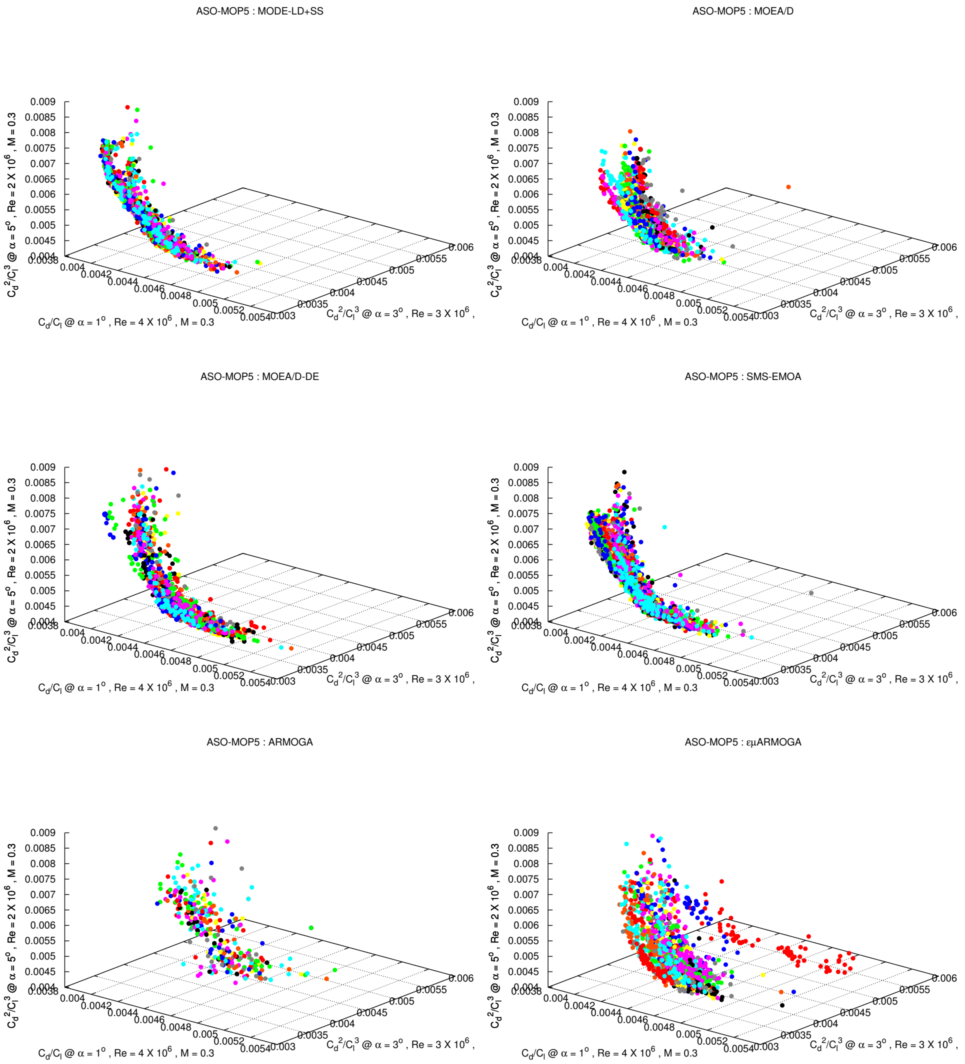


Figure 60: Pareto front approximation for ASO-MOP5 by different MOEAs

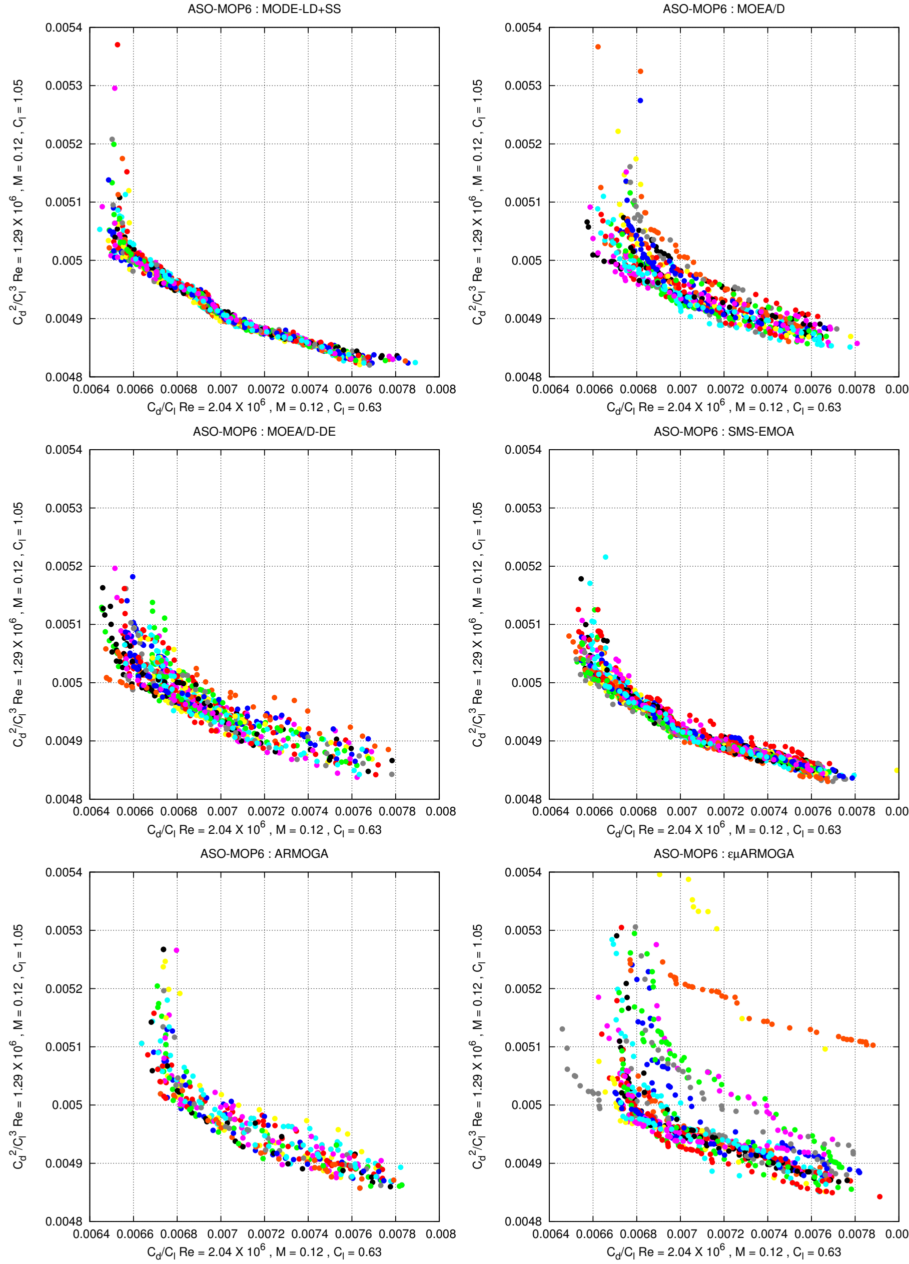


Figure 61: Pareto front approximation for ASO-MOP6 by different MOEAs

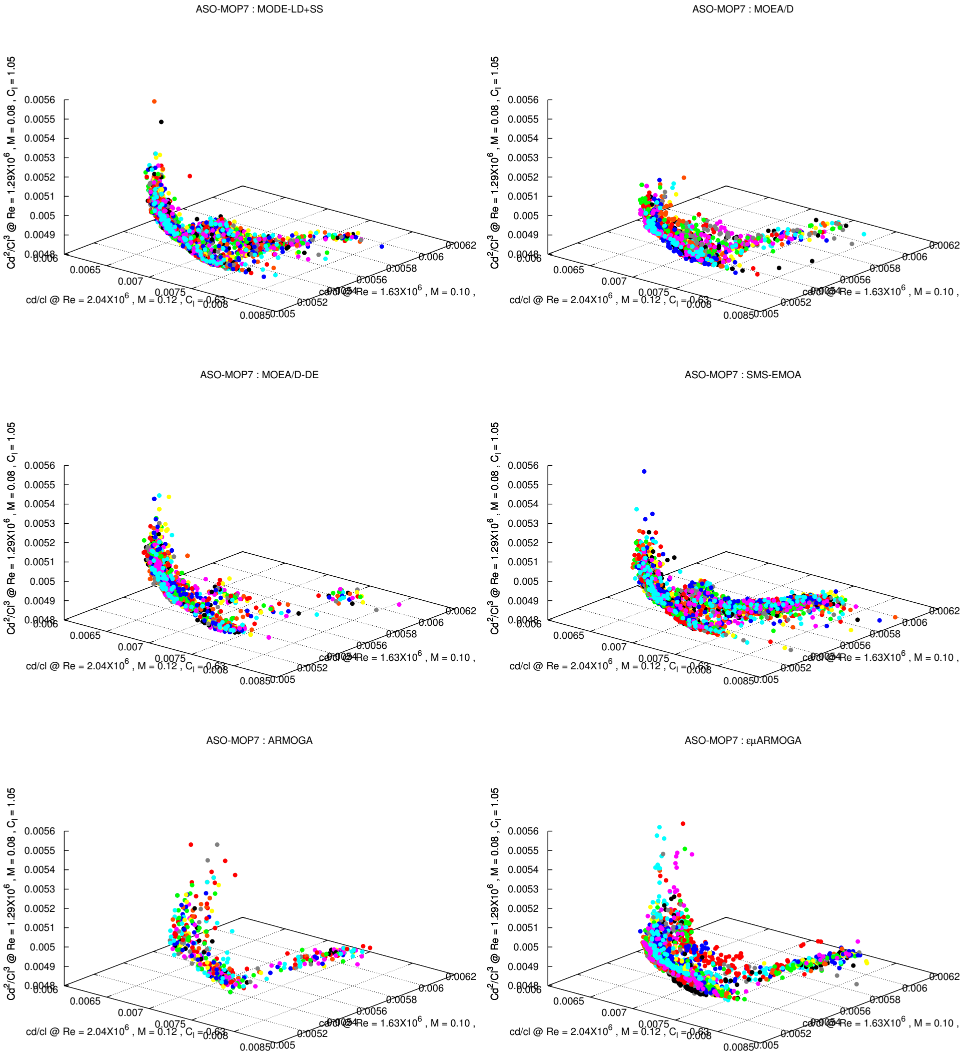


Figure 62: Pareto front approximation for ASO-MOP7 by different MOEAs

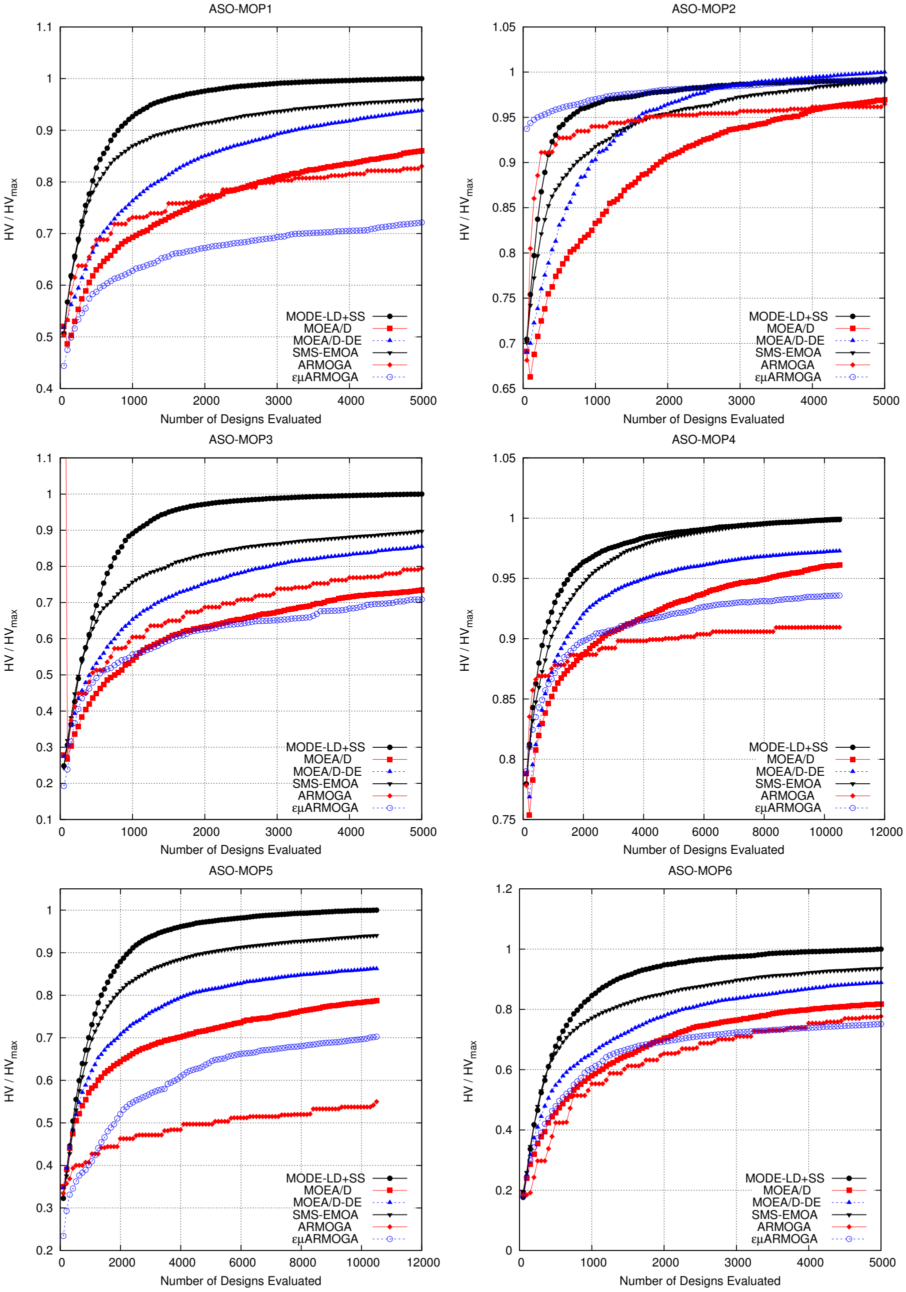


Figure 63: HV convergence for different MOEAs solving ASO-MOPs

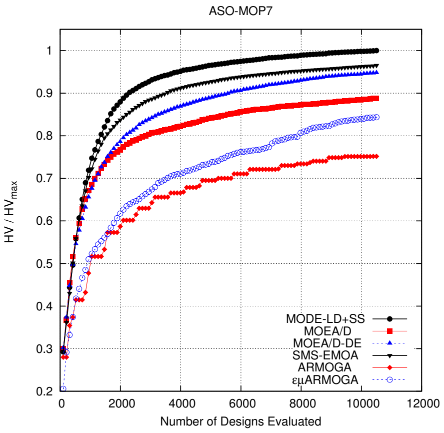


Figure 64: HV convergence for different MOEAs solving ASO-MOPs (continued)

factor F, crossover rate CR, and neighborhood size NB, they were set as follows:

- $N = 10$ for each island when solving two-objective ASO-MOPs, and $N = 20$ for each island when solving three objective ASO-MOPs.
- $GMAX = 83$ when solving two-objective ASO-MOPs, and $GMAX = 87$ when solving three objective ASO-MOPs.
- $F = 0.5$ for all ASO-MOPs.
- $CR = 0.5$ for all ASO-MOPs.
- $NB = 2$ for all ASO-MOPs.

It should be noted that with the combination of *number of islands*, population size per island N, and the number of generations executed $GMAX$, we are doing approximately the same number of objective function evaluations as in the case of the serial version *MODE-LD+SS*. Also, it is remarked that once the islands' populations are gathered and a global environmental selection is performed, the maximum population size reported for this approach is of 50 solutions for the two-objective ASO-MOPs, and 105 for the three-objective ASO-MOPs. This latter condition is due to the fact that each island searches for the Pareto extreme solutions (there are redundant solutions which are filtered out). The above described conditions allows us to do a fair comparison of results.

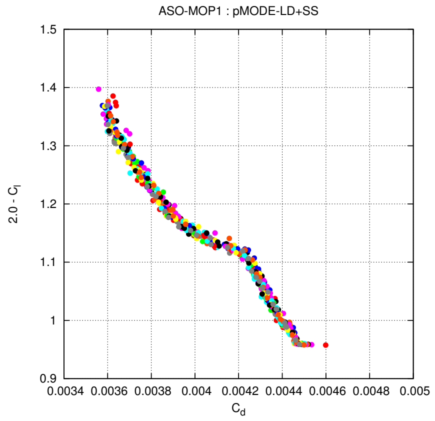


Figure 65: Pareto front approximation for ASO-MOP1 by pMODE-LD+SS

7.4.2 *Results and discussion*

Different to the results in Section 7.2.3, where two operating approaches were presented, in this case only the approach for *efficiency improvement* is reported. Table 31 shows the results obtained in solving the ASO-MOPs and using our proposed *MODE-LD+SS* and *pMODE-LD+SS* approaches. Also in this table are reported the average execution times per run for each ASO-MOP and for the serial version *MODE-LD+SS*. From this table we can observe that *pMODE-LD+SS* is able to attain very close values for the Hv performance measure in 6 of 7 ASO-MOPs, i.e., in all but ASO-MOP2. Surprisingly, for this latter ASO-MOP, *pMODE-LD+SS* was able to attain a higher Hv value. These conditions confirm the expected behavior for this operating approach, i.e., similar results are obtained with a reduced computing time. In the cases where *pMODE-LD+SS* attains close values to its serial counterpart, these differences are at best less than 1%, but not worse than 3%. With this condition and the speed-up attained by using our proposed parallel approach, we can claim that *pMODE-LD+SS* is efficient in solving the benchmark for the proposed aerodynamic shape optimization MOPs.

Finally, Figures 65 to 71 show the approximation of the Pareto front obtained by the proposed *pMODE-LD+SS*. Comparing these figures with the corresponding ones obtained by *MODE-LD+SS*, we can observe that they are very similar and also, they show low variation among different runs, confirming the good convergence properties of *pMODE-LD+SS*, even when smaller populations are used.

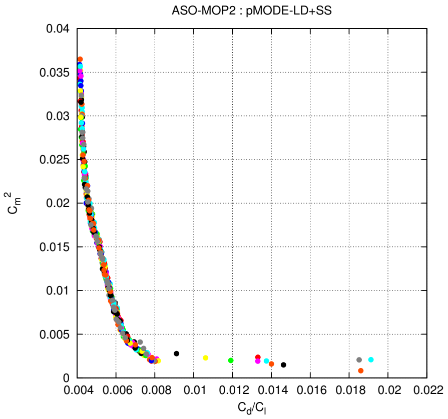


Figure 66: Pareto front approximation for ASO-MOP2 by pMODE-LD+SS

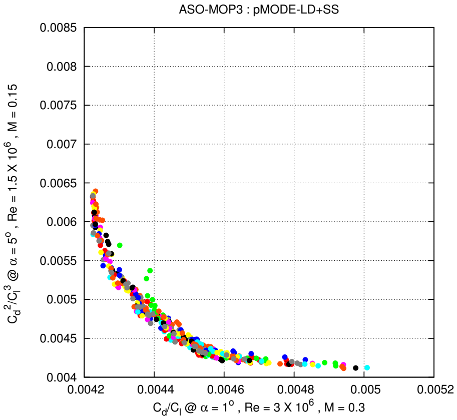


Figure 67: Pareto front approximation for ASO-MOP3 by pMODE-LD+SS

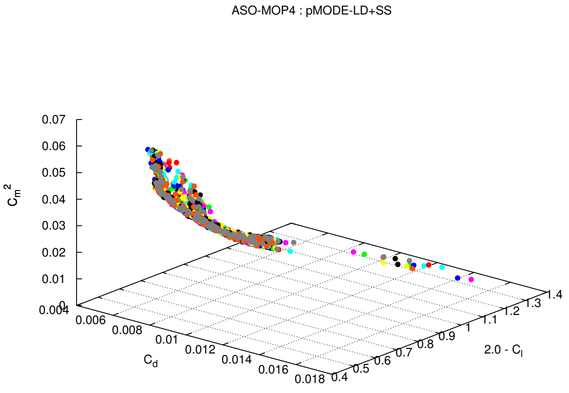


Figure 68: Pareto front approximation for ASO-MOP4 by pMODE-LD+SS

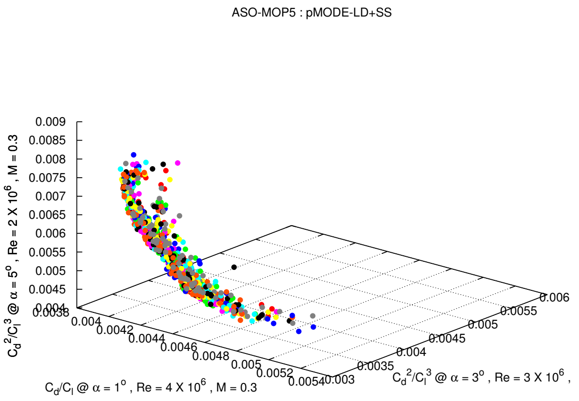


Figure 69: Pareto front approximation for ASO-MOP5 by pMODE-LD+SS

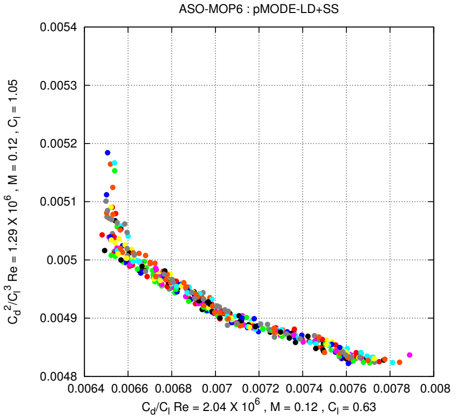


Figure 70: Pareto front approximation for ASO-MOP6 by pMODE-LD+SS

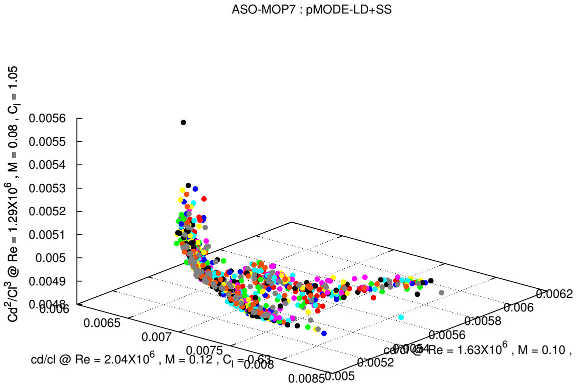


Figure 71: Pareto front approximation for ASO-MOP7 by pMODE-LD+SS

Test Function	MODE-LD+SS		pMODE-LD+SS(A)		pMODE-LD+SS(B)		Speed-up
	Hv	C-Metric	Hv	C-Metric	Hv	C-Metric	
	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)	
ZDT1	0.757395 (0.000397)	0.748125 (0.153569)	**0.757675** (0.000115)	**0.623750** (0.105517)	0.750276 (0.002323)	0.983219 (0.043928)	2.9977
ZDT2	**0.424895** (0.000331)	**0.586492** (0.100261)	0.424363 (0.000310)	0.766844 (0.105424)	0.421071 (0.001954)	0.815243 (0.099875)	2.6918
ZDT3	0.613846 (0.000307)	0.384729 (0.092223)	**0.614156** (0.000201)	0.385278 (0.078578)	0.608588 (0.001898)	0.741921 (0.122588)	2.5434
ZDT4	0.349325 (0.285549)	0.845625 (0.364496)	**0.758770** (0.000006)	**0.000000** (0.000000)	0.034668 (0.119046)	1.000000 (0.000000)	2.3341
ZDT6	0.407638 (0.000009)	0.000625 (0.003536)	**0.407650** (0.000001)	**0.000000** (0.000000)	0.406966 (0.000241)	0.002552 (0.006860)	2.5110
DTLZ1	124.967700 (0.000383)	0.021434 (0.014189)	124.967400 (0.000238)	**0.011863** (0.004840)	**124.970200** (0.000681)	0.022575 (0.009888)	4.9934
DTLZ2	124.397600 (0.003356)	0.171215 (0.009018)	124.391900 (0.002976)	**0.162893** (0.008310)	**124.404200** (0.001968)	0.175189 (0.008008)	4.7269
DTLZ3	**124.396900** (0.003004)	0.160711 (0.007169)	124.389000 (0.003113)	**0.155273** (0.005601)	124.015800 (1.228651)	0.242665 (0.247538)	4.8357
DTLZ4	124.393900 0.002684	0.156578 (0.008628)	124.389500 (0.002848)	**0.154082** (0.008518)	**124.402700** (0.002529)	0.149008 (0.008031)	4.8949

Table 20: Results for MODE-LD+SS(A) and pMODE-LD+SS(B)

Algorithm	η_c	P_c	η_m	Mut. Rate
MOEA/D	20	1.0	20	$1/NVARS$
MOEA/D-DE	20	1.0	20	$1/NVARS$
SMS-EMOA	15	1.0	20	$1/NVARS$
ARMOGA	See note	1.0	See note	0.1
$\epsilon\mu$*ARMOGA*	15	1.0	No mutation is used	

Note: The source code of *ARMOGA* algorithm uses SBX and polynomial-based mutation, but it was indicated to use a revised version. The respective coded function was used without changing anything.

Table 21: Common parameters for MOEAs used in solving ASO-MOPs

	Z_{ref}
ASO-MOP1	(0.0050 , 1.5000)
ASO-MOP2	(0.0220 , 0.0400)
ASO-MOP3	(0.0052 , 0.0085)
ASO-MOP4	(0.0180 , 1.4000 , 0.0700)
ASO-MOP5	(0.0054 , 0.0060 , 0.0090)
ASO-MOP6	(0.0080 , 0.0054)
ASO-MOP7	(0.0085 , 0.0062 , 0.0056)

Table 22: z_{ref} for different algorithms in solving aerodynamic shape optimization problems.

Test Problem	ALGORITHM					
	MODE-LD+SS	MOEA/D	MOEA/D-DE	SMS-EMOA	ARMOGA	$\epsilon\,\mu$ARMOGA
	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)	Mean (σ)
ASO-MOP1	**5.9283e-04** **(2.8857e-06)**	5.10010e-04 (3.6127e-05)	5.5641e-04 (1.6917e-05)	5.6889e-04 (1.2855e-05)	4.9225e-04 (7.4317e-06)	4.2776e-04 (6.3947e-05)
ASO-MOP2	6.4060e-04 (7.1550e-06)	6.2562e-04 (1.8536e-05)	**6.4533e-04** **(5.0172e-06)**	6.3852e-04 (7.9755e-06)	6.2251e-04 (6.3115e-06)	6.4007e-04 (5.6704e-06)
ASO-MOP3	**3.9190e-06** **(1.4313e-07)**	2.8807e-06 (3.4974e-07)	3.3515e-06 (3.2284e-07)	3.5148e-06 (2.8502e-07)	3.1135e-06 (1.3299e-07)	2.7779e-06 (4.9433e-07)
ASO-MOP4	5.9361e-04 (3.1540e-06)	5.7129e-04 (7.7717e-06)	5.7812e-04 (5.2332e-06)	**5.9433e-04** **(2.2422e-06)**	5.4000e-04 (4.2952e-06)	5.5623e-04 (2.0840e-05)
ASO-MOP5	**1.6917e-08** **(2.9378e-10)**	1.3314e-08 (9.6277e-10)	1.4589e-08 (5.1502e-10)	1.5912e-08 (6.7767e-10)	9.3017e-09 (6.2834e-10)	1.1884e-08 (1.6171e-09)
ASO-MOP6	**7.5111e-07** **(1.1750e-08)**	6.1422e-07 (3.5569e-08)	6.6777e-07 (3.2192e-08)	7.0319e-07 (2.2968e-08)	5.8377e-07 (1.9503e-08)	5.6470e-07 (1.0280e-08)
ASO-MOP7	**1.4400e-09** **(1.2175e-11)**	1.2791e-09 (4.5157e-11)	1.3653e-09 (2.7705e-11)	1.3902e-09 (1.9829e-11)	1.0792e-09 (3.9243e-11)	1.2146e-09 (8.5249e-10)

Table 23: Comparison of the Hypervolume (Hv) for different algorithms used for solving aerodynamic shape optimization problems.

C-M(A,B)	*MODE-LD+SS*	*MOEA/D-DE*	*SMS-EMOA*
	Mean	Mean	Mean
MODE-LD+SS		0.9015	0.7443
MOEA/D-DE	0.0139		0.1363
SMS-EMOA	0.0481	0.6682	

Table 24: Comparison of the C-Metric(A,B) in ASO-MOP1 for different algorithms.

C-M(A,B)	*MODE-LD+SS*	*MOEA/D-DE*	*SMS-EMOA*
	Mean	Mean	Mean
MODE-LD+SS		0.7042	0.6034
MOEA/D-DE	0.0415		0.1966
SMS-EMOA	0.0542	0.4904	

Table 25: Comparison of the C-Metric(A,B) in ASO-MOP2 for different algorithms.

C-M(A,B)	*MODE-LD+SS*	*MOEA/D-DE*	*SMS-EMOA*
	Mean	Mean	Mean
MODE-LD+SS		0.9003	0.8223
MOEA/D-DE	0.0231		0.2157
SMS-EMOA	0.0604	0.6033	

Table 26: Comparison of the C-Metric(A,B) in ASO-MOP3 for different algorithms.

C-M(A,B)	*MODE-LD+SS*	*MOEA/D-DE*	*SMS-EMOA*
	Mean	Mean	Mean
MODE-LD+SS		0.2838	0.0261
MOEA/D-DE	0.0077		0.0056
SMS-EMOA	0.0499	0.2936	

Table 27: Comparison of the C-Metric(A,B) in ASO-MOP4 for different algorithms.

C-M(A,B)	*MODE-LD+SS*	*MOEA/D-DE*	*SMS-EMOA*
	Mean	Mean	Mean
MODE-LD+SS		0.6318	0.6138
MOEA/D-DE	0.0136		0.0667
SMS-EMOA	0.0570	0.5004	

Table 28: Comparison of the C-Metric(A,B) in ASO-MOP5 for different algorithms.

C-M(A,B)	*MODE-LD+SS*	*MOEA/D-DE*	*SMS-EMOA*
	Mean	Mean	Mean
MODE-LD+SS		0.8994	0.8608
MOEA/D-DE	0.0278		0.1303
SMS-EMOA	0.0727	0.6989	

Table 29: Comparison of the C-Metric(A,B) in ASO-MOP6 for different algorithms.

C-M(A,B)	*MODE-LD+SS*	*MOEA/D-DE*	*SMS-EMOA*
	Mean	Mean	Mean
MODE-LD+SS		0.6104	0.4430
MOEA/D-DE	0.0492		0.0943
SMS-EMOA	0.0710	0.3681	

Table 30: Comparison of the C-Metric(A,B) in ASO-MOP6 for different algorithms.

Test Problem	ALGORITHM		Average execution time (hh:mm:ss)	Speed-up
	MODE-LD+SS	pMODE-LD+SS		
	Mean (σ)	Mean (σ)		
ASO-MOP1	**5.9283e-04** **(2.8857e-06)**	5.8303e-04 (2.7830e-06)	00:32:05	4.0020
ASO-MOP2	6.4060e-04 (7.1550e-06)	**6.4286e-04** **(5.6052e-06)**	00:41:47	3.9542
ASO-MOP3	**3.9190e-06** **(1.4313e-07)**	3.8219e-06 (2.0055e-07)	1:47:53	4.5360
ASO-MOP4	**5.9361e-04** **(3.1540e-06)**	5.8820e-04 (1.5115e-06)	2:20:28	4.9431
ASO-MOP5	**1.6917e-08** **(2.9378e-10)**	1.6857e-08 (1.1815e-10)	6:46:38	5.2604
ASO-MOP6	**7.5111e-07** **(1.1750e-08)**	7.4219e-07 (9.1472e-09)	1:25:45	4.4778
ASO-MOP7	**1.4400e-09** **(1.2175e-11)**	1.4330e-09 (1.4990e-11)	6:05:15	4.6737

Table 31: Comparison of the hypervolume performance measure (Hv) for MODE-LD+SS and pMODE-LD+SS in solving aerodynamic shape optimization problems.

8 METAMODEL-ASSISTED MULTI-OBJECTIVE EVOLUTIONARY OPTIMIZATION

8.1 INTRODUCTION

In real-world engineering problems, there are many areas in which designers are trying to achieve several goals or objectives simultaneously. For instance, in the aeronautical engineering industry, given an aircraft's mission and their respective design environmental operating conditions, some common objectives or goals to be minimized include aircraft's weight, cost, and fuel consumption. While all of these objectives are aimed to be minimized, at the same time, aircraft's performance and safety are goals or objectives to be kept at a maximum. From the above, it should be evident that engineers are continuously aiming to produce good trade-offs and producing designs that satisfy as many requirements as possible, while industry, commercial, and ecological standards simultaneously add tighter constraints.

Multi-Objective Evolutionary Algorithms (MOEAs) have become a popular search engine for solving multi-objective optimization problems in aeronautical engineering [10, 41]. However, MOEAs normally require a significant number of objective function evaluations in order to achieve a reasonably good approximation of the Pareto front, even when dealing with problems of low dimensionality. This is a major concern when attempting to use MOEAs for real-world applications, like the ones we are interested in solving as part of this thesis, since in many of them, we can only afford a fairly limited number of fitness function evaluations.

In spite of these concerns, relatively few efforts have been reported in the literature to reduce the computational cost of MOEAs, and several of them only focus on algorithmic complexity (see for example [103]), in which little else can be done because of the theoretical bounds related to nondominance checking [138].

It has been until relatively recently, that researchers have developed techniques to achieve a reduction of fitness function evaluations by exploiting knowledge acquired during the search [126]. Knowledge of past evaluations can also be used to build an empirical model that approximates the fitness function to optimize. This approximation can then be used to predict promising new solutions at a smaller evaluation cost than that of the original problem [126, 109]. Current functional approximation models include Polynomials (response surface methodologies [195, 74]), neural networks (e.g., multilayer perceptrons (MLPs) [87, 92, 187]), radial-basis function (RBF) networks [175, 243, 259], support vector machines (SVMs) [217, 18], Gaussian processes [244, 28], and Kriging [60, 196] models. Other authors have adopted fitness inheritance [200] or cultural algorithms [142] for the same purposes.

In this chapter we propose a surrogate-based approach, which is designed to be used in the context of multi-objective evolutionary optimization. The proposed approach is evaluated by solving different aerodynamic shape multi-objective optimization problems. Before presenting the proposed approach, we provide a general overview of surrogate modeling.

8.2 GENERAL OVERVIEW OF SURROGATE MODELING

The use of MOEAs to solve multi-objective optimization problems usually considers a large population size and a large number of generations. These two conditions make MOEAs an unaffordable choice (computationally speaking) in certain applications, even when parallelism is adopted. In general, MOEAs can be unaffordable for an application when:

- The evaluation of the fitness functions is computationally expensive (i.e., it takes from minutes to hours, or even days to weeks as indicated on some aeronautical engineering applications surveyed in Chapter 4).
- The fitness functions cannot be defined in an algebraic form (e.g., when the fitness functions are generated by a simulator).
- The total number of evaluations of the fitness functions is limited by financial constraints (i.e., there is a financial cost involved in computing the fitness functions).

Jin et al. [109] presented a taxonomy of approaches which incorporate knowledge into EAs (see Figure 72). From this taxonomy, we can distinguish three main types of strategies or approaches to deal with expensive fitness functions:

problem approximation: This approach tries to replace the original statement of the problem by one which is approximately the same as the original problem but which is easier to solve. To save the cost of the experiments, numerical simulations instead of physical experiments are used to pseudo-evaluate the performance of a design.

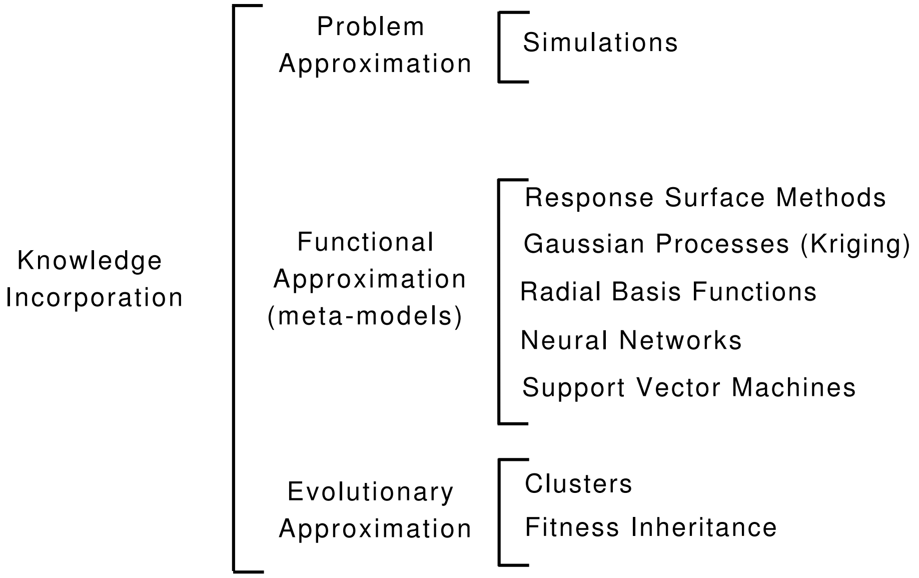


Figure 72: A taxonomy of approaches for incorporating knowledge into evolutionary algorithms

functional approximation: In this case, a new expression is constructed for the objective function based on previous data obtained from the real objective functions. The models obtained from the available data are often known as *meta-models* or *surrogates*.

evolutionary approximation: This approximation is specific for EAs and tries to save function evaluations by estimating an individual's fitness from other similar individuals. Two popular subclasses in this category are fitness inheritance and clustering.

In many practical engineering problems, we have black-box objective functions whose algebraic definitions are not known. In order to construct an approximation function, it is required to have a set of sample points that help us to build a meta-model of the problem. The objective of such meta-model is to reduce the total number of evaluations performed on the real objective functions, while maintaining a reasonably good quality of the results obtained. Thus, such meta-model is used to predict promising new solutions at a smaller evaluation cost than that of the original problem.

The accuracy of the surrogate model relies on the number of samples provided in the search space, as well as on the selection of the appropriate model to represent the objective functions. There exists a variety of techniques for constructing surrogate models (see for example [247]). Comparisons of several surrogate modeling techniques have been presented by Giunta and Watson [73] and by Jin et al. [108].

In this thesis we are interested in solving multi-objective aeronautical problems. In this context, commonly used surrogate-based approaches include: *(a)*.- Response surface methods based on polynomial approximation [150, 151, 152, 241, 185], *(b)*.- Gaussian process or Kriging [46, 224, 106, 253, 236, 237, 34, 36, 35, 137], *(c)*.- Radial basis functions (RBFs) [37, 43, 93], and *(d)*.- Artificial Neural Networks (ANNs) [193, 8, 3]. Next, we present a brief description of these approximation approaches.

8.2.1 *Response surface methods (RSM) based on polynomials*

The response surface methodology comprises three main components: (1) regression surface fitting, in order to obtain approximate responses, (2) design of experiments in order to obtain minimum variances of the responses and (3) optimizations using the approximated responses.

An advantage of this technique is that the fitness of the approximated response surfaces can be evaluated using powerful statistical tools. Additionally, the minimum variances of the response surfaces can be obtained using design of experiments with a small number of experiments.

For most response surfaces, the functions adopted for the approximations are polynomials because of their simplicity, although other types of functions are, of course, possible. For the cases of quadratic polynomials, the response surface is described as follows:

$$\hat{y} = (\beta_0) + \sum_{i=1}^{n} (\beta_i \cdot x_i) + \sum_{i,j=1,i\leqslant j}^{n} (\beta_{i,j} \cdot x_i \cdot x_j) \tag{49}$$

where n is the number of variables, and β_0 and β_i are the coefficients to be calculated. To estimate the unknown coefficients of the polynomial model, both the least squares method (LSM) and the gradient method can be used, but either of them requires at least the same number of samples of the real objective function than the β_i coefficients in order to obtain good results.

8.2.2 *Gaussian process or Kriging*

An alternative approach for constructing surrogate models is to use a Gaussian process model (also known as Kriging), which is also referred to as *Design and*

Analysis of Computer Experiments (DACE) model [206] and Gaussian process regression [258]. This approach builds probability models through sample data and estimates the function values at every untested point with a Gaussian distribution.

In Kriging, the meta-model prediction is formed by adding up two different models as follows:

$$y(\overrightarrow{x}) = a(\overrightarrow{x}) + b(\overrightarrow{x}) \tag{50}$$

where $a(\overrightarrow{x})$ represents the "average" long-term range behavior and the expected value of the true function. This function can be modeled in various ways, such as with polynomials or with trigonometric series such as:

$$a(\overrightarrow{x}) = a_0 + \sum_{i=1}^{L} \sum_{j=1}^{R} a_{ij} (x_i)^j \tag{51}$$

where R is the polynomial order with L dimensions and $b(\overrightarrow{x})$ stands for a local deviation term. $b(\overrightarrow{x})$ is a Gaussian random function with zero mean and non-zero covariance that represents a localized deviation from the global model. This function represents a short-distance influence of every data point over the global model. The general formulation for $b(\overrightarrow{x})$ is a weighted sum of N functions, $K_n(\vec{x})$ that represent the covariance functions between the n^{th} data point and any point $\vec{x}$:

$$\begin{gathered} b(\overrightarrow{x}) = \sum_{n=1}^{N} b_n K(h(x, x_n)) \text{ and} \\ h(x, x_n) = \sqrt{\sum_{i=1}^{L} (\frac{x_i - x_{i,n}}{x_i^{max} - x_i^{min}}^2)} \end{gathered} \tag{52}$$

where x_i^{min} and x_i^{max} are the lower an upper bounds of the search space and $x_{i,n}$ denotes the $i - th$ component of the data point x_n. However, the shape of $K(h)$ has a strong influence on the resulting aspect of the statistical model. That is the reason why it is said that Kriging is used as an estimator or an interpolator.

8.2.3 *Radial basis functions*

Radial Basis Functions (RBFs) were first introduced by R. Hardy in 1971 [80]. Let's suppose we have certain points (called centers) $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_n \in \mathbb{R}^d$. The linear combination of the function φ centered at the points $\overrightarrow{x}$ is given by:

$$\hat{y} : \mathbb{R}^d \mapsto \mathbb{R} : \overrightarrow{x} \mapsto \sum_{i=1}^{n} \omega_i \varphi(\overrightarrow{x} - \overrightarrow{x_i}) = \sum_{i=1}^{n} \omega_i \varphi(\|\overrightarrow{x} - \overrightarrow{x_i}\|) \quad (53)$$

where $\|\overrightarrow{x} - \overrightarrow{x_i}\|$ is the Euclidean distance between the points $\overrightarrow{x}$ and $\overrightarrow{x}_i$. So, φ becomes a function which is in the finite dimensional space spanned by the basis functions:

$$\varphi_i : \overrightarrow{x} \mapsto \varphi(\|\overrightarrow{x} - \overrightarrow{x_i}\|)$$

Now, let's suppose that we already know the values of a certain function $H : \mathbb{R}^d \mapsto \mathbb{R}$ at a set of fixed locations $\overrightarrow{x_i}, \ldots, \overrightarrow{x_n}$. These values are named $\hat{y}_j = H(\overrightarrow{x_j})$, so we try to use the $\overrightarrow{x_j}$ as centers in equation (53). If we want to force the function f to take the values f_j at the different points $\overrightarrow{x_j}$, then we have to put some conditions on the ω_i. This implies the following:

$$\forall j \in \{1, \ldots, n\}\ \hat{y}_j = \hat{y}(\overrightarrow{x_j}) = \sum_{i=1}^{n} \left(\omega_i \cdot \varphi(\|\overrightarrow{x_j} - \overrightarrow{x_i}\|)\right)$$

In these equations, only the ω_i are unknown, and the equations are linear in their unknowns. Therefore, we can write these equations in matrix form:

$$\begin{bmatrix} \varphi(0) & \varphi(\|x_1 - x_2\|) & \ldots & \varphi(\|x_1 - x_n\|) \\ \varphi(\|x_2 - x_1\|) & \varphi(0) & \ldots & \varphi(\|x_2 - x_n\|) \\ \vdots & \vdots & & \vdots \\ \varphi(\|x_n - x_1\|) & \varphi(\|x_n - x_2\|) & \ldots & \varphi(0) \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (54)$$

Typical choices for the basis function $\varphi(\overrightarrow{x})$ include linear splines, cubic splines, multiquadratics, inverse multiquadratics, thin-plate splines and Gaussian functions as shown in Table 32.

Type of Radial Function		
LS	linear splines	r
CS	cubic splines	r^3
MQS	multiquadratics	$\sqrt{r^2 + \sigma^2}$
IMQS	inverse multiquadratics	$\frac{1}{\sqrt{r^2+\sigma^2}}$
TPS	thin plate splines	$r^2 \ln(r)$
GA	Gaussian	$e^{-r^2/(2\sigma^2)}$

Table 32: Radial basis function kernels

8.2.4 *Artificial neural networks (ANN)*

An ANN basically builds a map between a set of inputs and the corresponding outputs, and are good to deal with nonlinear regression analysis with noisy signals [20]. A multilayer feedforward neural network consists of an array of input nodes connected to an array of output nodes through successive intermediate layers. Each connection between nodes has a weight, which initially has a random value, that is adjusted during a training process. The output of each node of a specific layer is a function of the sum on the weighted signals coming from the previous layer. The crucial points in the construction of an ANN are the selection of inputs and outputs, the architecture of the ANN, that is, the number of layers and the number of nodes in each layer, and finally, the training algorithm.

The multi-layer perceptron (MLP) is a multilayered feedforward network that has been widely used in function approximation problems, because it has been found to provide compact representations of mappings in a variety of real-world problems. An MLP is composed of neurons and the output (y) of each neuron is thus:

$$y = \phi\left(\sum_{i=1}^{n} w_i \cdot a_i + b\right)$$

where a_i are the inputs of the neuron, and w_i is the weight associated with the i^{th} input. The nonlinear function ϕ is called the activation function as it determines the activation level of the neuron.

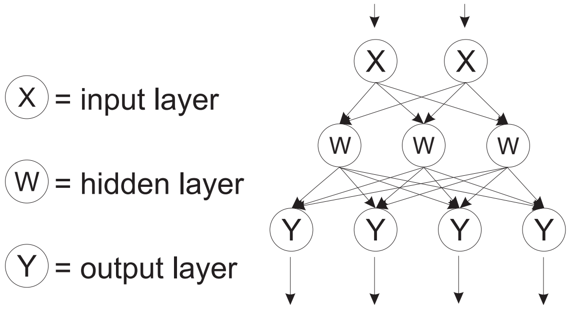


Figure 73: A graphical representation of an MLP network with one hidden layer

In Figure 73, we show an MLP network with one layer of linear output neurons and one layer of nonlinear neurons between the input and output neurons. The middle layers are usually called *hidden layers*.

To learn a mapping $\mathbb{R}^n \rightarrow \mathbb{R}^m$ by an MLP, its architecture should be the following: it should have n input nodes and m output nodes with a single or multiple hidden layer(s). The number of nodes in each hidden layer is generally a design decision.

8.3 METAMODEL ASSISTED MULTI-OBJECTIVE EVOLUTIONARY OPTIMIZATION IN AERONAUTICAL ENGINEERING

In this section, we present some selected research work in which an aeronautical/aerospace multi-objective engineering optimization problem was solved using a MOEA coupled to a technique for reducing the computational cost involved. In this type of engineering problems, designers commonly encounter: High CPU time demand, high nonlinearities and, some times, also high dimensionality. All of these features are also common in other engineering optimization problems, and we consider them representative of the main sources of difficulty in engineering optimization in general. Among the problems identified in aeronautical/aerospace engineering, having costly evaluations are the following:

- **Aerodynamic Shape Optimization**: This type of optimization problem ranges from 2D to complex 3D shapes. Typical optimization applications for 2D problems comprise wing and turbine airfoil shape optimization as well as inlet/nozzle design optimization, whereas for 3D problems, turbine blade, wing shape and wing-body configuration design optimizations are typical example applications.

- **Structural Optimization**: The aeronautical/aerospace design philosophy focuses on the design of structures with minimum weight that are strong enough to withstand certain design loads. These two objectives are conflicting in nature and, therefore, the aim of structural optimization is to find the best possible compromise between them. Typical applications for this type of problems comprise structural shape and topology optimization, robust structural design and structural weight optimization.

- **Multidisciplinary Design Optimization**: aeronautical/aerospace design has a multidisciplinary nature, since in many practical design applications, two or more disciplines are involved, each one with specific performances to accomplish. Typical applications for this type of problems are the aeroelastic applications in which aerodynamics and structural engineering are the interacting disciplines.

For all the optimization problems indicated above, the objective function evaluations are routinarily done by using complex computational simulations such as CFD (Computational Fluid Dynamics) in the case of aerodynamic problems, CAA (Computational Aero-Acoustics) for aero-acoustic problems, CSM (Computational Structural Mechanics, by means of Finite Element Method software) for structural optimization problems, or a combination of them in the case of multidisciplinary design optimization problems. Because of their nature, any of these computational simulations have a high computational cost (since they solve, in an iterative manner, the set of partial differential equation governing the physics of the problem) and evaluating the objective functions for the kind of problems indicated above, can take from minutes to hours for a single candidate solution, depending on the fidelity of the simulation.

Given the high computational cost required for the computer simulations and the population based nature of MOEAs, the use of hybrid methods or meta-models is a natural choice in order to reduce the computational cost of the design optimization process, as indicated by some representative research works that will be described next. It is important to recall to the reader that from the universe of aeronautical/aerospace applications indicated above, our main interest is in solving aerodynamic shape optimization problems like the ones described in Chapter 6 where we defined a set of ASO-MOPs. Thus, we will limit our discussion to the use of surrogate models in multi-objective ASO problems, since that is the scope of this research work.

8.3.1 *Multi-objective aerodynamic shape optimization*

In aeronautical systems design as well as in the design of propulsion system components, such as turbine engines, aerodynamics plays a key role. Thus, aerodynamic shape optimization (ASO) is a crucial task, which has been extensively studied and developed. By its nature, ASO is a MOP. With the current state-of-the-art in computing power and the recent advances in CFD, this discipline has become an integral part of the ASO process. CFD has been employed to cut aerodynamic design cost and time scales by reducing the number of required experiments. This discipline has recently benefited from the use of MOEAs, which have gained an increasing popularity in the last few years [10]. However, MOEA application to ASO problems still remains as a formidable challenge because of the following reasons:

1. The flow field for some ASO applications, can be extremely complex. Therefore, complex CFD Navier-Stokes computations (which are very expensive, computationally speaking) are required.

2. The performance of aerodynamic shapes such as wing's airfoils or turbine airfoils blades, is very sensitive to the shape itself. Thus, an airfoil must be modeled with a large number of decision variables. In addition, the objective function landscape of an ASO problem is often multimodal and nonlinear because the flow field is governed by a system of nonlinear partial differential equations.

3. ASO problems are usually subject to several constraints and in some cases, such constraints can be evaluated only after performing a CFD simulation, turning it into a very expensive process (computationally speaking). In cases where restrictions are geometrical ones, most researches decide to continuously generate new solutions until a valid geometric design is generated. However in other cases, constraints can only be evaluated, after the CFD simulation is done. This latter condition can render the constraint evaluation, also computationally intensive.

4. MOEAs require a considerable number of fitness function calls to the CFD simulation code in order to conduct an appropriate search. This may turn them impractical if the objective functions are too costly.

Considering that both, multi-objective optimization using evolutionary algorithms, and CFD technology are mature areas, the challenges listed above can be considered as technical hurdles. To tackle them, there is an evident need to have mechanisms that allow the solution of computationally expensive problems in reasonably short periods of time, mainly by reducing the required number of objective function evaluations. A common approach is the use of parallel processing techniques, which, however, may not be sufficient in some cases. Examples of these conditions are presented by Benini [15], who reported computational times of 2000 hrs. in the multi-objective re-design of a transonic turbine rotor blade, using a population with 20 design candidates, and 100 generations of evolution time, in a four-processors workstation. Another alternative that has been widely adopted in the engineering optimization literature is the use of surrogates (also called metamodels), which use (computationally cheap) approximate models of the problem which are periodically adjusted (using real objective function evaluations).

8.3.2 *Guidelines for design/using surrogate modeling approaches*

We are interested in the use of surrogate models (metamodels) [24, 114, 206] coupled to MOEAs for solving ASO MOPs. In this context, surrogate models [24] replace direct calls to any CFD simulation code. Figure 74 shows the general flow in any surrogate-based optimization approach.

When designing and/or using a surrogate modeling approach several issues need to be addressed [71]. Among them, we describe next four major issues to be aware of:

1. **Model to use:** The first issue is to select which mathematical model will be used for the surrogate. Several options are available: RSM based on low-order polynomial functions, Gaussian processes or Kriging, RBFs, ANNs, and SVMs, among others [109]. The interested reader is referred to Jin [109] for a comprehensive review of these and other approximation techniques, and their use in the context of EAs.

2. **Global/local surrogate model:** This choice encompasses a multi-objective decision problem. A global approximation model, associated with a reduced accuracy can be designed with a better ability to reflect general tendencies in the fitness landscape, allowing the designer to perform an explorative design search in the whole design space in an efficient and rapid manner. When a local surrogate model is adopted, the accuracy of the approximation can be increased with a better ability to capture local tendencies in the fitness landscape, but its region of validity is limited to a predefined neighborhood in the design space, and the designers are able to explore only small regions of it. This latter condition, in some cases,

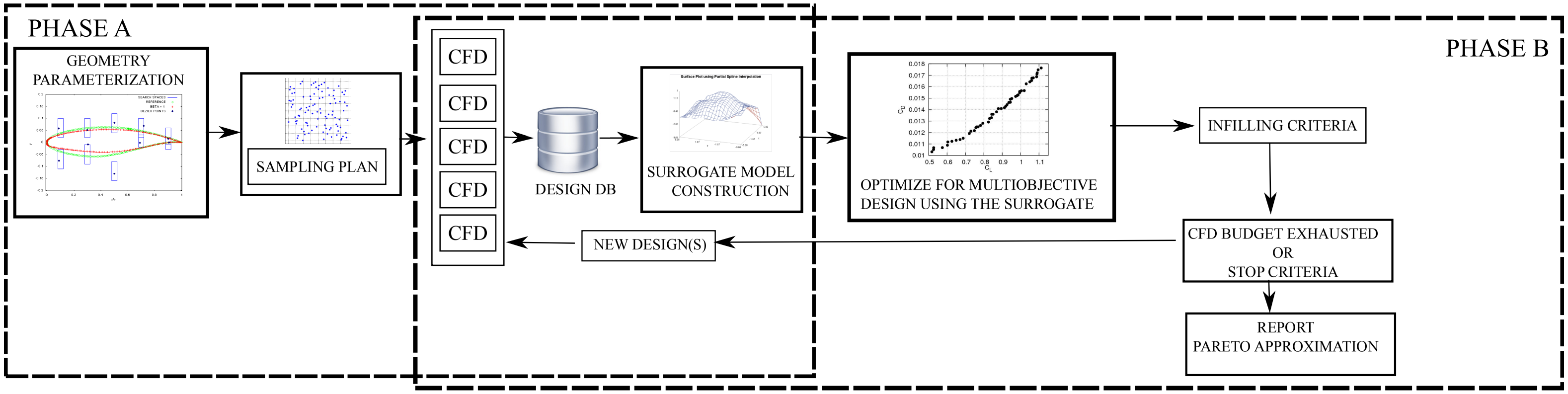

PHASE A
GEOMETRY PARAMETERIZATION
SAMPLING PLAN
CFD
CFD
CFD
CFD
CFD
DESIGN DB
SURROGATE MODEL CONSTRUCTION
OPTIMIZE FOR MULTIOBJECTIVE DESIGN USING THE SURROGATE
INFILLING CRITERIA
CFD BUDGET EXHAUSTED OR STOP CRITERIA
REPORT PARETO APPROXIMATION
NEW DESIGN(S)
PHASE B

Figure 74: Surrogate-based optimization framework

might also hinder the ability of any optimization method in looking for new and better designs.

3. **Sample size and distribution for initial surrogate training:** Independently of the designer's choice upon a local or a global surrogate model, an issue is that the metamodel must first be trained using a number of initial simulations, whose evaluation is costly (since the high cost CFD simulation code will be used). These initial points are defined by a design of experiments (DoE) technique [206], and must be kept to a minimum; otherwise, the limited budget of calls to the CFD simulation code can be exhausted only on the surrogate model initial training. Since, generally, there is no prior knowledge of the shape of the fitness landscape, either globally or locally, the same amount of computing effort is normally applied via the evenly distribution of the initial simulations throughout the design space region. Several initial point distributions have been proposed. A relatively common approach is to use Latin Hypercube Sampling (LHS) [206].

4. **Infilling criterion:** Once we have found answers to the previous issues, we have arrived to a surrogate model on which we can perform an efficient multi-objective optimization, with a reduced computational cost as compared to the use of the CFD simulation cost. The outcome of this latter process will be a set of promising candidates to evaluate for finding the trade-off among the competing objectives defined in the original ASO problem. At this point, the designer/user will be required to decide which (surrogate-obtained) optimal designs will be evaluated with the real objective functions. These evaluations will be used to adjust the model (aiming to reduce its approximation error). Clearly, these so-called *infill points*, must be carefully selected (using a good *infilling criterion*). A judicious selection of the aforementioned *infill points* is called an *infilling criteria*, and require some additional information to make the decision on. The reader must keep in mind that any surrogate model has inherently an approximation error, which depends on their tuning parameters and on the training process used [251]. This will be of course the primary information to look for. To define an infilling criterion is not an easy task, since we aim not only at reducing the approximation errors, but also at exploring as many different regions of the search space as possible. Thus, accuracy and diversity need somehow to be balanced within our infilling criterion (this can be considered a multi-objective decision problem presented to the designer). For example, regions of design space

with lower approximation errors allow designers an exploitation of the surrogate model. It must be clearly sound that exploitation of any surrogate model must be preceded of its accuracy improvement. This latter condition calls for evaluating solutions when the surrogate model has high approximation errors or in regions of the design space which have been poorly explored, i.e., an exploration of the surrogate model needs to be performed. But since we are interested in solving MOPs, the exploration/exploitation processes must also consider the diversity of solutions along the approximation of the Pareto Front, which can only be assessed, once the costly CFD simulations are performed. With all these conditions, we clearly see that a general rule in using any surrogate modeling approach, is to judiciously select a good infilling criterion that balances the exploration/exploitation of the surrogate model to perform the multi-objective optimization with a limited budget on the calls to the CFD simulation code.

The above list of issues is by far exhaustive, since many other practical implications can be devised and in some cases are problem dependent. However, it might be clear that in summary, any attempt in designing and/or using surrogate modeling for ASO problems must try to tackle the above issues, and to leverage their effects in a holistic manner.

8.3.3 *Surrogate modeling in ASO problems*

Next, we present a short review of some representative research work on the use of surrogate modeling for solving ASO problems. The way in which the issues indicated before are addressed in each paper is emphasized in our discussion. We identify how the researchers have solved the above mentioned issues.

Mathematical model used:

From the different mathematical models available, RSM based on low order polynomials are probably the most popular choice in the literature. For example, Lian and Liou [151] presented the use of MOGA [68] coupled to a second order polynomial based RSM, for solving the bi-objective ASO of a turbine blade. This is probably a natural extension of methods that have been found to be effective for the single-objective case. The main advantage of using polynomial based RSM is probably its generalization abilities. However, its training cost is proportional to the number of sampling points, and a high number of them is required for getting a good accuracy of the model. More recently, RSM based on low order polynomials have become less common, probably due to their limitations for accurately representing fitness landscapes which are very rough and with high nonlinearities. For such cases, some researchers have relied on more elaborated surrogate models as in Karakasis et al. [116], who presented the multi-objective optimization of a turbine engine compressor blade using RBF surrogate models. RBFs are very powerful functions to represent complex fitness landscapes, and for some kernel functions, there exist some tuning parameters to control the accuracy of the approximation model. Emmerich et al. [62] and Keane [117] presented the application of a Kriging-based metamodel for multi-objective airfoil shape optimization in transonic flow conditions. Kriging has a strong mathematical basis, and is probably one of the most powerful interpolation methods currently available. Also, kriging is able to provide an estimate of its associated accuracy, and it allows the model to be tuned for an improved accuracy. However, its cost increases as the dimensionality and the number of training points in the problem increase.

Globality/Locality of the surrogate model:

Concerning the globality/locality of the model, the RSM presented by Lian and Liou [151] corresponds to a global one, while the model presented by Karakasis et al. [116] corresponds to multiple local models. In this case, the database containing the history of designs evaluated with the CFD tool, is subdivided into clusters, using a self-organizing map technique. This sort of technique aims at training local RBF surrogate models with small subpopulations, but also guaranteeing that the whole design space is covered by allowing the overlapping of the local RBF models. In the work of Emmerich et al. [62], the model is also local, i.e., local kriging models in the neighborhood of the solution are evaluated. In the work of Keane [117], the global kriging model is adopted.

Sampling and distribution for initial surrogate training:

Regarding the initial sampling technique, most of the research works that we reviewed use a DoE technique, based on either a particular LHS approach [151] (Improved Hypercube Sampling (IHS) algorithm [12]), or the $\mathrm{LP}\tau$ technique [117]. The size of the initial population is mainly a choice based both on the dimensionality of the problem, as well as on the number of CFD evaluations available.

Infilling criteria:

Finally, regarding the infilling criterion, we identified the following options. Lian and Liou [151], do not adopt an infilling criterion. In this case, the trained model, which comprised 1,024 design solutions, is used to obtain the approximation of the Pareto front. From this set, some candidate solutions are selected to be evaluated with the CFD simulation code. This is probably the simplest possible technique, but designers must rely on very accurate models in this case. In the works of Karakasis et al. [116], and Emmerich et al. [62], the infilling criterion corresponds to a prescreening technique. In this technique, the offspring are evaluated with the local models and with this evaluation they are either ranked based on Pareto dominance [116] or their Hypervolume contribution is estimated [62]. In both cases, the most promising individuals are selected to be evaluated with the CFD simulation code. One important aspect of the work of Emmerich et al. [62] is that the evaluation of the hypervolume contribution of each design includes the uncertainty in the objective function evaluation, as provided by the kriging model used. Finally, in the work of Keane [117], the infilling criterion corresponds to a multi-objective extension of a commonly used technique for single-objective kriging models. This approach consists in adopting a metric defined in terms of the probability of improvement, and on the expected improvement which can be computed from the estimated accuracy of interpolation given by the model. Again, this technique is, by far, the most elegant and the most sound from the mathematical point of view. However as previously indicated it suffers from a high computational cost for its evaluation, specially when the dimensionality in the problem is very high and/or the number of training points is also high.

8.4 OUR PROPOSED SURROGATE MODELING APPROACH

In this section we present a surrogate-based multiobjective evolutionary approach to optimize airfoil aerodynamic designs. Our approach makes use of multiple surrogate models which operate in parallel, aiming to combine the features of different approximation models in order to produce the combination that reduces, as much as possible, the computational cost of the MOEA being used. The proposed approach is tested on a set of five multiobjective ASO problems, defined from similar ones in the specialized literature. The application problems illustrate the ability of the proposed approach to reduce in a substantial manner, the computational cost in terms of the number of objective function evaluations performed.

Our proposed surrogate-based multiobjective evolutionary approach has the following features:

(a).- **Model:** When a single surrogate is needed, it is common practice to train several surrogate models and pick one based on their accuracy or their cross-validation error [78]. Another option is to combine different surrogate models into a single one by weighting their contribution [250]. In our case, and inspired by the notion of *"blessing and curse of uncertainty"* in approximation models [269], we propose to use not only one surrogate model, but a set of them. So, the idea is to train, in parallel, N surrogate models (SM_1,SM_2,...SM_N). Thus, N new solutions will be selected for updating the database and to update the set of surrogate models itself. The motivation behind this choice is also that, by using different models, which are trained with the same data points, we can inherently balance the exploration/exploitation required in any surrogate-based optimization technique. Solutions selected from a surrogate model with high accuracy will emphasize exploitation, while solutions selected from a surrogate with low accuracy will emphasize exploration. Towards the end of the evolutionary process, as more data points are available to train the surrogate, it is expected that all surrogate models will have a high accuracy and will, therefore, contribute more to the exploitation. Also, by performing the search in parallel, we can reduce the computational cost associated with the MOEA being used.

In our approach, the surrogate models can be any combination of the options previously mentioned. Alternatively, it is also possible to use a single surrogate model but with different tuning parameters which will be tested in parallel. In the examples that we will present, we use a set of RBF surrogate models each one with a different kernel function.

(b).- **Globality/locality:** In our proposed approach we decided to adopt a global model, but this globality is defined in terms of the training points in the database. Thus, the models are trained in the design space implicitly defined by the database points. We propose to define an initial number of training points NTP^{init}, and a maximum allowable number of training points to hold in the database NTP^{max}. Once this upper limit is reached, we still allow the insertion of new points, but the database is pruned until reaching again the maximum allowable number of solutions. The pruning technique adopted here is based on Pareto ranking (i.e., individuals in the database with the highest Pareto ranks, which are the worst in terms of Pareto optimality, are removed, until reaching the upper limit allowed in the database). The motivation for defining a maximum number of points to be kept in the database, is to reduce the computational cost associated to the training process, and to adapt the model in the neighborhood of the Pareto front approximation, as the evolution progresses.

(c).- **Initial sampling:** In the experiments that we present here, we adopted a sampling procedure based on the Halton distribution of points. Regarding the initial database size $\mathsf{NTP}^{\mathsf{init}}$, this will depend on the budget of CFD evaluations available and on the number of dimensions of the problem. We will provide guidelines for this, later on.

(d).- **Infilling criterion:** In our approach, we adopt N parallel surrogate models, each of which is searched in parallel for Pareto solutions using a MOEA; and we extract one solution from each of them. Therefore, N new solutions will be generated at each design cycle, which will also be evaluated in parallel. For performing this selection, we first define a set of weight vectors $[\lambda^1, \lambda^2, \dots, \lambda^{NP}]$, where NP is the population size used in the MOEA. This set can be ordered according to the weight vector components λ_i^j to reflect an order in the tradeoffs among the objectives. Next, and from each surrogate model, we select a (surrogate-based) solution that minimizes a scalar function for a selected weight vector. For that sake, we adopt the Tchebycheff scalarization function given by:

$$g(\vec{x}|\lambda, z^*) = \max_{1 \leqslant i \leqslant k} \{\lambda_i^j |f_i(x) - z_i^*|\} \tag{55}$$

In the above equation, $\lambda^j, j = 1, \dots, NP$ represents the set of weight vectors used to distribute the solutions along the Pareto front. z^* corresponds to a reference point, defined in objective space and determined with the minimum objective values of the population used in the MOEA. In order to cover the whole Pareto front, each surrogate model must choose a different weight vector from the set. Additionally, in each design cycle, a different weight vector must be selected. For doing this, we perform, in each surrogate model, a sweeping in the set of weight vectors, and start from a different weight vector. This process is illustrated in Figure 75. The indicated sweeping is done in a cyclic manner, i.e., once the last weight vector is selected, the next one is picked from the beginning. The aim of this technique is to guarantee the coverage of all regions of the Pareto front. Evidently, once the N solutions are selected, they are evaluated with the CFD code and added to the database, and the approximation of the Pareto front is updated.

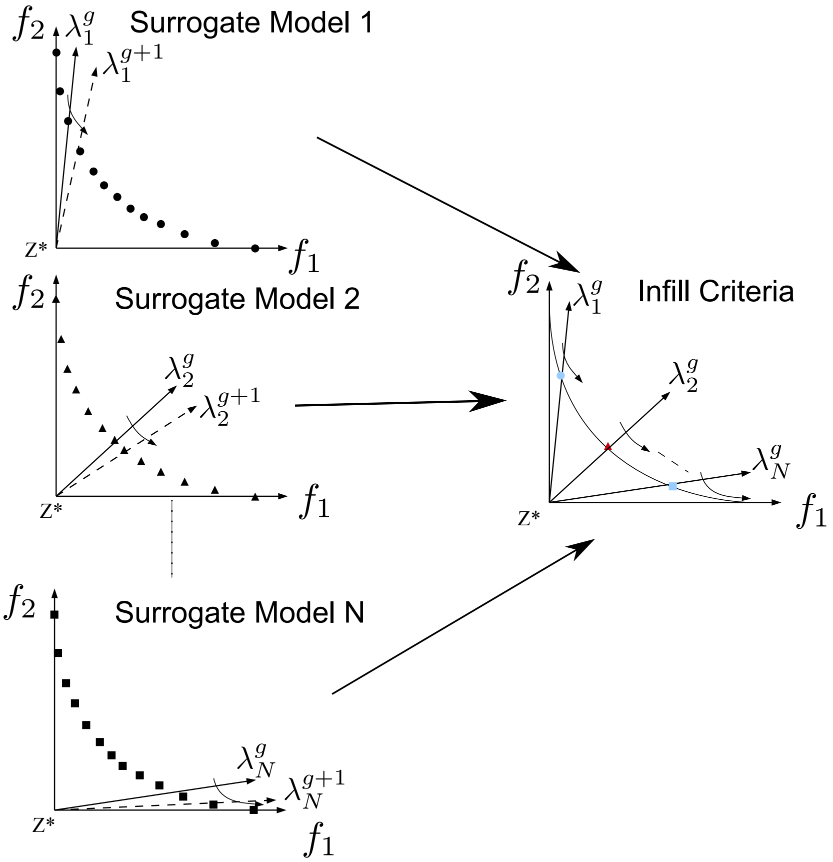


Figure 75: Infill criteria adopted by our proposed approach

8.5 EVALUATION OF THE PROPOSED APPROACH

In this section we present the evaluation of our proposed surrogate modeling approach presented in the previous section. Since our main interest is in its application to solve aerodynamic shape optimization problems, we report here its application to five of seven ASO-MOPs defined in Chapter 6. The corresponding ASO-MOPs used for validating our proposed surrogate modeling approach are: *ASO-MOP1* to *ASO-MOP5*. We use them to test the performance of the present surrogate approach when dealing with this type of real-world applications. Next, and for prompt reference, we summarize the experimental setup adopted.

8.5.1 *Experimental setup*

From the five ASO-MOPs selected, three are bi-objective, and the other two have three objectives each. We recall that we are interested in designing airfoil shapes ($\mathsf{AirfGeom}$), with optimal values in their aerodynamics forces and moments, and for different operating conditions. In airfoil design and analysis, it is common to define these forces and moments as scalar coefficients (see Section 6.3.1). It follows that for an airfoil shape ($\mathsf{AirfGeom}$) at a given flow incidence angle (α), the lift (C_l), drag (C_d), and pitching moment (C_m) are a function of:

$$[C_l, C_d, C_m]_{\mathsf{AirfGeom}} = f(\alpha, Re, M) \tag{56}$$

where the Reynolds number (Re) is the dimensionless ratio of the inertial forces to viscous forces and quantifies their respective relevance for a given operating condition. The Mach number (M) is a measure of the air velocity against the speed of sound. The CFD solver adopted in this benchmark corresponds to XFOIL [56]. These coefficients, as well as their ratio in some cases, have different effects on aircraft performance. Thus, for the benchmark we defined the following ASO MOPs, aiming at presenting different Pareto front geometries, as well as different fitness landscapes.

ASO-MOP1:

min(C_d) @ $\alpha = 0.0^o$, $Re = 4.0 \times 10^6$, $M = 0.2$

min($2.0 - C_l$) @ same flow conditions.

ASO-MOP2:

min(C_d/C_l) @ $\alpha = 4.0^o$, $Re = 2.0 \times 10^6$, $M = 0.1$

min(C_m^2) @ same flow conditions.

ASO-MOP3:

min(C_d/C_l) @ $\alpha = 1.0^o$, $Re = 3.0 \times 10^6$, $M = 0.3$

min(C_d^2/C_l^3) @ $\alpha = 5.0^o$, $Re = 1.5 \times 10^6$, $M = 0.15$

ASO-MOP4:

$\min(C_d)$ @ $\alpha = 4.0^o$, $Re = 3.0 \times 10^6$, $M = 0.3$

$\min(2.0 - C_l)$ @ same flow conditions.

$\min(C_m^2)$ @ same flow conditions.

ASO-MOP5:

$\min(C_d/C_l)$ @ $\alpha = 1.0^o$, $Re = 4.0 \times 10^6$, $M = 0.3$

$\min(C_d^2/C_l^3)$ @ $\alpha = 3.0^o$, $Re = 3.0 \times 10^6$, $M = 0.3$

$\min(C_d^2/C_l^3)$ @ $\alpha = 5.0^o$, $Re = 2.0 \times 10^6$, $M = 0.3$

8.5.2 *Geometry parameterization*

We adopt here the PARSEC airfoil representation [223]. Figure 76 illustrates the 11 basic parameters used for this representation. In our case, a modified PARSEC geometry representation was adopted, allowing us to define independently the leading edge radius, both for upper and lower surfaces. Thus, 12 variables in total were used. Their allowable ranges are defined in Table 33.

Parameter	Lower bound	Upper bound
r_{leup}	0.0085	0.0126
r_{lelo}	0.0020	0.0040
α_{te}	7.0	10.0
β_{te}	10.0	14.0
Z_{te}	-0.0060	-0.0030
ΔZ_{te}	0.0025	0.0050
X_{up}	0.4100	0.4600
Z_{up}	0.1100	0.1300
Z_{xxup}	-0.90	-0.70
X_{lo}	0.20	0.26
Z_{lo}	-0.0230	-0.0150
Z_{xxlo}	0.05	0.20

Table 33: Parameter ranges for PARSEC airfoil representation

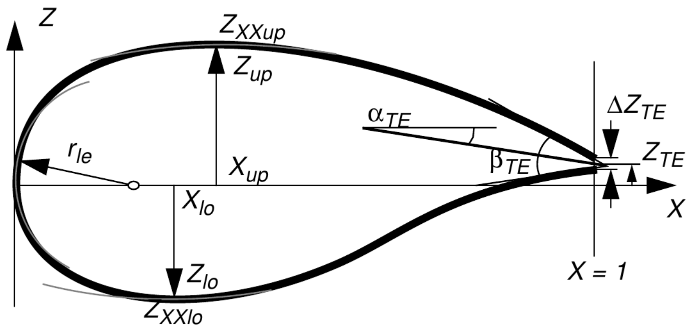


Figure 76: PARSEC airfoil parameterization

The PARSEC airfoil geometry representation adopted here uses a linear combination of shape functions for defining the upper and lower surfaces. These linear combinations are given by:

$$Z_{\text{upper}} = \sum_{n=1}^{6} a_n x^{\frac{n-1}{2}} \tag{57}$$

$$Z_{\text{lower}} = \sum_{n=1}^{6} b_n x^{\frac{n-1}{2}} \tag{58}$$

In the above equations, the coefficients a_n, and b_n are determined as a function of the 12 described geometric parameters, by solving two systems of linear equations (for a detailed description see Section 6.3.2).

For solving the above ASO-MOPs, we adopted our proposed MODE-LD+SS [9] as the search engine. The details of this MOEA, which has been compared to state-of-the-art MOEAs, can be found in Chapter 5. It is interesting to note that, since this MOEA already uses a weight vector set in its selection mechanism, coupling it to a surrogate model is straightforward by adopting the infilling criterion described in Section 8.4. Since we were only interested in evaluating the role of our performance modelling scheme, we only compared the results obtained by our original MODE-LD+SS with respect to those of the version that incorporates surrogate modelling.

8.5.3 *Performance measures*

In the context of MOEAs, it is common to compare results on the basis of some performance measures [41]. Next, and for performance assessment purposes, we report the hypervolume ($H\nu$) values attained by each of the two MOEAs compared (with and without surrogate modelling). A definition of this performance measure is given in Section 6.4, and is omitted here.

8.5.4 *Parameters settings*

Surrogate Model: As previously indicated, we used a set of surrogate models. For this benchmark, we adopted five RBF models defined by the following kernels:

- Cubic: $\varphi(r) = r^3$
- Thin Plate Spline: $\varphi(r) = r^2 \ln(r)$
- Gaussian: $\varphi(r) = e^{-r^2/(2\sigma^2)}$
- Multiquadratic: $\varphi(r) = \sqrt{r^2 + \sigma^2}$
- Inverse Multiquadratic: $\varphi(r) = 1/\sqrt{r^2 + \sigma^2}$

Above, $r = ||x - c_i||$, $c_i, i = 1, 2, \ldots, h$, is the center for the RBF, and h is the number of hidden layers. The first two RBF models contain no tuning parameters, while in the other three the σ parameter can be adjusted to improve the model accuracy. All the models are trained with the points stored in the actual database. For each model, the approximated function is defined by:

$$\hat{y}_{SM} = \Sigma_{j=1}^{h} \omega_j \varphi_j(x) \tag{59}$$

We used a value of $h = 20$ for the number of hidden layers. $\varphi(r)$ is the kernel of the hidden layer, and ω_j is the weighting coefficient. Since h is less than the number of training points in the database, we adopted a $K - means$ clustering technique to obtain the respective center for each hidden layer. The training process for each RBF model, required the determination of the weighting parameters ω_j by means of:

$$[\omega_1, \omega_2, \ldots, \omega_p]^T = (H^T H)^{-1} H^T Y_S \tag{60}$$

where Y_S corresponds to the vector of the objective functions for the sample points, and

$$H = \begin{bmatrix} \varphi_1(X_1) & \varphi_2(X_1) & \cdots & \varphi_p(X_1) \\ \varphi_1(X_2) & \varphi_2(X_2) & \cdots & \varphi_p(X_2) \\ \cdots & \cdots & \vdots & \cdots \\ \varphi_1(X_{NTP}) & \varphi_2(X_{NTP}) & \cdots & \varphi_p(X_{NTP}) \end{bmatrix} \tag{61}$$

MOEA parameters: We used the following set of parameters.

Parameters for MODE-LD+SS without surrogate modelling:

$F = 0.5$, $CR = 0.5$, $NB = 5$, $GMAX = 20$ and $NP = 100$ for the bi-objective problems. We only changed $NP = 120$ for the problems with three objectives. We defined a budget of 2000 objective function evaluations (OFEs). This was based on the OFEs commonly reported in the specialized literature for the problems of our interest, which range from 1000 [62] to 2000 [151].

Parameters for MODE-LD+SS with surrogate modelling:

$F = 0.5$, $CR = 0.5$, $GMAX = 100$ and $NP = 300$ (a higher population size is adopted, because the evolutionary process is performed on the surrogate model, and, therefore, has a low computational cost) for both, the bi-objective and the three objective cases. The number of cycles in the surrogate approach was adjusted for performing a total of 2000 OFEs.

Weight vector index for the infilling criterion: The index was defined in terms of the current iteration or generation (gen) using the following expression:

$$Weight\_Index = (SM_i - 1) \times \frac{NP}{N} + Shift \times (gen - 1) \tag{62}$$

From this expression, we can observe that each surrogate model starts the infilling criterion at a different weight vector, and then the whole set of vectors is swept during the evolutionary process. In this equation, NP is the population size used for the MOEA when the surrogate model is searched for, N is the number of surrogate models adopted, gen is the current generation number and $Shift$ is a constant used for the sweeping process defined in the infilling criterion. In our experiments, this constant was set to 13 in order to minimize the number of times that a weight vector is selected during the evolutionary process.

Initial and maximum training points in the database: We adopted: $\mathsf{NTP}^{\mathsf{init}} = 200$ and $\mathsf{NTP}^{\mathsf{max}} = 300$ for both, the bi-objective and the three objective problems. Here we propose to set this parameter to approximately twice the number of points corresponding to the population size of the MOEA, when no surrogate model is used. The upper limit aims at reducing the training cost associated to the RBF models, specially for those where the tuning parameters are adjusted for improving their accuracy.

8.5.5 *Results and discussion*

Table 34 summarizes the results obtained for the five ASO MOPs adopted and for each of the two MOEAs compared. In this table, the average HV measure, and its standard deviation are obtained from 32 independent runs for each MOP. The Hv measures shown here correspond to a total of 2000 real objective function evaluations. From this table, we can observe that the surrogate model approach consistently obtained better values than the approach not using it, both for the HV mean value and for their standard deviation. According to a Wilcoxon rank-sum statistical test [257] with a significance level of 0.05, for all the ASO-MOPs, the surrogate approach was significantly better.

	MODE-LD+SS		MODE-LD+SS w/s	
MOP	Hv Mean	Std Dev	Hv Mean	Std Dev
ASO-MOP1	5.6593E-04	5.3074E-06	**5.8790E-04**	4.3753E-06
ASO-MOP2	6.3550E-04	6.2108E-06	**6.4942E-04**	2.1426E-06
ASO-MOP3	1.6747E-06	7.4601E-08	**1.9076E-06**	1.7524E-08
ASO-MOP4	4.3639E-04	3.4658E-06	**4.4202E-04**	2.5353E-06
ASO-MOP5	5.8814E-09	3.0200E-10	**7.0975E-09**	6.0863E-11

Table 34: Hypervolume performance measure obtained for the different ASO-MOPs, without and with the proposed metamodel assisted approach.

In order to better analyze the impact of the proposed surrogate model approach, in Figures 77 through 81, we present, for all the test cases adopted, the Hv measure convergence plots, and the Pareto fronts approximations obtained after 1000 OFEs. From these convergence plots we can observe that, in general, at the beginning of the evolutionary process, the proposed surrogate model approach, has very good convergence properties. Considering as a first stage the first 500 OFEs, a high improvement of the Hv measure is achieved by the surrogate model approach. In fact, in all cases, except for the ASO-MOP4 problem, the surrogate model approach attains a Hv measure similar to that obtained after performing 2000 OFEs with the MOEA that does not incorporate a surrogate approach. For the case of the ASO-MOP4 test problem, about 50 additional OFEs are required for attaining the same state. Thus, if we consider the Hv measure improvement, we can estimate that our proposed surrogate-based optimization approach can produce savings of about 75% in the number of OFEs performed. This sort of savings can be considered as significant for the type of application being analyzed, because it translates into very important CPU time reductions.

Taking a closer look at the convergence plots for ASO-MOP1 and ASO-MOP3, after approximately 350 and 400 OFEs, respectively, we can observe that, prior to these points, the convergence rate clearly shows a tendency to be substantially reduced and probably even to stagnate. However, after these points, the convergence rate suddenly increases. This behavior can be explained in part by the combined action of the exploration/exploitation abilities of the different models incorporated in our approach. A more detailed analysis is, however, required, to confirm our hypothesis. After performing 500 OFEs, our proposed approach continues to show an improvement in the HV measure, but the rate has considerably reduced. Nevertheless, our approach is still able to consistently attain higher values than those achieved by the MOEA without surrogate modelling.

Looking at the Pareto front approximations, compared at an intermediate stage of 1000 OFEs, we can also observe that, in general, our proposed approach is able to, consistently, improve convergence towards the true Pareto front, and to cover a wider area along it. The first condition is clearly exemplified in the ASO-MOP3 problem, while the second condition is clearly seen on the ASO-MOP1 and the ASO-MOP2 problems. These same conditions apply to the problems with three objectives.

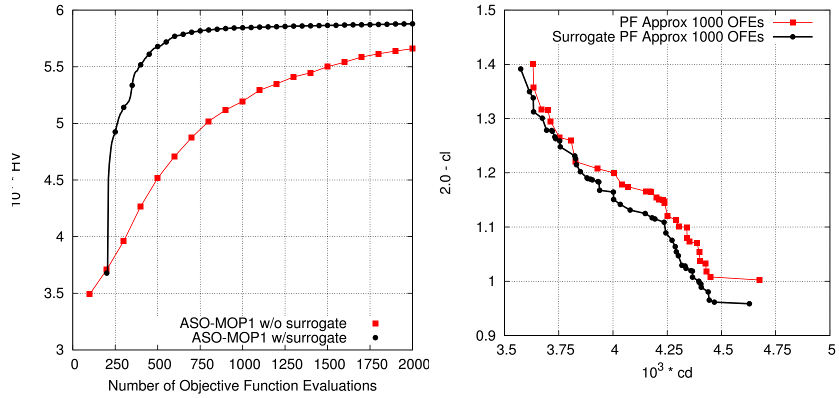


Figure 77: Metamodel assisted Pareto front approximation for ASO-MOP1

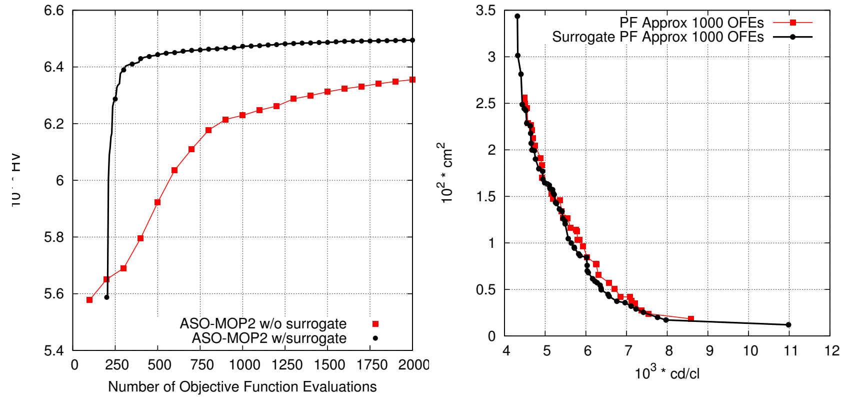


Figure 78: Metamodel assisted Pareto front approximation for ASO-MOP2

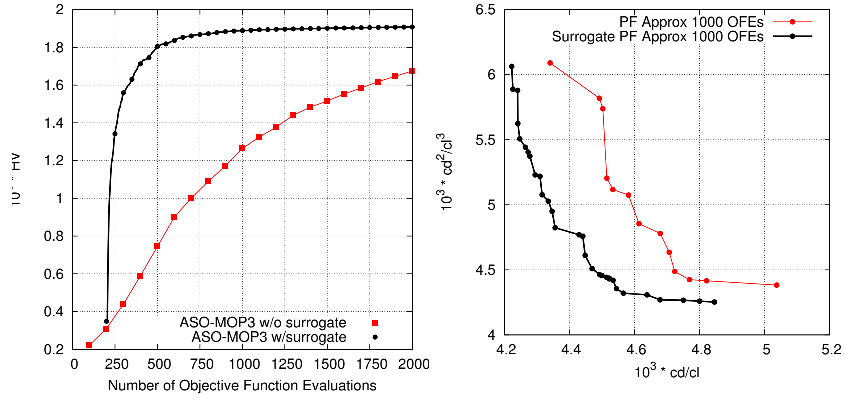


Figure 79: Metamodel assisted Pareto front approximation for ASO-MOP3

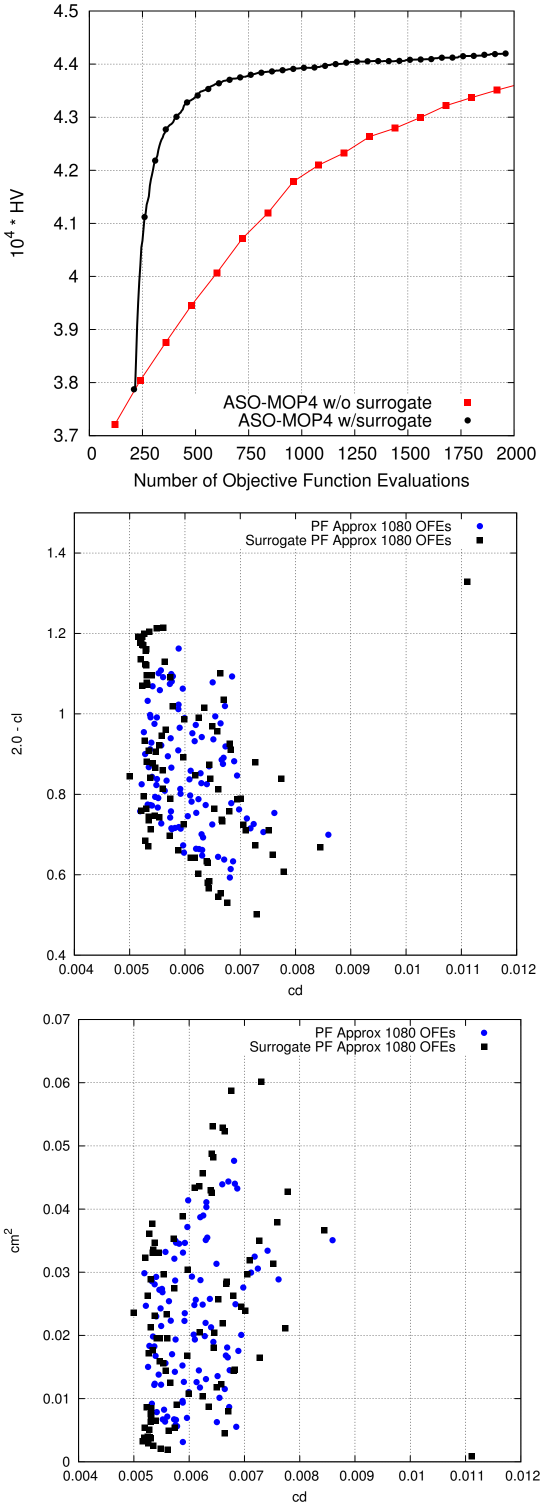


Figure 80: Metamodel assisted Pareto front approximation for ASO-MOP4

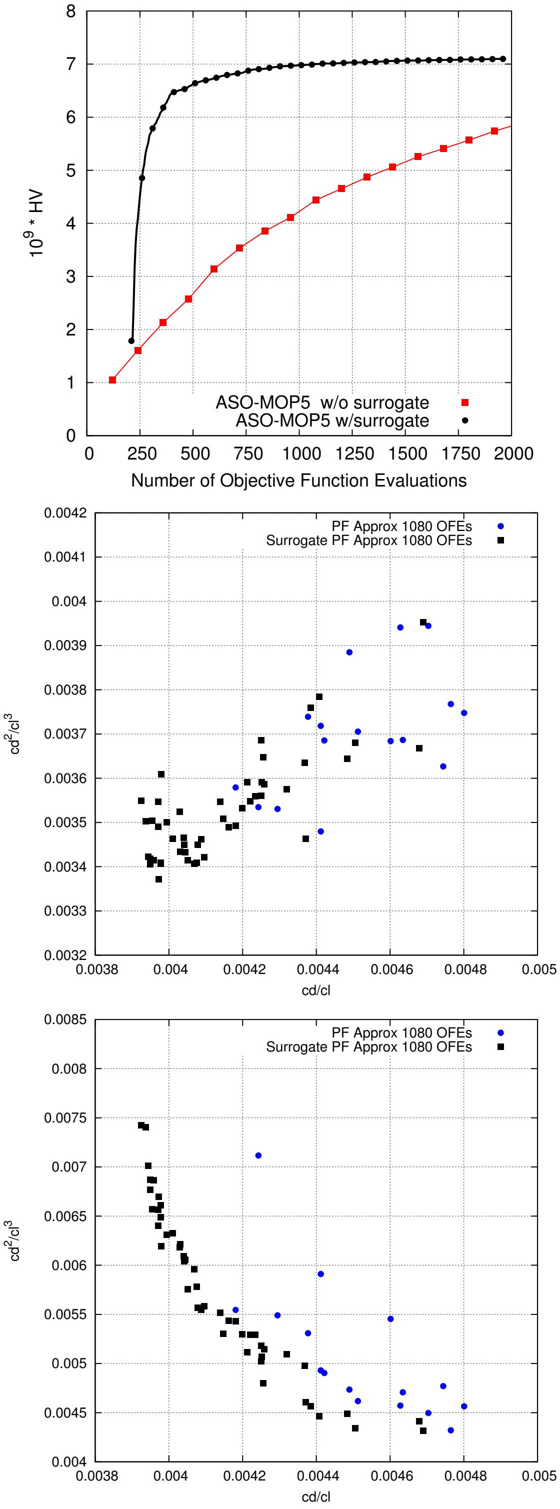


Figure 81: Metamodel assisted Pareto front approximation for ASO-MOP5

8.6 FINAL REMARKS

In this chapter, we have presented a surrogate-based multiobjective evolutionary optimization approach. The main characteristic of our proposed approach is that it uses not only one surrogate model, but a set of them. Unlike other approaches in which only one model is picked up from a set of trained models, or several models are combined by means of a weighting approach, here, each model is searched for Pareto optimal solutions using a MOEA. From the solutions obtained, some of them are selected to solve the MOP in parallel and in a collaborative manner.

Our proposal was tested on five ASO MOPs. Our results indicated that our proposal helps to speed up convergence and that it can produce a substantial reduction in the number of objective function evaluations performed (reaching savings of up to 75% with respect to the same MOEA not using surrogates).

We plan to add more combinations of surrogate models to the ones adopted here. It would also be interesting to couple our surrogate-based approach to other MOEAs and, more interestingly, to combinations of them. This would allow a higher degree of variability during the search which could probably lead to further reductions on the number of objective function evaluations performed.

9 CONCLUSIONS AND FUTURE WORK

9.1 CONCLUSIONS

The main goal of the research work reported in this thesis has been to design a new multi-objective evolutionary algorithm and companion techniques, with the aim of improving efficiency. The proposed approach was designed for being applied in the solution of aerodynamic shape optimization problems commonly found in aeronautical/aerospace engineering system design. In this case, efficiency can be measured in terms of the number of objective functions performed, i.e., our aim was to produce a better Pareto-front approximation than other state-of-the-art MOEAs, for a fixed and reduced number of objective function evaluations. In this context, efficiency can also be measured in terms of CPU time, given a (pre-defined) number of objective function evaluations.

The main conclusions of this research work are the following:

- We have reviewed the application of MOEAs used for solving a diversity of aeronautical/aerospace engineering problems (cf. Chapter 4). From this review we have identified that many of the existing applications rely on the use of state-of-the-art MOEAs such as: *NSGA*, *NCGA*, *MOGA*, *SPEA2*, *NSGA-II*. Most of them are Pareto-based MOEAs which use genetic algorithms as their search engine. More recently, there have been attempts to use other types of MOEAs, based on different search operators such as evolution strategies; and particle swarm optimization. Additionally, the use of indicator-based MOEAs such as the *SMS-EMOA* is also reported in the specialized literature. In the context of aerodynamic shape optimization problems, there are two MOEAs that were specifically designed for dealing with them: *ARMOGA* and $\epsilon\mu$-*ARMOGA*. These two latter algorithms are based on *MOGA*, and make use of range adaptation for the design variables in both cases, and a μ-*GA* and ϵ-dominance in the latter case. We also identified that the use of other metaheuristics with good convergence properties, such as *differential evolution*, has been scarce in this domain. We conclude that the use of MOEAs in aeronautical/aerospace engineering optimization is a mature area and that the use of metaheuristics such as *differential evolution* is a promising research path in this engineering field. This condition motivated us to design an algorithm based on this metaheuristic.

- Based on the previous survey and in the class of aerodynamic shape optimization problems we are interested in, we have defined a set of real-world multi-objective test problems. In this set, seven problems were defined and named *ASO-MOP1* to *ASO-MOP7* (cf. Section 6.3.4, page 131). Four of these problems have two objectives and the other three have three objectives. This set of *ASO-MOPs*, additional to the MOP suites currently available in the literature, was used to test the performance of our proposed approaches.

- We have designed a novel MOEA called *MODE-LD+SS*. This proposed approach adopts the *differential evolution* operators as its search engine, using the standard scheme *DE/rnd/1/bin* . Additionally, the proposed algorithm incorporates two mechanisms for improving both the convergence towards the Pareto front and the uniform distribution of nondominated solutions along the Pareto front. These mechanisms correspond to the concept of local dominance and the use of an environmental selection based on a scalar function.

- We have tested the performance of our proposed *MODE-LD+SS* using benchmark functions taken from the specialized literature, and using two performance measures, *Hv* and *C-Metric*. The performance results of our approach, shown in tables 7 to 16 (pages 145–149), were compared against those obtained by other state of the art MOEAs such as: *NSGA-II*, *SPEA2*, and *MOEA/D*. Our comparative study showed that our proposed *MODE-LD+SS* outperforms *NSGA-II*, *SPEA2* and *MOEA/D*, with respect to the hypervolume performance measure, in five of the nine MOPs used. Our approach was also found to be competitive with respect to *SPEA2* and *MOEA/D* in other three MOPs from this benchmark. Regarding the C-Metric, our proposed MODE-LD+SS outperformed *NSGA-II*, *SPEA2* and *MOEA/D* in the nine MOPs adopted. Based on these results, we can conclude that our proposed approach has good convergence properties. As indicated by the C-Metric, our proposed approach is able to converge closer to the true Pareto-front than other state-of-the-art MOEAs.

- We have tested the performance of our proposed approach *MODE-LD+SS* using the aerodynamic shape optimization problems defined as part of this research work. In this study we have compared its performance against other state-of-the-art MOEAs such as: *MOEA/D*, *MOEA/D-DE*, *SMS-EMOA*, *ARMOGA*, and $\epsilon\mu$*ARMOGA*. The performance measure selected for comparison was the hypervolume attained. This comparative study (cf. table 23 in page 179) showed that our proposed approach *MODE-LD+SS* outperformed the other MOEAs in five of seven ASO-MOPs. In the case of the other ASO-MOPs, their results were competitive to those of *MOEA/D-DE* and *SMS-EMOA*. These results further support the good convergence properties of our approach, since it is able to outperform the *SMS-EMOA*, with respect to the Hv measure, even when the *SMS-EMOA* makes use of this metric in its selection process. Additionally, in this study, the convergence history of all algorithms was analyzed, and we conclude that our proposed approach is able to converge closer to the Pareto-front, than the other MOEAs, if on the order of 1,000 to 2,000 objective function evaluations are allowed. This condition is important since in most of the applications surveyed, this is the number of objective function evaluations commonly performed. With this condition, we also conclude that *differential evolution* is a competitive evolutionary approach for solving aerodynamic shape optimization problems.

- We have designed a parallel approach called *pMODE-LD+SS*. It is based on its serial counterpart *MODE-LD+SS*, inheriting its evolutionary operators and selection mechanisms. The design of this approach was motivated by the aim of improving the efficiency of *MODE-LD+SS*, in terms of the computational time required. Our proposed parallel approach is based on the *Island* paradigm. Its main characteristics comprise: a "random pair-islands" bidirectional migration scheme, use of a pollination scheme, random migration policy, and a replacement based on the environmental selection of *MODE-LD+SS*.

- We have tested the performance of our proposed *pMODE-LD+SS*, using the two benchmarks proposed in this research work (cf. Chapter 6 in page 101). From these studies, and for both benchmarks, we conclude that our proposed parallel approach is able to improve efficiency (with respect to its serial counterpart) in terms of its execution time, i.e., it is able to attain similar Pareto-front approximations, with respect to those attained by its serial version *MODE-LD+SS*, but with a reduced computational time, as indicated by the speed-up attained with the parallel approach (cf. Table 20 in page 177, and Table 31 in page 182).

- We designed and implemented a surrogate-based multi-objective evolutionary optimization approach. This approach makes use of multiple surrogate models which operate in parallel with the aim of combining their features when solving a costly multi-objective optimization problem. Our proposal was tested in five ASO-MOPs from the benchmark defined. The results of this study showed that our proposal helps to speed up convergence and that it can produce a substantial reduction in the number of objective function evaluations performed, reaching savings of up to 75% with respect to the same MOEA not using surrogates. From this results we conclude that our surrogate-based approach has obvious advantages for dealing with expensive objective functions such as those involved in aeronautical optimization problems.

9.2 FUTURE WORK

As part of the future work, derived from the present thesis, we consider the following:

- **Use of alternative DE schemes**: In our proposals, which are based on *differential evolution*, we have used the basic *DE/rnd/1/bin* scheme. However, for this metaheuristic, it is possible to use other alternative schemes. We plan to test and validate different schemes, which might result in an improved convergence rate for our proposed basic approaches. The use of different schemes is more appealing, particularly in the parallel version of our proposal, since having different schemes in each island, will allow to promote diversity in the solutions searched.

- **Parameter tuning for aerodynamic shape optimization problems**: For our proposals, we have tuned the F, CR, and NB parameters to be used in *MODE-LD+SS*; and, the *number of islands*, *migration rate*, and *epoch* to be used for the parallel approach *pMODE-LD+SS*. In both cases, the ZDT and DTLZ MOPs were used for this process, then the tuned values were directly used in our approaches, when solving the aerodynamic shape optimization problems. In this regard we plan to do a tuning process, using the defined ASO-MOPs, aiming to better understand the behavior of our proposed approaches, to identify the sensitivity of solutions with respect to the parametes.

- **Use of different benchmarks**: The main interest in this research work, was the solution of aerodynamic shape optimization problems. This type of problem is evidently, only a particular class in the universe of multi-objective problems found in aeronautical/aerospace engineering. We plan to select and/or define other classes of problems such as structural and aeroelastic multi-objective optimization, to test our proposed approaches and to evaluate their performance in such cases.

- **Use of alternative parallelization techniques**: Due to the high computational cost required by many aeronautical and aerospace engineering optimization problems, the use of parallelism is relatively common. However, more elaborate parallelization techniques based, for example, on co-evolution [232], cellular computing [2], GPU computing [260] and asynchronous techniques [11] are still scarce in this area and more work in that direction is expected in the next few years.

- **Different surrogate approaches**: In the surrogate-based approach proposed in this research work, we have used a combination of metamodels based on radial basis functions. However, we plan to test more combinations of surrogate models. It would also be interesting to couple our surrogate-based approach to other MOEAs and, more interestingly, to combinations of them. This would allow a higher degree of variability during the search which could probably lead to further reductions in the number of objective function evaluations performed.

- **Efficient constraint-handling techniques**: There are many applications in aeronautical/aerospace engineering that are subject to constraints. In most cases, infeasible solutions are discarded and generated again, or a simple external penalty function is adopted. However, many other constraint-handling approaches exist, which could be very useful in multi-objective optimization, since they can explore the boundary between the feasible and the infeasible region in a more efficient way than traditional penalty functions (see for example [159, 213]). It would also be interesting to design approaches that can efficiently deal with problems having many nonlinear constraints.

BIBLIOGRAPHY

[1] Hussein A. Abbass, Ruhul Sarker, and Charles Newton. PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems. In *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 971–978, Piscataway, New Jersey, May 2001. IEEE Service Center.

[2] Enrique Alba and Bernabé Dorronsoro. *Cellular Genetic Algorithms*. Springer, New York, 2008. ISBN 978-0-387-77609-5.

[3] J.J. Alonso, P. LeGresley, and V. Pereyra. Aircraft Design Optimization. *Mathematics and Computer in Simulation*, 79:1948–1958, 2008.

[4] Julio E. Alvarez-Benitez, Richard M. Everson, and Jonathan E. Fieldsend. A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 459–473, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[5] J.D. Anderson and J.F. Wendt. *Computational fluid dynamics*, volume 206. McGraw-Hill, 1995.

[6] John D. Anderson. *Fundamentals of Aerodynamics*. Mc Graw-Hill Book Company, New York, 1985.

[7] Murray B. Anderson. Genetic Algorithms In Aerospace Design: Substantial Progress, Tremendous Potential. Technical report, Sverdrup Technology Inc./TEAS Group, 260 Eglin Air Force Base, FL 32542, USA, 2002.

[8] Mohammad Arabnia and Wahid Ghaly. A strategy for multi-objective shape optimization of turbine stages in three-dimensional flow. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia Canada, 10 –12 September 2008.

[9] Alfredo Arias Montaño, Carlos A. Coello Coello, and Efrén Mezura-Montes. MODE-LD+SS: A Novel Differential Evolution Algorithm Incorporating Local Dominance and Scalar Selection Mechanisms for Multi-Objective Optimization. In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, Barcelona, Spain, July 2010. IEEE Press.

[10] Alfredo Arias-Montaño, Carlos A. Coello Coello, and Efrén Mezura-Montes. Multi-Objective Evolutionary Algorithms in Aeronautical and Aerospace Engineering. *IEEE Transactions on Evolutionary Computation*, page In Press., 2012.

[11] Varvara G. Asouti and Kyriakos C. Giannakoglou. Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes. *Engineering Optimization*, 41(3): 241–257, March 2009.

[12] Brian Beachkofski and Ramana Grandhi. Improved distributed hypercube sampling. In *AIAA Paper 2002-1274, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, Colorado, Apr. 22–25 2002.

[13] A. Ben-Tal. Characterization of pareto and lexicographic optimal solutions. *Multiple criteria decision making theory and application*, 177: 1–11, 1980.

[14] R. Benayoun, J. De Montgolfier, J. Tergny, and O. Laritchev. Linear programming with multiple objective functions: Step method (stem). *Mathematical programming*, 1(1):366–375, 1971.

[15] Ernesto Benini. Three-Dimensional Multi-Objective Design Optimization of a Transonic Compressor Rotor. *Journal of Propulsion and Power*, 20(3):559–565, May–June 2004.

[16] Ernesto Benini and Andrea Toffolo. Development of High-Performance Airfoils for Axial Flow Compressors Using Evolutionary Computation. *Journal of Propulsion and Power*, 18(3):544–554, May–June 2002.

[17] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.

[18] M. Bhattacharya and G. Lu. A dynamic approximate fitness based hybrid ea for optimization problems. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1879–1886, 2003.

[19] Chen Bing, Xu Xu, and Cai Guobiao. Single- and Multi-Objective Optimization of Scramjet Components Using Genetic Algorithms Based on a Parabolized Navier-Stokes solver. In *AIAA Paper 2006-4686, 42nd AIAA AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Sacramento, California, USA, July 9-12 2006.

[20] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, UK, 1995.

[21] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA — A Platform and Programming Language Independent Interface for Search Algorithms. TIK Report 154, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, October 2002.

[22] Stefan Bleuler, Marco Laumanns, Lothar Thiele, and Eckart Zitzler. PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 494–508, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[23] G. Boothroyd. Product design for manufacture and assembly. *Computer-Aided Design*, 26(7):505–520, 1994.

[24] G.E.P. Box and N.R. Draper. *Empirical model-building and response surfaces.* John Wiley & Sons, 1987.

[25] Steven A. Brandt, Randall J. Stiles, John J. Bertin, and Ray Whitford. *Introduction to Aeronautics: A Design Perspective*. AIAA Education Series, 1997.

[26] Jürgen Branke, Thomas Kaussler, and Harmut Schmeck. Guidance in Evolutionary Multi-Objective Optimization. *Advances in Engineering Software*, 32:499–507, 2001.

[27] Matthew Brown, Neil R. Mudford, Andrew J. Neely, and Tapabrata Ray. Robust design optimization for two-dimensional scramjet inlets.

In *AIAA Paper 2006–8140, 14th AIAA/AHI Space Planes and Hypersonic Systems and Technology Conference*, Australia, November 6 – 9 2006.

[28] D. Bueche, N.N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics: Part C*, 35(2): 183–194, May 2005.

[29] Michael A. Buonanno and Dimitri N. Mavris. A new method for aircraft concept selection using multicriteria interactive genetic algorithms. In *AIAA Paper 2005–1020, 43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, 10–13 January 2005.

[30] Stuart E. Burge. *Systems Engineering: Using Systems Thinking to Design Better Aerospace Systems*, chapter . John Wiley & Sons, Ltd, 2010. ISBN 9780470686652. doi: 10.1002/9780470686652.eae536. URL `http://dx.doi.org/10.1002/9780470686652.eae536`.

[31] Zhihua Cai, Wenyin Gong, and Yongqin Huang. A Novel Differential Evolution Algorithm Based on ϵ-Domination and Orthogonal Design Method for Multiobjective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 286–301, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[32] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley, New York, 1961.

[33] Kazuhisa Chiba, Shigeru Obayashi, and Kazuhiro Nakahashi. Design exploration of aerodynamic wing shape for reusable launch vehicle flyback booster. *Journal of Aircraft*, 43(3):832–836, May–June 2006.

[34] Seongim Choi, Juan J. Alonso, and Hyoung S. Chung. Design of a low-boom supersonic business jet using evolutionary algorithms and an adaptive unstructured mesh method. In *AIAA Paper 2004-1758, 45th AIAA/ASME/ASCE/AHS/ASC Structure, Structural Dynamics and Materials Conference*, Palm Springs, CA, USA, April 19-22 2004.

[35] Hyoung-Seog Chung and Juan J. Alonso. Multiobjective optimization using approximation model-based genetic algorithms. In *AIAA Paper 2004-4325, 10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Albany, New York, USA, August 30-September 1 2004.

[36] Hyoung-Seog Chung, Seongim Choi, and Juan J. Alonso. Supersonic business jet design using a knowledge-based genetic algorithm with an adaptive, unstructured grid methodology. In *AIAA Paper 2003-3791, 21st Applied Aerodynamics Conference*, Orlando, Florida, USA, 23-26 June 2003.

[37] P. Cinnella and P. M. Congedo. Optimal airfoil shapes for viscous transonic flows of dense gases. In *AIAA Paper 2006-3881, 36th AIAA Fluid Dynamics Conference and Exhibit*, San Francisco, California, USA, June 5-8 2006.

[38] CA Coello Coello. The emoo repository: a resource for doing research in evolutionary multiobjective optimization. *Computational Intelligence Magazine, IEEE*, 1(1):37–45, 2006.

[39] Carlos 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, January 2002.

[40] Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[41] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[42] J.L. Cohon. Multiobjective programming and planning. 1978. *Academic, New York*, 2010.

[43] P. M. Congedo, C. Corre, and P. Cinnella. Airfoil Shape Optimization for Transonic Flows of Bethe-Zel'dovich-Thompson Fluids. *AIAA Journal*, 45(6):1303–1316, June 2007. DOI: 10.2514/1.21615.

[44] J. Corbett, M. Dooner, J. Meleka, and C. Pym. *Design for manufacture*. Addison Wesley, 1991.

[45] Mario Costa and Edmondo Minisci. MOPED: A Multi-objective Parzen-Based Estimation of Distribution Algorithm for Continuous Problems. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 282–294, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[46] Salvatore D'Angelo and Edmondo A. Minisci. Multi-Objective Evolutionary Optimization of Subsonic Airfoils by Kriging Approximation and Evolutionary Control. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, pages 1262–1267, Vol. 2, Edinburg, Scotland, September 2005.

[47] I. Das. An improved technique for choosing parameters for pareto surface generation using normal-boundary intersection. In *Short Paper Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization*, volume 2, pages 411–413, 1999.

[48] I. Das and JE Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[49] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA)*, pages 825–830. Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA), 2002.

[50] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[51] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[52] Kalyanmoy Deb, Pawan Zope, and Abhishek Jain. Distributed Computing of Pareto-Optimal Solutions with Evolutionary Algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 534–549, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[53] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[54] D. J. Doorly and J. Peiró. Distributed evolutionary computational methods for multiobjective and multidisciplinary optimization. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, MO, Sept. 2–4 1998.

[55] D. J. Doorly, J. Peiró, and S. Spooner. Design optimisation using distributed evolutionary methods. In *37th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, Jan. 11–14 1999.

[56] Mark Drela. Xfoil: An analysis and design system for low reynolds number aerodynamics. In *Conference on Low Reynolds Number Aerodynamics*, University Of Notre Dame, IN, June 1989.

[57] David Eakin. *Aircraft Design for Manufacture and Assembly*, chapter . John Wiley & Sons, Ltd, 2010. ISBN 9780470686652. doi: 10.1002/9780470686652.eae363. URL http://dx.doi.org/10.1002/9780470686652.eae363.

[58] David Eby, R. C. Averill, Boris Gelfand, William F. Punch, Owen Mathews, and Erik D. Goodman. An injection island ga for flywheel design optimization. In *Proc. EUFIT '97, - 5th European Congress on Intelligent Techniques and Soft Computing*, pages 167–190. Morgan Kaufmann, 1997.

[59] A.E. Eiben and J.E. Smith. *Introduction to evolutionary computing*. Springer Verlag, 2003.

[60] M. Emmerich, A. Giotis, M. Özdenir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In *Parallel Problem Solving from Nature*, number 2439 in Lecture Notes in Computer Science, pages 371–380. Springer, 2002.

[61] Michael Emmerich, Nicola Beume, and Boris Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 62–76, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[62] Michael T. M. Emmerich, Kyriakos Giannakoglou, and Boris Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions Evolutionary Computation*, 10(4):421–439, 2006.

[63] Andries P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2005.

[64] Ali Farhang-Mehr and Shapour Azarm. Diversity Assessment of Pareto Optimal Solution Sets: An Entropy Approach. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 723–728, Piscataway, New Jersey, May 2002. IEEE Service Center.

[65] Gerald Farin. *Curves and surfaces for computer-aided geometric design*. Elsevier Science and Technology Books, 1997.

[66] Marc Fischer, David Kennedy, and Carol Featherson. Multilevel optimization of a composite aircraft wing using viconopt mlo. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia, Sep. 4–6 2002.

[67] M. Fleischer. The Measure of Pareto Optima. Applications to Multiobjective Metaheuristics. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 519–533, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[68] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization.

In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[69] Carlos M. Fonseca and Peter J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.

[70] C.M. Fonseca and P.J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*, 3(1):1–16, 1995.

[71] A.I.J. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009.

[72] K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38(1):43 – 76, 2002. ISSN 0376-0421.

[73] A.A. Giunta and L. Watson. A comparison of approximation modeling techniques: Polynomial versus interpolating models. Technical Report 98-4758, AIAA, 1998.

[74] T. Goel, R. Vaidyanathan, R. Haftka, W. Shyy, N. Queipo, and K. Tucker. Response surface approximation of pareto optimal front in multiobjective optimization. Technical Report 2004-4501, AIAA, 2004.

[75] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[76] David E. Goldberg and Jon Richardson. Genetic algorithm with sharing for multimodal function optimization. In John J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, New Jersey, 1987. Lawrence Erlbaum.

[77] Luis F. Gonzalez. *Robust Evolutionary Methods for Multi-objective and Multidisciplinary Design Optimization in Aeronautics*. PhD thesis, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Australia, 2005.

[78] ZH Han, KS Zhang, WP Song, and ZD QIAO. Optimization of active flow control over an airfoil using a surrogate-management framework. *Journal of aircraft*, 47(2), 2010.

[79] M.P. Hansen and A. Jaszkiewicz. *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical Universityof Denmark, 1998.

[80] R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. res*, 76:1905–1915, 1971.

[81] Martina Hasenjäger and Bernhard Sendhoff. Crawling Along the Pareto Front: Tales From the Practice. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 174–181, Edinburgh, Scotland, September 2005. IEEE Service Center.

[82] Alfredo G. Hernández-Díaz, Luis V. Santana-Quintero, Carlos Coello Coello, Rafael Caballero, and Julián Molina. A New Proposal for Multi-Objective Optimization using Differential Evolution and Rough Sets Theory. In Maarten Keijzer et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 675–682, Seattle, Washington, USA, July 2006. ACM Press. ISBN 1-59593-186-4.

[83] Alfredo G. Hernández-Díaz, Luis V. Santana-Quintero, Carlos A. Coello Coello, and Julián Molina. Pareto-adaptive ϵ-dominance. *Evolutionary Computation*, 15(4):493–517, Winter 2007.

[84] Raymond M. Hicks and Preston A. Henne. Wing Design by Numerical Optimization. *AIAA Journal Of Aircraft*, 15(7):407–412, July 1978.

[85] C. Hirsch and C. Hirsch. *Numerical computation of internal and external flows: fundamentals of computational fluid dynamics*, volume 1. Butterworth-Heinemann, 2007.

[86] Terry L. Holst. Genetic algorithms applied to multi-objective aerospace shape optimization. *Journal of Aerospace Computing, Information, and Communication*, 2(4):217–235, April 2005.

[87] Y.-S. Hong, H.Lee, and M.-J. Tahk. Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks. *Engineering Optimization*, 35(1):91–102, 2003.

[88] Jeffrey Horn. Multicriterion Decision Making. In Thomas Bäck, David Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, volume 1, pages F1.9:1 – F1.9:15. IOP Publishing Ltd. and Oxford University Press, 1997.

[89] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.

[90] Jun Hua, Fanmei Kong, Po-Yang Liu, and David Zingg. Optimization of long-endurance airfoils. In *AIAA-2003-3500, 21st AIAA Applied Aerodynamics Conference*, Orlando, FL, June 23-26 2003.

[91] Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.

[92] M. Hüscken, Y. Jin, and B. Sendhoff. Structure optimization of neural networks for aerodynamic optimization. *Soft Computing*, 9(1):21–28, 2005.

[93] Ioannis C. Kampolis and Kyriakos C. Giannakoglou. A multilevel approach to single- and multiobjective aerodynamic optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2963–2975, 2008.

[94] Antony W. Iorio and Xiaodong Li. Solving rotated multi-objective optimization problems using differential evolution. In *AI 2004: Advances in Artificial Intelligence, Proceedings*, pages 861–872. Springer-Verlag, Lecture Notes in Artificial Intelligence Vol. 3339, 2004.

[95] Antony W. Iorio and Xiaodong Li. Incorporating Directional Information within a Differential Evolution Algorithm for Multi-objective Optimization. In Maarten Keijzer et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 691–697, Seattle, Washington, USA, July 2006. ACM Press. ISBN 1-59593-186-4.

[96] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 2424–2431, Hong Kong, June 2008. IEEE Service Center.

[97] A. Jameson. Computational aerodinamics for aircraft design. *Science*, 245:361–371, 28 July 1989.

[98] A. Jameson. Essential elements of computational algorithms for aerodynamic analysis and design. Nasa/cr-97-206268, icase report no. 97-68, ICASE/NASA, Langley Research Center, VA., 1997.

[99] A. Jameson. Re-engineering the design process through computation. *J. Aircraft*, 36(1):36–50, 1999.

[100] A. Jameson. A perspective on computational algorithms for aerodynamic analysis and design. *Progress in Aerospace Sciences*, 37(2):197–243, 2001.

[101] Antony Jameson. Aerodynamic shape optimization: Exploring the limits of design. In *Proceedings of KSAS 2003*, Gyeongju Korea, November 14-15 2003.

[102] Antony Jameson, Kasidit Leoviriyakit, and Sriram Shankaran. Multipoint aero-structural optimization of wings including planform variations. In *AIAA Paper 2007-764, 45th AIAA Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA, 8-11 January 2007.

[103] Mikkel T. Jensen. Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5):503–515, October 2003.

[104] S. Jeong, Y. Minemura, and S. Obayashi. Optimization of combustion chamber for diesel engine using kriging model. *Journal of Fluid Science and Technology*, 1:138–146, 2006.

[105] Shinkyu Jeong, Kazuhisa Chiba, and Shigeru Obayashi. Data mining for aerodynamic design space. In *AIAA Paper 2005–5079, 23rd AIAA Applied Aerodynamic Conference*, Toronto, Ontario Canada, June 6–9 2005.

[106] Shinkyu Jeong, Kunihiro Suzuki, Shigeru Obayashi, and Mitsuru Kurita. Improvement of nonlinear lateral characteristics of lifting-body

type reentry vehicle using optimization algorithm. In *AIAA Paper 2007–2893, AIAA infotech@Aerospace 2007 Conference and Exhibit*, Rohnert Park, California, USA, 7–10 May 2007.

[107] Shinkyu Jeong, Kunihiro Suzuki, Shigeru Obayashi, and Mitsuru Kurita. Optimization of nonlinear lateral characteristic of lifting-body type reentry vehicle. *Journal Of Aerospace Computing, Information, and Communication*, 6(3):239–255, March 2009.

[108] R. Jin, W. Chen, and T.W. Simpson. Comparative studies of metamodeling techniques under miltiple modeling criteria. Technical Report 2000-4801, AIAA, 2000.

[109] Yaochu Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.

[110] Yaochu Jin and Jürgen Branke. Evolutionary optimization in uncertain environments-a survey. *IEEE Trans. Evolutionary Computation*, 9(3): 303–317, 2005.

[111] Yaochu Jin, Tatsuya Okabe, and Bernhard Sendhoff. Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How? In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 1042–1049, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[112] Che Jing and Tang Shuo. Research on integrated optimization design of hypersonic cruise vehicle. *Aerospace Science and Technology*, 2008. DOI:10.1016/j.ast.2008.01.008.

[113] D.R. Jone, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box function. *Journal of Global Optimization*, 13: 455–492, 1998.

[114] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[115] Jiro Kamiura, Tomoyuki Hiroyasu, Mitsunori Miki, and Shinya Watanabe. MOGADES: Multi-objective genetic algorithm with distributed

environment scheme. In *Computational Intelligence and Applications (Proceedings of the Second International Workshop on Intelligent Systems Design and Applications: ISDA'02)*, pages 143–148, 2002.

[116] Marios K. Karakasis and Kyriakos C. Giannakoglou. Metamodel-assisted multi-objective evolutionary optimization. In *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, EUROGEN 2005*, 2005.

[117] AJ Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal*, 44(4):879–891, 2006.

[118] A.J. Keane and P.B. Nair. Problem solving environments in aerospace design. *Advances in Engineering Software*, 32(6):477–487, 2001.

[119] A.J. Keane and P.B. Nair. Computational approaches for aerospace design: the pursuit of excellence. *John Wiley&Sons, Ltd, West Sussex*, 582, 2005.

[120] AJ Keane and JP Scanlan. Design search and optimization in aerospace engineering. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1859):2501–2529, 2007.

[121] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.

[122] Hyoung-Jin Kim and Meng-Sing Liou. New multi-objective genetic algorithms for diversity and convergence enhancement. In *47th AIAA Aerospace Sciences Meeting The new Horizons Forum and Aerospace Exposition*, Orlando, Florida, January 5–8 2009.

[123] S. Kirkpatrick, C.D. Gellatt, and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[124] J. Knowles and D. Corne. On metrics for comparing nondominated sets. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, pages 711–716. IEEE, 2002.

[125] J. Knowles and E. J. Hughes. Multiobjective optimization on a budget of 250 evaluation. In *Proceedings of the Third International Conference of EMO*, pages 176–190, 2005.

[126] Joshua Knowles. ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, February 2006.

[127] Joshua Knowles and David Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.

[128] Joshua Knowles and David Corne. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 757–771, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[129] Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[130] Joshua D. Knowles, David W. Corne, and Mark Fleischer. Bounded Archiving using the Lebesgue Measure. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2490–2497, Canberra, Australia, December 2003. IEEE Press.

[131] K. Krishnakumar. Micro–genetic algorithms for stationary and non-stationary function optimization. In *SPIE proceedings: intelligent control and adaptive systems*, pages 289–296, 1989.

[132] Ilan Kroo. Multidisciplinary Optimization Applications in Preliminary Design – Status and Directions. In *38th, and AIAA/ASME/AHS Adaptive Structures Forum*, Kissimmee, FL, Apr. 7–10 1997.

[133] Ilan Kroo. Innovations in aeronautics. In *42nd AIAA Aerospace Sciences Meeting*, Reno, NV, January 5-8 2004.

[134] T. Kuhn, C. Rösler, and H. Baier. Multidisciplinary design methods for the hybrid universal ground observing airship (HUGO). In *AIAA Paper 2007–7781*, Belfast, Northern Ireland, 18–20 September 2007.

[135] Saku Kukkonen and Jouni Lampinen. An Extension of Generalized Differential Evolution for Multi-objective Optimization with Constraints. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 752–761, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

[136] Saku Kukkonen and Jouni Lampinen. GDE3: The third Evolution Step of Generalized Differential Evolution. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 443–450, Edinburgh, Scotland, September 2005. IEEE Service Center.

[137] Takayasu Kumano, Shinkyu Jeong, Shigeru Obayashi, Yasushi Ito, Keita Hatanaka, and Hiroyuki Morino. Multidisciplinary Design Optimization of Wing Shape for a Small Jet Aircraft Using Kriging model. In *AIAA Paper 2006-932, 44th AIAA Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA, January 9-12 2006.

[138] H.T. Kung, F. Luccio, and F.P. Preparata. On finding the maxima of a set of vectors. *Journal of the Association for Computing Machinery*, 22 (4):469–476, 1975.

[139] Frank Kursawe. A variant of evolution strategies for vector optimization. In Hans-Paul Schwefel and Reinhard MÃ€nner, editors, *Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197. Springer Berlin / Heidelberg, 1991. ISBN 978-3-540-54148-6. URL http://dx.doi.org/10.1007/BFb0029752. 10.1007/BFb0029752.

[140] G. Kuruvila, Shlomo Ta'asan, and M. D. Salas. Airfoil design and optimization by the one-shot method. In *AIAA-95-0478, AIAA 33rd Aerospace Sciences Meeting*, Reno, NV, January 9-12 1995.

[141] J. Lampinen. De's selection rule for multiobjective optimization. Technical report, Lappeenranta University of Technology, 2001.

[142] Ricardo Landa Becerra and Carlos A. Coello Coello. Solving Hard Multiobjective Optimization Problems Using ε-Constraint with Cultured Differential Evolution. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund Burke, Juan J. Merelo-Guervós, L. Darrell Whitley, and Xin Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, pages 543–552. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland, September 2006.

[143] Harald Langer. *Extended Evolutionary Algorithms for Multiobjective and Discrete Design Optimization of Structures*. PhD thesis, Lehrstuhl fur Leichtbau, Technische Universitat Munchen, Germany, October 2005.

[144] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.

[145] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.

[146] D.S. Lee, L.F. Gonzalez, J. Periaux, and K. Srinivas. Robust design optimisation using multi-objective evolutionary algorithms. *Computer & Fluids*, 37:565–583, 2008.

[147] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1994.

[148] Jeffrey C. Lewis and Ramesh K. Agarwal. Airfoil design via control theory using full potential and euler equations. In *AIAA 14th Applied Aerodynamics Conference*, New Orleans, LA, June 1996.

[149] Hui Li and Qingfu Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, April 2009.

[150] Yongsheng Lian and Meng-Sing Liou. Multiobjective Optimization Using Coupled Response Surface Model end Evolutinary Algorithm. In *AIAA Paper 2004–4323, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, USA, 30 August–1 September 2004.

[151] Yongsheng Lian and Meng-Sing Liou. Multi-objective optimization of transonic compressor blade using evolutionary algorithm. *Journal of Propulsion and Power*, 21(6):979–987, November-December 2005.

[152] Yongsheng Lian and Meng-Sing Liou. Multi-objective optimization of a transonic compressor blade using evolutionary algorithm. In *AIAA*

Paper 2005–1816, 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Austin, Texas, USA, 18–21 April 2005.

[153] D. Lim, Y.S. Ong, M.H. Lim, and Y. Jin. Single/multi-objective inverse robust evolutionary design methodology in the presence of uncertainty. *Evolutionary Computation in Dynamic and Uncertain Environments*, pages 437–456, 2007.

[154] F. Luna, A. Nebro, and E. Alba. Parallel evolutionary multiobjective optimization. *Parallel Evolutionary Computations*, pages 33–56, 2006.

[155] Nateri K. Madavan. Multiobjective Optimization Using a Pareto Differential Evolution Approach. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1145–1150, Piscataway, New Jersey, May 2002. IEEE Service Center.

[156] D.W. Marquardt. An algorithm for the least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11(2):431–441, 1963.

[157] Bento S. Mattos and S. Wagner. New numerically designed advanced helicopter-type airfoils. In *AIAA Paper 96-2374-CP, 14th AIAA Applied Aerodynamics Conference*, New Orleans, LA, USA, June 17-20 1996.

[158] B.W. McCormick. *Aerodynamics, aeronautics, and flight mechanics*. Wiley (New York), 1979.

[159] Efrén Mezura-Montes, editor. *Constraint-Handling in Evolutionary Optimization*, volume 198 of *Studies in Computational Intelligence*. Springer-Verlag, 2009.

[160] Efrén Mezura-Montes, Lucía Muñoz-Dávila, and Carlos A. Coello Coello. A preliminary study of fitness inheritance in evolutionary constrained optimization. In Natalio Krasnogor, Giuseppe Nicosia, Mario Pavone, and David Pelta, editors, *Nature Inspired Cooperative Strategies for Optimization*, pages 1–14. Springer, Berlin, 2008. ISBN 978-3-540-78986-4.

[161] Efrén Mezura-Montes, Margarita Reyes-Sierra, and Carlos A. Coello Coello. Multi-Objective Optimization using Differential Evolution: A

Survey of the State-of-the-Art. In Uday K. Chakraborty, editor, *Advances in Differential Evolution*, pages 173–196. Springer, Berlin, 2008. ISBN 978-3-540-68827-3.

[162] B. Mialon, T. Fol, and C. Bonnaud. Aerodynamic optimzation of subsonic flying wing configurations. In *AIAA-2002-2931, 20th AIAA Applied Aerodynamics Conference*, St. Louis Missouri, June 24-26 2002.

[163] Z. Michalewicz and D.B. Fogel. *How to solve it: modern heuristics*. Springer-Verlag New York Inc, 2004.

[164] J.N. Morse. Reducing the size of the nondominated set: Pruning by clustering. *Computers and Operations Research*, 7(1–2):55–66, 1980.

[165] Boris Naujoks, Werner Haase, Jörg Ziegenhirt, and Thomas Bäck. Multi-objective airfoil design using single parent populations. In W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1156–1163, New York, 9-13 July 2002. Morgan Kaufmann Publishers. ISBN 1-55860-878-8.

[166] Boris Naujoks, Lars Willmes, Thomas Bäck, and Werner Haase. Evaluating Multi-criteria Evolutionary Algorithms for Airfoil Optimization. In Juan Julián Merelo Guervós, Panagiotis Adamidis, Hans-Georg Beyer, José-Luis Fernández-Villaca nas, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN VII*, pages 841–850, Granada, Spain, September 2002. Springer-Verlag. Lecture Notes in Computer Science No. 2439.

[167] Boris Naujoks, Nicola Beume, and Michael T. M. Emmerich. Multi-objective optimisation using s-metric selection: Application to three-dimensional solution spaces. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, pages 1282–1289, Vol. 2, Edinburg, Scotland, September 2005.

[168] K.Y. Ng, C.M. Tan, T. Ray, and H.M. Tsai. Single and multiobjective wing planform and airfoil shape optimization using a swarm algorithm. In *AIAA Paper 2003–45, 41st Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA, 6–9 January 2003.

[169] Shigeru Obayashi and Takanori Tsukahara. Comparison of optimization algorithms for aerodynamic shape design. In *AIAA-96-2394-CP, AIAA 14th Applied Aerodynamics Conference*, New Orleans, LA, USA, June 17-20 1996.

[170] Shigeru Obayashi, Yoshihiro Yamaguchi, and Takashi Nakamura. Multiobjective genetic algorithm for multidisciplinary design of transonic wing planform. *Journal of Aircraft*, 34(5):690–693, September–October 1996.

[171] Shigeru Obayashi, Shinichi Takahashi, and Yukihiro Takeguchi. Niching and Elitist models for MOGAs. In *Parallel Problem Solving from Nature - PPSN V, 5th International Conference*, 1998.

[172] Shigeru Obayashi, Daisuke Sasaki, Yukishiro Takeguchi, and Naoki Hirose. Multiobjective evolutionary computation for supersonic wing-shape optimization. *IEEE Trans. Evolutionary Computation*, 4(2):182–187, 2000.

[173] Shigeru Obayashi, Daisuke Sasaki, and Akira Oyama. Finding tradeoffs by using multiobjective optimization algorithms. *Transactions of JASS*, 47(155):51–58, May 2004.

[174] Tatsuya Okabe, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. On Test Functions for Evolutionary Multi-objective Optimization. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 792–802, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

[175] Y. S. Ong, P. B. Nair, A. J. Keane, and K. W. Wong. Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, Studies in Fuzziness and Soft Computing, pages 307–332. Springer, 2004.

[176] Yew-Soon Ong, Prasanth B. Nair, and Kai Yew Lum. Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Trans. Evolutionary Computation*, 10(4):392–404, 2006.

[177] YS Ong and AJ Keane. A domain knowledge based search advisor for design problem solving environments. *Engineering Applications of Artificial Intelligence*, 15(1):105–116, 2002.

[178] Akira Oyama. *Wing Design Using Evolutionary Algorithms*. PhD thesis, Department of Aeronautics and Space Engineering, Tohoku University, Sendai, Japan, March 2000.

[179] Akira Oyama and Kozo Fujii. A study on airfoil design for future mars airplane. In *AIAA Paper 2006-1484, 44th AIAA Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA, January 9-12 2006.

[180] Akira Oyama and Meng-Sing liou. Multiobjective optimization of rocket engine pumps using evolutionary computation. In *AIAA Paper 2001–2581*, Anaheim, CA, June 11–14 2001.

[181] Akira Oyama and Meng-Sing Liou. Multiobjective optimization of a multi-stage compressor using evolutionary algorithm. In *AIAA Paper 2002–3535, 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Indianapolis, Indiana, USA, July 7–10 2002.

[182] Akira Oyama and Meng-Sing Liou. Multiobjective Optimization of Rocket Engine Pumps Using Evolutionary Algorithm. *Journal of Propulsion and Power*, 18(3):528–535, May–June 2002.

[183] Akira Oyama, Yoshiyuki Okabe, Koji Shimoyama, and Kozo Fujii. Aerodynamic Multiobjective Design Exploration of a Flapping Airfoil Using a Navier-Stokes solver. *Journal Of Aerospace Computing, Information, and Communication*, 6(3):256–270, March 2009.

[184] Ingo Paenke, Jürgen Branke, and Yaochu Jin. Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Trans. Evolutionary Computation*, 10(4):405–420, 2006.

[185] Antonio Pagano, Luigi Federico, Mattia Barbarino, Fabio Guida, and Marco Aversano. Multi-objective aeroacoustic optimization of an aircraft propeller. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia Canada, 10 –12 September 2008.

[186] Vilfredo Pareto. *Cours D'Economie Politique*. F. Rouge, 1896.

[187] S. Pierret. Turbomachinery blade design using a Navier-Stokes solver and artificial neural network. *ASME Journal of Turbomachinery*, 121 (3):326–332, 1999.

[188] C. Poloni, A. Giurgevich, L. Onesti, and V. Pediroda. Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 186(2):403–420, 2000.

[189] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Springer, Berlin, 2005. ISBN 3-540-20950-6.

[190] Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Springer, Berlin, 2005. ISBN 3-540-20950-6.

[191] T.H. Pulliam, N. Nemec, T. Holst, and D.W. Zingg. Comparison of evolutionary (genetic) algorithm and adjoint methods for multi-objective viscous airfoil optimization. In *AIAA Paper 2003-0298, 41st Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, 2003.

[192] Man Mohan Rai. Toward a hybrid aerodynamic design procedure based on neural networks and evolutionary methods. In *AIAA Paper 2002-3143, 20th AIAA Applied Aerodynamics Conference*, St. Louis, Missouri, 24–26 June 2002.

[193] Man Mohan Rai. Robust optimal design with differential evolution. In *AIAA Paper 2004-4588, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, USA, 30 August – 1 September 2004.

[194] S. Rajogapal, Ranjan Ganguli, ACR Pillai, and A. Lurdharaj. Conceptual Design of Medium Altitude Long Endurance UAV using Multi Objective Genetic Algorithm. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honololu, Hawaii, USA, 23–26 April 2007.

[195] K. Rasheed, X. Ni, and S. Vattam. Comparison of methods for developing dynamic reduced models for design optimization. *Soft Computing*, 9(1):29–37, January 2005.

[196] A. Ratle. Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In A. Eiben, Th. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature*, volume V, pages 87–96, 1998.

[197] T. Ray and H. M. Tsai. Swarm algorithm for single- and multiobjective airfoil design optimization. *AIAA Journal*, 42(2):366–373, February 2004.

[198] T. Ray and H.M. Tsai. A parallel hybrid optimization algorithm for robust airfoil design. In *AIAA Paper 2004–905, 42nd AIAA Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA, 5 –8 January 2004.

[199] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA Education Series, 1992.

[200] Margarita Reyes Sierra and Carlos A. Coello Coello. A Study of Fitness Inheritance and Approximation Techniques for Multi-Objective Particle Swarm Optimization. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 65–72, Edinburgh, Scotland, September 2005. IEEE Service Center.

[201] Margarita Reyes-Sierra and Carlos A. Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

[202] Tea Robič and Bogdan Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 520–533, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[203] J. Roskam and C.T.E. Lan. *Airplane aerodynamics and performance*. DARcorporation, 1997.

[204] B.D. Roth and I.M. Kroo. Enhanced collaborative optimization: a decomposition-based method for multidisciplinary design. *Proceedings of the ASME design engineering technical conferences, Brooklyn, New York*, pages 3–6, 2008.

[205] R. Roy, S. Hinduja, and R. Teti. Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals-Manufacturing Technology*, 57(2):697–715, 2008.

[206] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.

[207] Luis Vicente Santana-Quintero and Carlos A. Coello Coello. An Algorithm Based on Differential Evolution for Multiobjective Problems. In Cihan H. Dagli, Anna L. Buczak, David L. Enke, Mark J. Embrechts, and Okan Ersoy, editors, *Smart Engineering System Design: Neural Networks, Evolutionary Programming and Artificial Life*, volume 15, pages 211–220, St. Louis, Missouri, USA, November 2005. ASME Press.

[208] Daisuke Sasaki and Shigeru Obayashi. Efficient search for tradeoffs by adaptive range multi-objective genetic algorithm. *Journal Of Aerospace Computing, Information, and Communication*, 2(1):44–64, January 2005.

[209] Daisuke Sasaki, Shigeru Obayashi, Keisuke Sawada, and Ryutaro Himen. Multiobjective Aerodynamic Optimization of Supersonic Wings Using Navier-Stokes Equations. In *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2000*, Barcelona, Spain, 11–14 September 2000.

[210] Daisuke Sasaki, Guowei Yang, and Shigeru Obayashi. Automated Aerodynamic Optimization System for SST Wing-Body Configuration. In *AIAA-2002-5549, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, September 4-6 2002.

[211] Daisuke Sasaki, Andy J. Keane, and Shahrokh Shahpar. Multiobjective evolutionary optimization of a compressor stage using a grid-enabled environment. In *AIAA Paper 2006-340, 44th AIAA Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA, January 9-12 2006.

[212] Daisuse Sasaki and Shigeru Obayashi. Low-Boom Design Optimization for SST Canard-Wing-Fuselage. In *AIAA Paper 2003–3432, 16th AIAA Computational Fluid Dynamics Conference*, Orlando, Florida, USA, June 23–26 2003.

[213] Dhish Kumar Saxena, Tapabrata Ray, Kalyanmoy Deb, and Ashutosh Tiwari. Constrained Many-Objective Optimization: A Way Forward. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 545–552, Trondheim, Norway, May 2009. IEEE Press.

[214] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, 1984.

[215] J. David Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

[216] Robert J. Schalkoff. *Artificial Neural Networks*. McGraw-Hill, 1997.

[217] K. Abboud M. Schoenauer. Surrogate deterministic mutation. In *Artificial Evolution'01*, pages 103–115. Springer, 2002.

[218] Oliver Schütze, Carlos A. Coello Coello, Sanaz Mostaghim, El-Ghazali Talbi, and Michael Dellnitz. Hybridizing Evolutionary Strategies with Continuation Methods for Solving Multi-Objective Problems. *Engineering Optimization*, 40(5):383–402, May 2008.

[219] Marc Secanell and Afzal Suleman. Numerical Evaluation of Optimization Algorithms for Low-Reynolds Number Aerodynamic Shape Optimization. *AIAA Journal*, 10:2262–2267, October 2005.

[220] Richard S. Shevell. *Fundamentals of Flight*. Prentice-Hall, Inc, 1983.

[221] Koji Shimoyama. *Robust Aerodynamic Design of Mars Exploratory Airplane Wing with a New Optimization Method*. PhD thesis, School of Engineering, The University of Tokyo, Japan, February 2006.

[222] Koji Shimoyama, Akira Oyama, and Kozo Fujii. Development of Multi-Objective Six-Sigma Approach for Robust Design Optimization. *Journal Of Aerospace Computing, Information, and Communication*, 5(8): 215–233, August 2008.

[223] Helmut Sobieczky. Parametric airfoils and wings. In K. Fuji and G. S. Dulikravich, editors, *Notes on Numerical Fluid Mechanics, Vol.. 68*, pages 71–88, Wiesbaden, 1998. Vieweg Verlag.

[224] Wenbin Song and Andy J. Keane. Surrogate-based aerodynamic shape optimization of a civil aircraft engine nacelle. *AIAA Journal*, 45(10): 265–2574, October 2007. DOI: 10.2514/1.30015.

[225] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[226] Ralph E. Steuer and Eng-Ung Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(1):326–344, 1983.

[227] Rainer Storn and Kenneth Price. Differential Evolution - A Fast and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[228] Felix Streichert, Holger Ulmer, and Andreas Zell. Parallelization of Multi-objective Evolutionary Algorithms Using Clustering Algorithms. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 92–107, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[229] Andras Szöllös, Miroslav Smid, and Jaroslav Hajek. Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and epsilon-dominance. *Advances in engineering software*, 40(6):419–430, June 2009.

[230] Tetsuyuki Takahama and Setsuko Sakai. Reducing Function Evaluations in Differential Evolution using Rough Approximation-Based Comparison. In *2008 IEEE Congress on Evolutionary Computation (CEC2008)*, pages 2307–2314. IEEE, 2008.

[231] El-Ghazali Talbi, Sanaz Mostaghim, Tatsuya Okabe, Hisao Ishibuchi, Gunter Rudolph, and Carlos A. Coello Coello. Parallel approaches for multi-objective optimization. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors, *Multiobjective Optimization*, volume 5252 of *Lecture Notes in Computer Science*, pages 349–372. Springer, 2008. ISBN 978-3-540-88907-6.

[232] K.C. Tan, Y.J. Yang, and C.K. Goh. A Distributed Cooperative Coevolutionary Algorithm for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 10(5):527–549, October 2006.

[233] Naoki Tani, Akira Oyama, Koichi Okita, and Nobuhira Yamanishi. Feasibility study of multi objective shape optimization for rocket engine turbopump blade design. In *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Hartford, CT, 21 – 23 July 2008.

[234] Robert Taylor, Terrence Weisshaar, and Vladimir Sarukhanov. Structural design process improvement using evolutionary finite element models. *Journal of Aircraft*, 43(1):172–181, January–February 2006.

[235] John E. Theisinger and Robert D. Braun. Multiobjective hypersonic entry aeroshell shape optimization. *Journal Of Spacecraft and Rockets*, 46(5):957 – 966, September – October 2009.

[236] Akira Todoroki and Masahito Sekishiro. Dimensions and laminates optimization of hat-stiffened composite panel with buckling load constraint using multi-objective ga. In *AIAA Paper 2007–2880, AIAA infotech@Aerospace 2007 Conference and Exhibit*, Rohnert Park, California, USA, 7–10 May 2007.

[237] Akira Todoroki and Masahito Sekishiro. Modified efficient global optimization for a hat-stiffened composite panel with buckling constraint. *AIAA Journal*, 46(9):2257–2264, September 2008.

[238] E. Torenbeek. *Synthesis of subsonic airplane design: an introduction to the preliminary design, of subsonic general aviation and transport aircraft, with emphasis on layout, aerodynamic design, propulsion, and performance*. Springer, 1982.

[239] Gregorio Toscano Pulido. *On the Use of Self-Adaptation and Elitism for Multiobjective Particle Swarm Optimization*. PhD thesis, Computer Science Section, Department of Electrical Engineering, CINVESTAV-IPN, Mexico, September 2005.

[240] Shigeyoshi Tsutsui and Ashish Ghosh. Genetic algorithms with a robust solution searching scheme. *IEEE Trans. Evolutionary Computation*, 1 (3):201–208, 1997.

[241] Tushar Goel, Rajkumar Vaidyanathan, Raphael T. Haftka, Wei Shyy, Nestor V. Queipo, and Kevin Tucker. Response Surface Approximtion of Pareto Optimal Front in Multi-Objective Optimization. *Computer Methods in Applied Mechanics and Engineering*, 196(4):879–893, 2007.

[242] Tea Tušar and Bogdan Filipič. Differential Evolution Versus Genetic Algorithms in Multiobjective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 257–271, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[243] H. Ulmer, F. Streicher, and A. Zell. Model-assisted steady-state evolution strategies. In *Proceedings of Genetic and Evolutionary Computation Conference*, LNCS 2723, pages 610–621, 2003.

[244] H. Ulmer, F. Streichert, and A. Zell. Evolution startegies assisted by gaussian processes with improved pre-selection criterion. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 692–699, 2003.

[245] Somasundaram Valliyappan and Timothy W. Simpson. Exploring visualization strategies to support product family design optimization. In *AIAA Paper 2006–6949, 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, USA, 6–8 September 2006.

[246] David A. Van Veldhuizen, Jesse B. Zydallis, and Gary B. Lamont. Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):144–173, April 2003.

[247] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[248] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.

[249] P. Venkataraman. Optimum airfoil design in viscous flows. In *AIAA 13th Applied Aerodynamics Conference*, San Diego, CA, 1995.

[250] F.A.C. Viana, R.T. Haftka, V. Steffen Jr, S. Butkewitsch, and M.F. Leal. Ensemble of surrogates: A framework based on minimization of the mean integrated square error. *AIAA Paper*, 1885:2008, 2008.

[251] F.A.C. Viana, V. Picheny, and R.T. Haftka. Using cross validation to design conservative surrogates. *AIAA journal*, 48(10):2286–2298, 2010.

[252] Rémy Viennet, Christian Fontiex, and Ivan Marc. Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set. *International Journal of Systems Science*, 27(2):255–260, 1996.

[253] Ivan Voutchkov, Andy J. Keane, and Rob Fox. Robust structural design of a simplified jet engine model, using multiobjective optimization.

In *AIAA Paper 2006–7003*, Portsmouth, Virginia, USA, 6–8 September 2006.

[254] David R. Wallace, Mark J. Jakiela, and W.C. Flowers. Design Search Under Probabilistic Specifications Using Genetic Algorithms. *Computer-Aided Design*, 28(5):405–421, 1996.

[255] Shinya Watanabe, Tomoyuki Hiroyasu, and Mitsunori Miki. Neighborhood Cultivation Genetic Algorithm for Multi-Objective Optimization Problems. In Lipo Wang, Kay Chen Tan, Takeshi Furuhashi, Jong-Hwan Kim, and Xin Yao, editors, *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, volume 1, pages 198–202, Orchid Country Club, Singapore, November 2002. Nanyang Technical University.

[256] U. Wickramasinghe Rajapaksa, X. Li, and R. Carrese. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In *IEEE Congress on Evolutionary Computation 2010 (CEC 2010)*, pages 1857–1864. IEEE, 2010.

[257] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. ISSN 00994987.

[258] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo., editors, *Advances in Neural Information Processing Systems 8*. MIT Press, 1996.

[259] K.S. Won and T. Ray. Performance of kriging and cokriging based surrogate models within the unified framework for surrogate assisted optimization. In *Congress on Evolutionary Computation*, pages 1577–1585. IEEE, 2004.

[260] Man-Leung Wong, Tien-Tsin Wong, and Ka-Ling Fok. Parallel evolutionary algorithms on graphics processing unit. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 3, pages 2286–2293, Edinburgh, Scotland, September 2005. IEEE Service Center.

[261] Chen Xiaoqing, Hou Zhongxi, He Lietang, and Liu Jianxia. Multi-object optimization of waverider generated from conical flow and osculating cone. In *AIAA Paper 2008–131, 46th AIAA Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA, 7 – 10 January 2008.

[262] Feng Xue, Arthur C. Sanderson, and Robert J. Graves. Pareto-based Multi-Objective Differential Evolution. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 862–869, Canberra, Australia, December 2003. IEEE Press.

[263] Yoshihiro Yamaguchi and Toshiyuki Arima. Multi-objective optimization for the transonic compressor stator blade. In *AIAA Paper 2000–4909, 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, USA, September 6 – 8 2000.

[264] Daniela Zaharie and Dana Petcu. Adaptive pareto differential evolution and its parallelization. In Roman Wyrzykowski, Jack Dongarra, Marcin Paprzycki, and Jerzy Wasniewski, editors, *PPAM*, volume 3019 of *Lecture Notes in Computer Science*, pages 261–268. Springer, 2003. ISBN 3-540-21946-3.

[265] Sanyou Y. Zeng, Lishan S. Kang, and Lixin X. Ding. An Orthogonal Multi-objective Evolutionary Algorithm for Multi-objective Optimization Problems with Constraints. *Evolutionary Computation*, 12(1):77–98, Spring 2004.

[266] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[267] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[268] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 2011.

[269] Z. Zhou, Y.S. Ong, M.H. Lim, and B.S. Lee. Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 11(10):957–971, 2007.

[270] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.

[271] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. Technical Report 139, Computer Engineering and Networks Laboratory, ETH Zurich, June 2002.

[272] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

[273] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[274] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.

[275] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.