

Cranfield University

Paolo Fantini

**Effective Multiobjective MDO for Conceptual Design
An Aircraft Design Perspective**

School Of Engineering

PhD Thesis

Cranfield University

School Of Engineering

PhD Thesis

2007

Paolo Fantini

**Effective Multiobjective MDO for Conceptual Design
An Aircraft Design Perspective**

Supervisor: Professor Marin D. Guenov

Academic Years 2004 to 2007

Abstract

Once the requirements for a new aircraft have been defined, the Conceptual design phase is launched. During this phase one or more designers have the goal of defining and investigating a number of alternative solutions. Through discussion with industry it has become apparent that optimisation tools are seldom used, even though these could greatly enhance the work of the designers.

The objective of the work carried forward has been of identifying, comparing and where necessary improving the most suitable techniques, as well as schemes for their integration, in order to perform effectively Multidisciplinary Design and Optimisation (MDO) in the Conceptual phase of the aircraft design. The techniques that have been investigated include: multi-objective optimisation algorithms, MDO algorithms for treating non-hierarchically decomposable systems and Automatic Differentiation (AD).

As a result an algorithm for performing multiobjective MDO has been developed. Given a complete model for a complex non-hierarchically decomposable system and given a number of objectives and constraints, the algorithm is capable of determining a set of well distributed solutions, representative of both local and global Pareto frontiers.

A number of test cases have been used for evaluating the alternative methodologies and the proposed algorithm. These include a set of complex algebraic test cases typically used for evaluating global optimisation algorithms and a simplified aircraft conceptual design model, which was provided by industry.

The results demonstrate the unique capability of the algorithm of determining well distributed solutions on the global and local Pareto frontiers for global multiobjective continuous nonlinear constrained optimisation problems. The results also show this capability when the algorithm is applied to non-hierarchically decomposable systems, as typically encountered when performing MDO. Further work could extend the approach in order to handle mixed discrete/continuous variables.

Acknowledgments

My greatest thanks go to the most important person in my life Arianna, who followed me in this adventure. It was her who gave me strength and determination.

Special thanks go to my family; nothing would have been possible without their support, and Arianna's family for the encouragement and all the lovely food they sent us!

My gratitude goes to Professor Marin D. Guenov, for giving me this opportunity and for his help and advice, and my new, present and past colleagues (in alphabetical order so that no one will be upset!) Dunbing, Helen, Jeremy, Libish, Marco, Mattia, Michele (detto Sergio) and Yves. Special thanks go also to Professor Sergei Utyuzhnikov for his precious help.

My gratefulness goes also to all the members of the VIVACE Project, with whom it was a privilege to work.

I also wish to thank all my friends in Milan who, even if far away, are always in my heart and all my new friends here in the UK.

Finally, as suggested by Yves, enormous gratitude goes to my 1990 VW Polo which, at the brink of 300000 km (7.5 times the circumference of the Earth), is still driving us (more or less) safely to the University every day.

Table of Contents

ACRONYM LIST.....	VI
1 INTRODUCTION	1
1.1 CONCEPTUAL DESIGN IN INDUSTRY	1
1.2 MULTIOBJECTIVE-MULTIDISCIPLINARY DESIGN AND OPTIMISATION AT THE CONCEPTUAL DESIGN PHASE, PROBLEM OUTLINE	3
1.3 HIERARCHICALLY AND NON-HIERARCHICALLY DECOMPOSABLE SYSTEMS	3
1.4 MULTIOBJECTIVE OPTIMISATION AND THE PARETO FRONTIER	5
1.5 RESEARCH AIMS AND OBJECTIVES.....	7
2 LITERATURE REVIEW	9
2.1 DEFINITIONS AND PROBLEM FORMULATION	10
2.2 OPTIMISATION OF NON-HIERARCHICALLY DECOMPOSABLE SYSTEMS.....	12
2.2.1 <i>Nested Analysis and Design (NAND)</i>	15
2.2.2 <i>Simultaneous Analysis and Design (SAND)</i>	15
2.2.3 <i>Collaborative Optimisation (CO)</i>	15
2.2.4 <i>Concurrent Subspace Optimisation (CSSO)</i>	15
2.2.5 <i>Bi-Level Integrated System Synthesis (BLISS)</i>	16
2.2.6 <i>Considerations on the reviewed MDO algorithms</i>	16
2.3 MULTIOBJECTIVE OPTIMISATION	18
2.3.1 <i>Multiobjective optimisation concepts</i>	18
2.3.2 <i>Multiobjective optimisation strategies and methods</i>	19
2.3.3 <i>Methods for generating a well-distributed set of Pareto points</i>	23
2.4 AUTOMATIC DIFFERENTIATION.....	35
2.5 SUMMARY	38
3 FROM AN ANALYSIS OF THE MULTIOBJECTIVE REFORMULATION METHODS TO THE DEVELOPMENT OF THE NC+ METHOD	39
3.1 THE PERIPHERAL REGION PROBLEM.....	39
3.2 OPTIMISATION PROBLEM REFORMULATION	46
3.2.1 <i>Evaluation of the NBI, PP-based and NC method</i>	46
3.2.2 <i>Some considerations with regard to the NBI, PP-based and NC method</i>	53
3.3 THE GLOBAL OPTIMISATION PROBLEM	55
3.4 REQUIREMENTS FOR AN EFFECTIVE MULTIOBJECTIVE OPTIMISATION METHOD.....	63
3.5 THE PATH LEADING TO A NEW MULTIOBJECTIVE OPTIMISATION METHOD.....	63
3.5.1 <i>The modified PP based method</i>	64
3.5.2 <i>The Double Hyper-Cone Boundary Intersection Method</i>	73

3.5.3	<i>The NC+ method</i>	80
3.5.4	<i>Obtaining a complete representation of the Pareto Front</i>	86
3.5.5	<i>Gradient Based Global Multiobjective Optimisation</i>	87
3.6	SUMMARY	89
4	MULTIOBJECTIVE MDO AND APPLICATION OF AUTOMATIC DIFFERENTIATION	91
4.1	MULTIOBJECTIVE MDO FOR CONCEPTUAL DESIGN	91
4.2	AUTOMATIC DIFFERENTIATION AND GLOBAL SENSITIVITY EQUATIONS	95
4.3	SUMMARY	99
5	THE CODE-M ALGORITHM	100
5.1	THE ALGORITHM	100
5.2	IMPLEMENTATION	107
5.3	SUMMARY	108
6	TESTS, RESULTS, ANALYSIS AND DISCUSSION	110
6.1	THE NC+ METHOD	110
6.1.1	<i>NC+ computational cost</i>	110
6.1.2	<i>Independence of the NC+ method from the positioning of the anchor points</i>	111
6.1.3	<i>Complete representation of the Pareto front</i>	112
6.2	NC+ GLOBAL OPTIMISATION	113
6.2.1	<i>Problem Deb1</i>	114
6.2.2	<i>Problem Deb2</i>	117
6.2.3	<i>Problem Kursawe</i>	119
6.2.4	<i>Problem Fonseca</i>	121
6.2.5	<i>Problem Viennet</i>	124
6.3	MULTIOBJECTIVE MULTIDISCIPLINARY OPTIMISATION	127
6.4	AUTOMATIC DIFFERENTIATION	130
6.5	USMAC	132
6.6	SUMMARY	144
7	SUMMARY, CONCLUSIONS AND FUTURE WORK	146
8	REFERENCES	149
9	PUBLICATIONS OF THE AUTHOR	154
9.1	CONFERENCE PAPERS	154
9.2	JOURNAL PUBLICATIONS	154

10	BIBLIOGRAPHY	155
10.1	APPROXIMATIONS	155
10.2	SENSITIVITY ANALYSIS	156
10.3	MULTIDISCIPLINARY DESIGN AND OPTIMISATION (MDO).....	156
10.4	UNCERTAINTY	158
10.5	DESIGN OF EXPERIMENTS (DOE)	159
10.6	OTHER	159
11	APPENDIX.....	160
11.1	TERMS AND LEGEND	160
11.2	ARCHITECTURE DESCRIPTION	161
11.3	COMPONENTS DETAILS	166
11.3.1	<i>Component GUI</i>	167
11.3.2	<i>Component Computation</i>	168
11.3.3	<i>Component Optimiser</i>	169

Acronym list

AD	Automatic Differentiation
AOF	Aggregate Objective Function
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BLISS	Bi-Level Integrated System Synthesis
CA	Contributing Analysis
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CHIM	Convex Hull of Individual Minima
CO	Collaborative Optimisation
COP	Coordination Optimisation Procedure
CSSO	Concurrent Subspace Optimisation
DHCBI	Double Hyper-Cone Boundary Intersection
DM	Decision Maker
EA	Evolutionary Algorithms
EMO	Evolutionary Multiobjective Optimisation
FAR	Federal Aviation Regulations
GFCL	Generate First – Choose Later
GSE	Global Sensitivity Equations
JAR	Joint Aviation Regulations
MDO	Multidisciplinary Design and Optimisation
NAND	Nested Analysis and Design
NBI	Normal-Boundary Intersection
NC	Normal Constraint
PP	Physical Programming
SAND	Simultaneous Analysis and Design
SLO	System Level Optimiser
SQP	Sequential Quadratic Programming
USMAC	Ultra Simplified Model of Aircraft
VIVACE	Value Improvement Through a Virtual Aeronautical Collaborative Enterprise
WS	Weighted Sum

1 Introduction

Historically the development of a new aircraft follows three subsequent design phases: the conceptual, the preliminary and the detailed design phase. With each successive phase an increased level of detail is considered. At the conceptual design phase the main configuration of the aircraft is defined. The preliminary design phase allows a first subdivision of the aircraft in its main components, for each of which a preliminary design is conducted. At the detailed design phase a final decomposition of all the main components is carried out. At the end of this phase the aircraft will be ready for manufacturing. Unfortunately the design process is usually not such a straightforward process. Because of the inconsistencies between the different levels of detail of the various design phases, several iterations are frequently required before the final design is obtained.

The current research focuses on the first phase, the conceptual design phase. During this phase the designers have the possibility of investigating alternative solutions, some of which can greatly differ one from another. The basic layout of the aircraft is still to be defined, thus the basic geometric characteristics as wing span, wing area and many other are still to be determined. Since none of this information is available it is obvious that no high granularity tools such as CAD or CAE software tools can be of use. Such tools, in fact, require a detailed description of the geometrical characteristics of the aircraft to be known and can only be used in later phases of the design. Therefore, for investigating possible solutions, the designers have to rely on simple physical and empirical equations which allow an approximate evaluation of the behaviour of the aircraft being studied. For the development of a complex system such as an aircraft the number of equations considered is in the order of hundreds.

1.1 Conceptual design in industry

Thanks to some work performed in partnership with a major airframe manufacturer, in the context of the European project VIVACE, it was possible to have an insight into how the Conceptual design process is performed within industry.

The goal of the designers is that of developing an aircraft capable of satisfying a set of predefined requirements. These will comprise the details of the mission for which the aircraft is to be designed, as well as all those regulations such as the Federal Aviation Regulations (FAR) and Joint Aviation Regulations (JAR) to which all aircraft have to comply.

All the equations modelling the various characteristics of an aircraft are coded in some programming language. According to the mission and some characteristics of the aircraft to be investigated the appropriate models are selected. These are then assembled by the designer with the aid of appropriate software tools. As a result a complete model for simulating the behaviour of the aircraft is generated. The model obtained will be controlled by means of its input variables.

According to the particular configuration that the designer wishes to analyse, the values of the input variables are set and the complete model is executed. By analysing the results obtained the designer can decide if and how to modify these, eventually reexecuting the model. This iterative process is repeated until a satisfactory solution is obtained. Following this approach, it is clear that the experience of the designer becomes crucial for obtaining a satisfactory solution.

For developing a competitive aircraft the aim of the designers will not only be that of developing one capable of satisfying the requirements, but also that of developing an optimal aircraft with respect to some chosen criteria. The main problem is that at this initial phase of the design the models used are not sufficiently exhaustive for providing all the information required for selecting a particular configuration. Then, additional considerations based on the experience of the designers as well as further, higher level, investigations of specific solutions might be required. This is particularly true when the configurations investigated differ significantly from conventional ones, for which a great amount of empirical data is available.

Through discussion with industry it has become apparent that optimisation tools are seldom used, even though they could greatly enhance the work of the designers. The job of the designer, in fact, corresponds to solving an optimisation problem, where the requirements coincide with the constraints and the objectives coincide with minimising or maximising a number of criteria. The difficulties reported in using

these numerical tools are mainly related with their limitations as instruments for investigating the design space.

Ideally the designer would require a tool capable of allowing him/her to gain an insight into the possible alternative solutions and their characteristics, rather than one generating a unique solution automatically.

1.2 Multiobjective-Multidisciplinary Design and Optimisation at the conceptual design phase, problem outline

Multidisciplinary design and optimisation (MDO) is the process of optimising a complex system, taking into account concurrently all the contributing disciplines through which it is possible to model its global behaviour. Such disciplines will be referred to as Contributing Analysis (CA).

In most of the problems to which MDO is applied, the considered complex systems contain internally coupled disciplines. In such a case the complex system is said to be non-hierarchically decomposable. MDO can be viewed as the superposition of two problems: the handling of the CAs, in order to obtain consistent data and the optimisation of the system as a whole.

Furthermore, an additional problem must be considered. The aim of the designer is that of obtaining an optimal solution with respect to a number of criteria. When multiple conflicting criteria are present no single optimal solution exists, but rather infinity of alternative solutions satisfying different tradeoffs. It is important then to identify representative solutions that will allow the designer to gain an insight into all the possible alternatives, this to allow him/her to perform an informed final decision. In order to do this, multiobjective optimisation methods can be a valuable tool.

1.3 Hierarchically and Non-Hierarchically decomposable systems

When performing conceptual design, the first task that the designer will have to perform, is selecting the CAs to be used for simulating the behaviour of the system. Each CA will simulate a particular engineering discipline, a physical part of the system or a non-engineering discipline. Some examples of these, for aircraft, are

respectively: the aerodynamic behaviour, the seating distribution of the passengers in the fuselage and the life cycle economic model. From a computational point of view each of the CAs corresponds to a program or a piece of code, each of which has a set of input and a set of output variables. The ensemble of all such variables coincides with the variables of the complex system.

In a complex system, such as an aircraft, the various CAs mutually interact. The interaction results in an exchange of information among the CAs. Hence, for analysing a particular system the CAs have to be organised and executed in a sequence satisfying such interactions.

The way in which the CAs are organised is not only dependent on the input and output variables of each CA, but also according to the set of variables that the designer wishes to use as input to the whole complex system. Such variables will obviously have to be selected by the designer among the ensemble of the variables belonging to the CAs.

When organizing the CAs two cases can occur: the CAs can be ordered hierarchically or the CAs can not be ordered hierarchically (AIAA Technical Committee on Multidisciplinary Design Optimization, 1991). While in the first case it is possible to execute the CAs in a sequence, in the second the CAs are not all sequentially connected and internal coupling between the CAs are present. In these two cases the complex system is said to be respectively: hierarchically decomposable and non-hierarchically decomposable.

In aircraft conceptual design the number of CAs required for modelling the complete system is in the order of the hundreds. Furthermore in a complex system such as an aircraft many interactions exist among the CAs, this frequently resulting in internal couplings among some of the CAs. When the complex system is non-hierarchically decomposable, an adequate strategy for dealing with the internal coupling has to be applied. This is the case when the values of the input variables to the system are set by the designer, as well as when the values are given automatically by an optimisation algorithm. In this latter case, as will be discussed in the following chapter, the application of appropriate techniques is fundamental.

1.4 Multiobjective optimisation and the Pareto frontier

At the conceptual design phase the aim of the designer will not be that of simply developing an aircraft capable of satisfying a set of requirements; his/her aim will be that of designing the best aircraft possible. The concept of “best” is obviously a relative concept. When comparing different solutions, one can be considered better than another one only with respect to a specific comparison criterion or objective. As an example, in aircraft conceptual design a typical objective is that of minimising the takeoff weight of the aircraft.

In optimisation terms the best solution for a particular set of requirements and a particular criterion is said to be an optimal solution. In fact, conceptual design can be seen as an optimisation problem where the requirements coincide with a set of constraints to be satisfied and the objective will be a criterion chosen by the designer.

In real engineering design the number of criteria is usually greater than one. For instance, when designing an aircraft the designer might wish to minimise the fuel capacity, while maximising the range of the aircraft and minimising the life-cycle cost. Such problems are known as *multiobjective* or *multi criteria* optimisation problems (Miettinen, 1999). The distinction between single objective optimisation and the multiobjective optimisation is a result of conflicting criteria. To give a better description of this concept let us consider an optimisation problem where the objective is that of minimising a set of parameters. It is possible that minimising separately different objectives will yield different solutions. In such case minimising one objective doesn't correspond to minimising another one. In fact, in general the solution of a multiobjective optimisation problem is non unique, but an infinite number of solutions exist, each satisfying a different compromise between the objectives.

When performing multiobjective optimisation the common approach is that of defining a set of preferences for each criterion, such preference representing a particular tradeoff or compromise between the criteria. The optimisation problem is then formulated in such a way to integrate the preferences. The methods making use of this approach are frequently referred to as *a priori articulation of preferences methods* (Marler and Arora, 2004) (Figure 1). The application of such methods

presents two main problems: the first is due to the difficulty in defining the preferences by the designer, given an initial lack of knowledge, and secondly the difficulty in formulating correctly the preferences within the optimisation problem.

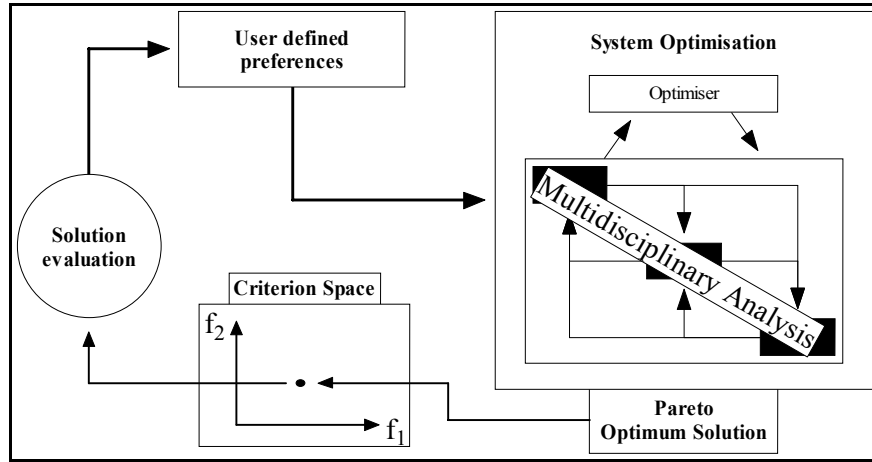


Figure 1: Example of process scheme for *a priori articulation of preferences* methods.

An alternative possibility is that of the Generate First – Choose Later (GFCL) methods (Balling, 2000), or *a posteriori articulation of preferences methods* (Marler and Arora, 2004) (Figure 2), in which a set of solutions are generated, each satisfying a particular combination of preferences or tradeoffs. It is then up to the designer to perform his/her choice. Given the difficulties in defining effectively the preferences the *a posteriori articulation of preferences* approach seems more appealing for the scope of the present work. This considering that in the context of conceptual design calculation time for the analysis of a design point is of several orders of magnitude faster than in more detailed design phases. Therefore performing a significant number of analyses, as required by GFCL methods, becomes acceptable.

As previously mentioned, the solution of the multiobjective optimisation problem is not unique. It is then natural to exclude from consideration any design solution which can be improved without deterioration of any objective and violation of the constraints; in multiobjective optimisation terms, a solution which is dominated. This leads to the Pareto optimal solutions (Miettinen, 1999). Mathematically, each Pareto point is a solution of the multiobjective optimisation problem. In practice, however, the designer would select the ultimate solution among the Pareto set on the basis of

additional (often subjective) requirements. It follows then that the quality of the decisions depends, on the availability of a sufficient number of well-distributed Pareto points ensuring good representation of the entire Pareto surface. Considering the size and complexity of industrial multiobjective optimisation problems such as in aircraft conceptual design, it also follows that the effectiveness of the optimisation methods used for tradeoff analysis is of great importance.

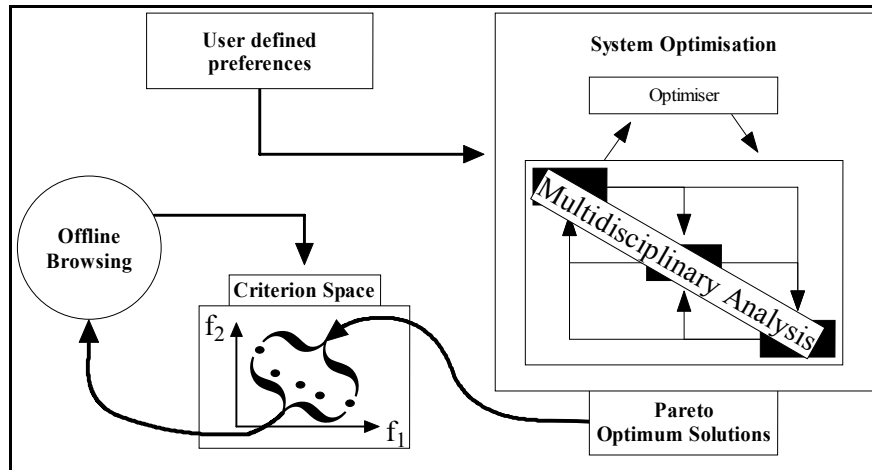


Figure 2: Example of process scheme for *a posteriori* articulation of preferences methods.

1.5 Research Aims and Objectives

When looking at conceptual design from an optimisation point of view, the problem to be solved corresponds to a multiobjective multidisciplinary constraint optimisation problem, the solution of which is not unique. Furthermore, when performing an optimisation of a non-hierarchically decomposable system the additional problem of solving the coupled system has to be taken care of.

In this context the aim of the presented work is that of tackling the problem of performing multiobjective MDO at the conceptual design phase, allowing an effective investigation of the design space. In fact, rather than substituting the designer, it is of interest to view how optimisation tools could be used to assist the designer. For this reason, it is considered that the optimisation should be applied to investigate the design space in the proximity of a particular reference configuration of interest.

The objective of the work is to identify, compare and where necessary improve the most suitable techniques, as well as schemes for their integration, in order to

perform effectively Multidisciplinary Design and Optimisation (MDO) in the Conceptual phase of the aircraft design.

In the following chapter (Chapter 2) a literature review on relevant MDO algorithms and multiobjective optimisation methods is presented. The techniques that have been investigated include: multi-objective optimisation algorithms, MDO algorithms for treating non-hierarchically decomposable systems and Automatic Differentiation (AD). In Chapter 3 an in depth analysis of the multiobjective optimisation algorithms considered of interest is presented. Through the analysis some areas of potential improvement are identified. To solve them three novel methods are proposed. In Chapter 4 considerations on the integration of the multiobjective optimisation algorithms considered with MDO algorithms, together with considerations on the application of automatic differentiation (AD), are discussed. In Chapter 5 a novel algorithm for global multiobjective multidisciplinary constrained optimisation of complex systems is presented, together with some details on its implementation. In Chapter 6 the results for a set of optimisation problems are given, including the results for an aircraft conceptual design optimisation problem. Finally in Chapter 7 a summary of the research work undertaken is given and the final conclusions are drawn.

2 Literature Review

For identifying the most suitable numerical methods, for application to the conceptual design phase, it is necessary to define the characteristics of the problem to be solved. Through discussion with industry, the following assumptions were made with respect to the conceptual design phase from an optimisation point of view:

- **Multiobjective**

In conceptual design the designer is interested in investigating solutions which are optimal with respect to a number of criteria. The optimisation problem to be solved is then said to be a multiobjective optimisation problem.

- **Constraint**

When designing an aircraft a number of requirements have to be satisfied, such requirements can be stated by imposing that one or more variables are required to be smaller or greater than a particular value.

- **Multidisciplinary**

Aircraft conceptual design is inherently multidisciplinary, thus resulting in the possibility of having to deal with internal couplings between the disciplines.

- **Global and local optimum solutions**

When performing optimisation the aim is that of obtaining a solution which minimises or maximises one or more objectives. In real life optimisation problems it is possible that multiple solutions exist. All such solutions are referred to as local optima, local with respect to a particular region of the design space. The global solution is the best, with respect to the objectives, among all such solutions. When performing optimisation this must be kept into account.

- **Continuous design variables**

It is considered that the optimisation should be applied to investigate the design space in the proximity of a particular reference configuration of interest to the designer. Thus in this research all the integer variables, as number of engines or number of passenger seats, are considered to be initially fixed by the designer.

- **Generally continuous functions**

Because of the nature of the functions used for modelling the aircraft at the conceptual design phase, the functions are considered to be predominantly continuous, thus generally differentiable at least to the first order.

For understanding the difficulties in solving such a problem it is necessary to investigate the existing techniques for MDO and those for multiobjective optimisation. In the following sections a number of concepts will be introduced, after which, a review of the existing techniques for handling the optimisation of complex systems is presented. Following this, existing methods for multiobjective optimisation problem are reviewed. Finally a description of automatic differentiation AD will be given.

2.1 Definitions and problem formulation

Before entering into the details of the problem of multiobjective multidisciplinary optimisation, the following concepts have to be defined:

Design variables vector

$\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ is the vector of the N design variables. The design variables correspond to the independent variables of the problem.

Design space

$X \in R^N$ is the design space of design vector $\mathbf{x}$, thus $\mathbf{x} \in X$. The design space X is defined by setting the ranges of variation of each of the design variables by means

of the lower bound vector $\mathbf{x}_L = (x_{L1}, x_{L2}, \dots, x_{LN})^T$ and the upper bounds $\mathbf{x}_U = (x_{U1}, x_{U2}, \dots, x_{UN})^T$, thus $x_i : x_{Li} \leq x_i \leq x_{Ui}$ for $i = 1, \dots, N$.

Objective variables vector

$\mathbf{y} = (y_1, y_2, \dots, y_M)^T$ is the vector of the M objective variables to be optimised, where $\mathbf{y} = \mathbf{F}(\mathbf{x})$. According to the optimisation problem to be solved, each of the objectives y_i can be minimised or maximised. Without loss of generality, in this chapter only the minimisation case will be considered. In fact, in case an objective $y_i = F_i(\mathbf{x})$ was to be maximised it would be sufficient to substitute $y_i = F_i(\mathbf{x})$ with $\tilde{y}_i = -F_i(\mathbf{x})$.

Criterion space

$Y \in R^M$ is the criterion or objective space of vector $\mathbf{y}$, thus $\mathbf{y} \in Y$. Being $\mathbf{y} = \mathbf{F}(\mathbf{x})$ then X is said to be mapped by $\mathbf{F}$ onto Y .

Multiobjective optimisation problem

The general formulation of the multiobjective optimisation problem is the following:

$$\begin{aligned} & \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \\ & \text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K \\ & \text{and } P \text{ equality constraints: } h_p(\mathbf{x}) = 0, \quad p = 1, 2, \dots, P \end{aligned} \quad (1)$$

where each of the M objective variables $y_i = F_i(\mathbf{x})$ is to be minimised while satisfying the K inequality constraints and the P equality constraints.

Feasible point

A vector $\mathbf{x}$ is said to be feasible, with respect to the optimisation problem (1), if and only if $\mathbf{x}$ satisfies the K inequality constraints and the P equality constraints of problem (1).

Feasible design space

$X^* \in R^N$ is the feasible design space, where each vector $\mathbf{x} \in X^*$ satisfies the K equality constraints and the P inequality constraints of problem (1).

Feasible criterion space

$Y^* \in R^M$ is the feasible criterion or object space, where each vector $\mathbf{y} \in Y^*$ is such that $\mathbf{y} = \mathbf{F}(\mathbf{x}^*)$ where $\mathbf{x}^* \in X^*$.

2.2 Optimisation of non-hierarchically decomposable systems

A complex system can be described as a physical object whose behaviour is determined through a number of contributing analyses (CAs). CAs can be also described as the tractable problems in which the complex system is decomposed. CAs are usually associated with particular engineering disciplines or particular physical parts of a system, but could also not be strictly related to engineering disciplines.

According to the complex system to be analysed, its decomposition can lead to two main different cases: a hierarchical decomposition and a non-hierarchical decomposition (AIAA Technical Committee on Multidisciplinary Design Optimization, 1991). These two cases are depicted in Figure 3.

When a hierarchical decomposition is attainable (Figure 3a) all the disciplines are sequentially connected. Thus, given a set of parameters defining one particular system, all the data required by each CA is available from the previously executed ones. Consequently no possibility of inconsistency is present. In case of a non-hierarchical decomposition (Figure 3b) the disciplines are not all sequentially connected. Thus, complex interactions between the disciplines are present. In such a case the system is also said to be internally coupled. The couplings between the CAs

which do not flow downstream through the decomposition are also frequently termed feedbacks. Because of the presence of feedbacks, a non-hierarchically decomposable system will require the application of convergence techniques in order to obtain consistent results.

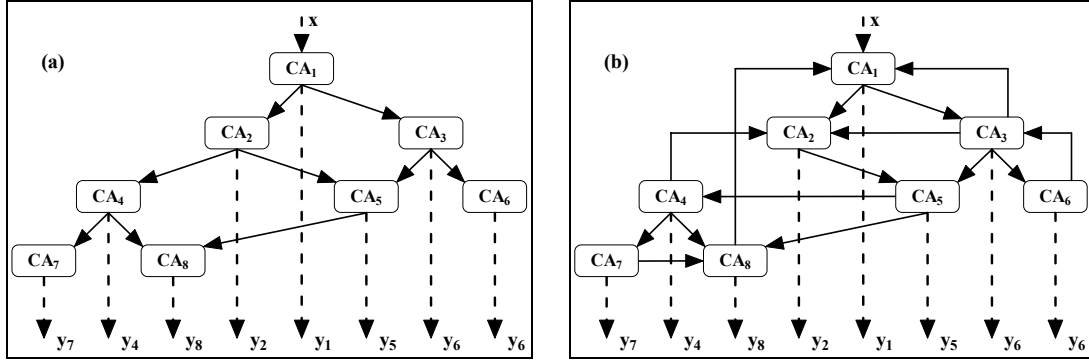


Figure 3: Example of (a) hierarchically decomposable system and (b) non-hierarchically decomposable system.

To be more explicit, given a vector $\mathbf{x}$ of independent variables, the behaviour of the complex system will be determined by the vector of the L dependent variables $\mathbf{z}$ obtained through the execution of the n CAs in which the system is decomposed. The L dependent variables $z_1, \dots, z_L$ are also frequently referred to as coupling variables. Therefore, for evaluating a non-hierarchically decomposable system at point $\mathbf{x}$ the following system has to be solved:

$$\begin{cases} \mathbf{z}_1^* = \mathbf{CA}_1(\mathbf{x}, \mathbf{z}) \\ \mathbf{z}_2^* = \mathbf{CA}_2(\mathbf{x}, \mathbf{z}) \\ \vdots \\ \mathbf{z}_n^* = \mathbf{CA}_n(\mathbf{x}, \mathbf{z}) \end{cases} \quad \text{where} \quad \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_L \end{pmatrix}$$

where vector $\mathbf{z}_i^*$ is composed of the set of coupling variables z_j relative to $\mathbf{CA}_i$.

This can be also written as:

$$\mathbf{z} = \mathbf{S}(\mathbf{x}, \mathbf{z}). \quad (2)$$

Obviously not each of the n CAs will necessarily be coupled with all the others; this must be considered a general case. Nevertheless for a non-hierarchically decomposed system, for every vector of independent variables $\mathbf{x}$, the application of convergence techniques is required in order to obtain $\mathbf{z}$ which satisfies (2).

From (1) and (2) the optimisation problem for a complex system can be formulated as follows:

$$\begin{aligned}
& \min_{\mathbf{x}, \mathbf{z}} \mathbf{F}(\mathbf{x}, \mathbf{z}) \\
& \text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}, \mathbf{z}) \leq 0, \quad k = 1, 2, \dots, K \\
& \text{and } P \text{ equality constraints: } h_p(\mathbf{x}, \mathbf{z}) = 0, \quad p = 1, 2, \dots, P \\
& \text{with } \mathbf{z} = \mathbf{S}(\mathbf{x}, \mathbf{z}).
\end{aligned} \tag{3}$$

Obviously, in case of a hierarchically decomposable system the problem is reduced to a relatively simpler optimisation problem or at the most to a multiobjective optimisation problem (1).

Since the current work is focused on the most general case of MDO of complex systems, the case of non-hierarchically decomposable systems is considered. For such cases different approaches have been attempted in the past decades. The most commonly used algorithms are briefly described in the following paragraphs.

It is to be noted that the aim is not that of fully investigating and evaluating the existing approaches for MDO, for which an extensive literature already exists (Sobieszczanski-Sobieski and Haftka, 1996; Balling and Sobieszczanski-Sobieski, 1994; Kodiyalam, 1998; Cramer et al., 2000; Kodiyalam and Yuan, 2000; Perez et al., 2004), but that of gaining an insight into the difficulties associated with performing MDO. This investigation is conducted, in order to understand which MDO algorithms will be more suitable to be applied in the context of multiobjective optimisation for conceptual design.

Before introducing the algorithms for MDO it is important to introduce the concept of system level optimiser (SLO). An SLO is an optimisation algorithm which is applied to the whole of the system, while a local optimiser can be applied to a particular CA. As it will be seen, an SLO can be applied either to minimise the

objective functions or to enforce the consistency of the coupling, or in some cases to both.

2.2.1 Nested Analysis and Design (NAND)

In the NAND approach (Balling and Sobieszczanski-Sobieski, 1994) for each design point $\mathbf{x}$, requested by the SLO, a full system analysis is carried out in order to provide the SLO with a consistent solution $\mathbf{y}$. As defined by its name the system analysis is nested inside the optimisation process. Such approach is also referred to as Multidiscipline Feasible (MDF) approach (Cramer et al., 2000; Perez et al., 2004), due to the fact that only data $\mathbf{y}$ obtained from a fully converged system, is used by the SLO.

2.2.2 Simultaneous Analysis and Design (SAND)

While in the NAND approach the system consistency is obtained by reiterating the convergence process of the system for each design point $\mathbf{x}$, with the SAND approach (Balling and Sobieszczanski-Sobieski, 1994) consistency is enforced by the SLO itself augmenting the optimisation process. Thus for every design point $\mathbf{x}$ the feedbacks are eliminated and the satisfaction of the feedback couplings is enforced through additional constraints. The SLO will then have to take care of the minimisation of the objective, as well as the consistency of the system through the satisfaction of the additional constraints. Such approach is also referred to as Individual Discipline Feasible (IDF) (Cramer et al., 2000; Perez et al., 2004).

2.2.3 Collaborative Optimisation (CO)

While in all the other MDO approaches the coupling between the disciplines is at the most partially removed, in CO (Kroo et al., 1994; Perez et al., 2004) the couplings are completely removed. While the system optimisation is performed at a system level, enforcement of consistency and satisfaction of constraints is performed for each of the CAs at a local level, through local optimisation procedures.

2.2.4 Concurrent Subspace Optimisation (CSSO)

In the CSSO (Sobieszczanski-Sobieski, 1988) the optimisation is performed at a local level. Subsequently to the evaluation of a base-line design $\mathbf{x}$, Sobieski solves

global sensitivity equations GSE (Sobieszczanski-Sobieski, 1990) to determine for each design variable x_i its influence on the contribution of a given CA to the objective. According to such contribution portions of the $\mathbf{x}$ vector are allocated to the different CAs. For each CA a local optimisation of the allocated variables is then performed in a concurrent fashion. Each local optimisation is performed to improve the design while satisfying a portion of a total cumulative constraint, introduced in order to maintain the consistency of the complete system. Such portion is defined through responsibility coefficient and tradeoff coefficients, according to the responsibility that is evaluated that each CA should have on the enforcement of the total cumulative constraint. Finally a coordination optimisation procedure (COP) has the responsibility of minimising the objective function through the adjustment of the responsibility and tradeoff coefficients.

2.2.5 Bi-Level Integrated System Synthesis (BLISS)

In BLISS (Sobieszczanski-Sobieski, 1998; Kodiyalam and Yuan, 2000; Perez et al., 2004), given a base-line design $\mathbf{x}$, system analyses are computed for each of the CAs as well as sensitivities. Total derivatives are obtained by solving Global Sensitivity Equations (Sobieszczanski-Sobieski, 1990). Local disciplinary optimisations are performed in order to obtain the optimal values of the portion of $\mathbf{x}$ which is local with respect to each CA (local design variables), while the portion of $\mathbf{x}$ shared among multiple CAs is maintained fixed (global or share design variables). The optimisation of each CA is performed in order to minimise the portion of objective relative to the CA according to the total derivatives, while satisfying the set of constraints which are relative to the CA. Finally sensitivities are computed with respect to the global variables and are used by the SLO for minimising the objective function.

2.2.6 Considerations on the reviewed MDO algorithms

Evaluating the previously described approaches two main characteristics can be identified:

- Handling of part or all couplings at a system level (SAND, CO), through compatibility constraints, rather than achieving consistency through convergence techniques for each design point $\mathbf{x}$ evaluated (NAND).
- Handling of the system optimisation at a system level (NAND, SAND, CO) rather than locally for each CA (CSSO, BLISS).

However, for the CSSO and BLISS approaches it is more difficult to subdivide their characteristics in such a precise way, as it can be done for the other approaches described here. CSSO, for example, handles the coupling at first through convergence techniques, in order to attain correct sensitivities. Local level optimisations are then performed, producing inconsistency which will then be handled at a system level.

In the following paragraphs a number of considerations will be made on the characteristics described previously, each of which bringing advantages and disadvantages.

For the coupling handling, the advantages of attaining consistency through convergence techniques for each design point $\mathbf{x}$ is that of having always consistent values of $\mathbf{z}$ and thus of $\mathbf{y}$. The drawback is that for each design point a number of analyses have to be performed. On the other hand, when handling consistency at a system level through compatibility constraints, the advantages are: the removal of the nested iterations and the acknowledgment of the distinctiveness of the disciplines; while the disadvantages are: the increased dimensionality of the optimisation problem carried out by the SLO, and the possibility that consistency is not constantly enforced. In this second case, if during the optimisation the compatibility constraints are not constantly enforced, the complete system optimisation is performed making use of the data obtained from a not constantly converging system; this implies that such data is not always consistent. Using inconsistent data, while attempting to optimise the system, could deceive the optimiser potentially leading to a suboptimal solution.

Before commenting system level optimisation opposed to local level optimisation it is necessary to introduce the concept of tradeoffs. When more than one CA affect one single objective the problem becomes that of handling the tradeoffs; that is the extent to which each of the CAs should influence such objective.

In the case of local level optimisation, an optimisation is performed over each CA, thus multiple CAs will eventually attempt to optimise the same objective. This introduces the need of determining compromises or tradeoffs between the amount of influence that each of the CAs should have on such objective. If this is not done the superposition of the effects produced by the CA optimisations, over the complete system, could produce the opposite effect. Performing optimisation at a local level has anyway its benefits, which is the maintenance of the single expertise of the CAs. It is then clear that the advantage introduced by performing the optimisation at a system level is that of a much greater control over the tradeoffs between the CAs, since the complete behaviour of the system is controlled by one unique optimisation process, with the drawback of losing the single expertise of the CAs.

2.3 Multiobjective optimisation

The main concept associated with multiobjective optimisation is that of Pareto optimal solutions, according to which: a design point $\mathbf{x}$ is Pareto optimal if there is no other point that improves at least one objective function without a loss for any other objective function.

For a multiobjective optimisation problem there is no single solution, but an infinite number of Pareto solutions each of which is optimised for various tradeoffs between the objectives. Therefore, for each set of defined preferences on the relationships between the objectives, one Pareto optimal point can be obtained.

2.3.1 Multiobjective optimisation concepts

Before introducing the numerous methods that have been developed for multiobjective optimisation it is necessary to introduce a number of concepts which will be useful for understanding the problem.

Pareto optimality (Miettinen, 1999; Marler and Arora, 2004)

A point $\mathbf{x}^* \in X^*$ is a Pareto optimal point, with respect to the optimisation problem (1), if and only if there does not exist another point $\mathbf{x} \in X^*$, such that $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$ and $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$ for at least one objective.

Then this coincides with saying that the Pareto optimal solution $\mathbf{x}^* \in X^*$ can not be improved without a loss with respect to at list one other objective.

Weak Pareto optimality (Miettinen, 1999; Marler and Arora, 2004)

A point $\mathbf{x}^* \in X^*$ is a weak Pareto optimal point if and only if there does not exist another point $\mathbf{x} \in X^*$, such that $\mathbf{F}(\mathbf{x}) < \mathbf{F}(\mathbf{x}^*)$.

As from its name the weak Pareto optimality concepts is a weakened version of the Pareto optimality concept. In this case a weak Pareto optimal solution $\mathbf{x}^* \in X^*$ can be improved while maintaining the value of some other objective unaltered.

Non Dominated / Dominated points (Marler and Arora, 2004)

A vector $\mathbf{F}(\mathbf{x}^*) \in Y$ is non-dominated if and only if there does not exist another $\mathbf{F}(\mathbf{x}) \in Y$ such that $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$ with at least one $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$. Otherwise, $\mathbf{F}(\mathbf{x}^*)$ is dominated.

Defined the above concepts it is now possible to review a number of existing approaches allowing to obtain Pareto and weak Pareto solutions.

2.3.2 Multiobjective optimisation strategies and methods

The most common approach in multiobjective optimisation is the *global criterion method* (Marler and Arora, 2004) in which all objective functions are combined to form a single scalar function $AOF(\mathbf{x})$, also called *aggregate objective function* (AOF). Optimisation problem (1) can then be reformulated as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & AOF(\mathbf{x}) = AOF(F_1(\mathbf{x}), F_2(\mathbf{x}), \dots, F_M(\mathbf{x})) \\ \text{subject to } & K \text{ inequality constraints: } \quad g_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K \\ & \text{and } P \text{ equality constraints: } \quad h_p(\mathbf{x}) = 0, \quad p = 1, 2, \dots, P \end{aligned} \quad (4)$$

As shown, through the global criterion method the multiobjective optimisation problem is reduced to a single objective optimisation problem.

One of the most commonly known global criterion method is the *Weighted Sum Method* (WS) (Das and Dennis, 1997; Miettinen, 1999; Marler and Arora, 2004) for which the aggregate function is:

$$AOF(\mathbf{x}) = \sum_{i=1}^M w_i F_i(\mathbf{x}), \quad (w_i \in R, i = 1, 2, \dots, M). \quad (5)$$

Although this method is relatively simple it has two main drawbacks, the first one is due to the fact that the WS is capable of generating only the convex part of a Pareto frontier (Koski, 1985), while problems often result in non-convex Pareto frontier. This drawback can be avoided by using either a more complex consideration of the AOF (Athan and Papalambros, 1996) or the weighted Tchebycheff method, which is an extension of the well-known Min-Max method (Miettinen, 1999; Marler and Arora, 2004).

A second drawback is given by the definition of the values of the weights w_i . If the approach followed is that of the *a priori articulation of preferences* (Marler and Arora, 2004), it is the designer or decision maker (DM) who has to define such values. Since the weights remain to be unknown functions of the objectives (Messac, 2000), a number of iterations may be required in order to obtain the correct combination of weights leading to a solution matching the expectations of the DM (Miettinen, 1999).

An alternative approach is that of the *a posteriori articulation of preferences* or Generate First – Choose Later (GFCL). In such case rather than looking for a single solution a number of alternative Pareto solutions are sought (Marler and Arora, 2004).

Optimisation algorithms based on the GFCL approach aim at determining a number of Pareto optimal solutions or points representative of the Pareto frontier, rather than generating a unique solution representative of a particular tradeoff between the objectives. An extensive review of *a posteriori articulation of preferences* methods is presented in Marler and Arora (2004).

Obtaining a well-distributed set of Pareto points is essential for the GFCL approach to be effective. Since the Pareto front can not be obtained as a function of the design variables $\mathbf{x}$, but only a finite set of Pareto points can be obtained, it is

fundamental that the Pareto points are well-distributed on the full extent of the Pareto front. Hence, allowing a complete representation of the Pareto front in all of its regions.

It has been demonstrated (Das and Dennis, 1997) that attempting to use the WS as a GFCL method, through an even distribution of the weights, doesn't necessarily yield a well-distributed set of Pareto points. Thus, conventional approaches, such as the WS method, are not suitable for generating well-distributed Pareto points. As a result new methods have been recently developed.

Shukla and Deb (2005) subdivide methods for obtaining a representation of the Pareto frontier in two main groups: classical and non-classical methods. Where classical are gradient-based, while non-classical follow "some natural or physical principle" (Shukla and Deb, 2005). They also report, as an additional distinction between the two, that the former are mostly based on the classical approach of reformulating the multiobjective optimisation problem into a single-objective problem, as in the *global criterion method* (4); while the non-classical methods deal directly with the original objective vector $\mathbf{y} = \mathbf{F}(\mathbf{x})$.

Rather than making a distinction between classical and non-classical methods, the latter distinction between "reformulation" and "non-reformulation" methods is considered to be of greater interest for discussion. It is this characteristic, in fact, which defines how the multiobjective optimisation problem is formulated. Therefore it is substantial in defining the approach to be used for solving it.

As has been shown with the AOF, through reformulation the multiobjective optimisation problem is transformed into a single objective optimisation problem (4). This implies that any conventional single objective optimisation algorithm can be applied. Thus, with the AOF approach the problem of dealing with multiple objectives is managed through the reformulation. On the contrary, when the multiobjective optimisation problem is maintained in its original form (1), it is the optimisation algorithm itself that needs to be capable of handling multiple objectives, thus the algorithm has to be specifically developed for such purpose.

Because of this difference, algorithms making use of the reformulation approach require multiple optimisations to be performed. In fact, to obtain each of the Pareto points the formulation of the problem has to be modified, as for example when

modifying the weights w_i of the WS method (5), and an optimisation has to be performed. Shukla and Deb (2005) refer to the principle followed by such methods as *one-at-a-time* strategy. Among these methods some of those considered to be the most interesting for producing well-distributed Pareto points (Marler and Arora, 2004, Shukla and Deb, 2005) are the normal boundary intersection method (Das and Dennis, 1998), the physical programming method (Messac and Mattson, 2002) and the normal constraint method (Messac et al., 2003; Messac and Mattson, 2004).

Methods not making use of the reformulation approach follow a different principle, the *simultaneous* strategy (Shukla and Deb, 2005). The algorithms following this principle are mainly archive or population based and make use of the concept of dominance for dealing with the multiple objectives. Through application of this concept to an archive of points, a number of these can be compared and those which are dominated can be eradicated. Such algorithms are then built specifically for making use of the dominance concept and the optimiser is applied once for obtaining all the Pareto points. Among these algorithms those considered most effective are the evolutionary multiobjective optimisation (EMO) algorithms, among which one of the best known is Deb's NSGA II (Deb et al., 2002).

Because of the characteristics of the conceptual design optimisation problem *one-at-a-time* strategy algorithms are considered more interesting for the purpose of this work. This is mainly due to a number of considerations. Firstly, as stated previously, *one-at-a-time* strategy algorithms can be applied independently with respect to the optimiser used for obtaining each of the Pareto points. This allows using both evolutionary based single-objective as well as gradient based methods for solving each optimisation. Secondly, given the nature of the problem, which is a constraint optimisation problem for which the functions considered are generally differentiable, gradient based methods are considered more effective (Frank et al., 1992). Gradient based methods, in fact, make use of the gradients of the constraints in order to determine a search direction.

Evolutionary Algorithms (EA), to which also the EMO algorithms belong, follow instead a different approach. EA are stochastic population based optimisation methods, for which each chromosome belonging to the population corresponds to a particular design point. At each iteration, chromosomes belonging to one or more

populations are selected according to some criteria and through crossover and mutation operators an offspring is generated.

When dealing with constraint optimisation problems, as postulated by Deb et al. (2002): “Constrained multiobjective optimisation is important from the point of view of practical problem solving, but not much attention has been paid so far in this respect among the EA researchers.”. The constraints, in fact, are handled either through penalty functions or through the selection criteria, but the offspring are still obtained through the stochastic crossover and mutation operators, thus no information on the search direction for satisfying the constraints is used.

With respect to computational cost, it must be considered that one main disadvantage of gradient based methods, with respect to EA, is that obtaining gradients is computationally expensive. But still, Automatic Differentiation (AD) can be used for obtaining analytical derivatives in place of the common and more costly, finite difference approach.

In the following sections the methods which have been considered of greatest interest for the current work are presented. All of these make use of the reformulation approach for generating well-distributed Pareto points. Together with a mathematical description of the formulation of such methods, an easily understandable graphical description is given, in order to allow a better understanding of their differences.

2.3.3 Methods for generating a well-distributed set of Pareto points

For obtaining an evenly distributed set of Pareto points a number of methods have been developed in recent years. All of these are based on performing a subdivision of the criterion (objective) space in a set of domains. The optimisation problem is then reformulated for each domain, for each of which a Pareto point is generated.

The first to provide a method for generating well-distributed Pareto points were Das and Dennis (Das and Dennis, 1998; Das, 1999) with the Normal-Boundary intersection (NBI) method. A second method is the PP-based method developed by Messac (Messac and Mattson, 2002) as an extension of the *a priori articulation of*

preference method known as the Physical Programming (PP) method (Messac, 1996). The new Normal Constraint (NC) method (Messac et al., 2003; Messac and Mattson, 2004) developed recently also looks very promising.

All of the cited methods have a clear geometrical interpretation, they are all based on the well-known fact that a Pareto frontier belongs to the boundary of the feasible space towards the minima of the objective functions (Miettinen, 1999).

In order to provide a geometrical interpretation of the commonalities and differences between the various methods, the two objective optimisation problem shown in Figure 4 will be used.

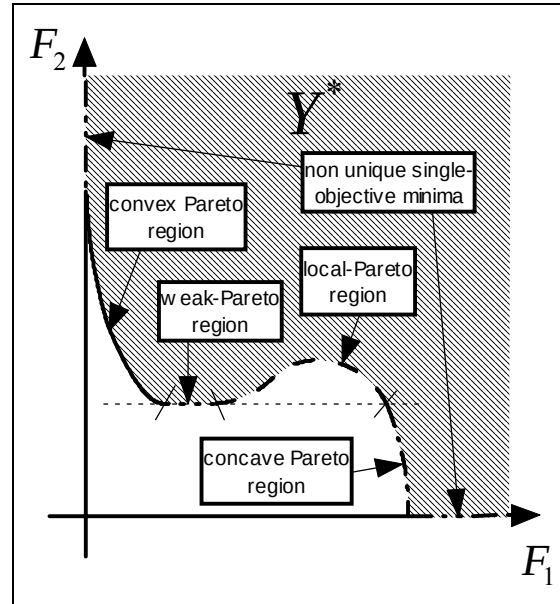


Figure 4: Example of two objectives optimisation problem.

The Pareto frontier of the problem shown in Figure 4 contains all possible kinds of Pareto points:

- **Convex and non convex Pareto regions**

The points belonging to these two regions of the Pareto frontier are all Pareto points since they satisfy the definition for Pareto optimality.

- **Week-Pareto region**

The points belonging to this region satisfy the definition of week-Pareto optimality.

- **Local-Pareto region**

The points belonging to this region are Pareto optimal points, but with respect to a localised region of the design space.

The procedure followed by the NBI, NC and PP-based method can be summarised as follows:

- 1. First one anchor point is obtained for each of the objectives considered.**

A single-objective minimisation is performed for each of the M objectives yielding, as termed by Mattison et al. (2002), *anchor points*, $\mu_i \in \mathbb{R}^M$, $i \in \{1, \dots, M\}$. In other words, an *anchor point* corresponds to the optimal value of one and only one objective function in the feasible space. Thus, M objective functions give M anchor points.

The anchor points are then used for defining the trade-off matrix $\mathbf{T}$:

$$\mathbf{T} = \begin{bmatrix} F_{1,1\min} & F_{1,2} & \cdots & F_{1,M} \\ F_{2,1} & F_{2,2\min} & \cdots & F_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ F_{M,1} & F_{M,2} & \cdots & F_{M,M\min} \end{bmatrix} \quad (6)$$

In the trade-off matrix T , the i -th row represents the coordinates of an *anchor point* μ_i^* corresponding to the solution of the single-optimisation problem $\min(F_i)$ in the feasible criterion space Y^* .

The *anchor points* for the optimisation problem of Figure 4 are shown as white dots in Figure 5. It must be noted that the *anchor points* are not necessarily unique. This can be seen in the example of Figure 4, considering the feasible region Y^* . In this example, in fact, any of the points belonging to the axes and simultaneously belonging to region Y^* would be a solution to the single objective minimisations problems and therefore could be an *anchor point*.

2. Utopia plane points are obtained.

Das and Dennis (1996) refer to the polygon of vertices μ_i^* for $i \in \{1, \dots, M\}$ as *Convex Hull of Individual Minima* (CHIM). Messac et al. (2003) terms *utopia plane*, *utopia hyper-plane* in the general case, the plane to which the M anchor points μ_i^* belong.

Uniformly spaced points belonging to the CHIM are obtained as linear combinations of the M anchor points μ_i^* . Messac (2003) refers to these points as *utopia plane points*. *Utopia plane point* $\mathbf{p}$, belonging to the interiority of a convex polygon defined by M vertices μ_i^* , is represented as:

$$\mathbf{p} = \sum_{i=1}^M \alpha_i \mu_i^*, \quad \text{where} \quad \alpha_i : 0 \leq \alpha_i \leq 1, \quad \sum_{i=1}^M \alpha_i = 1 \quad (7)$$

By evenly distributing the α_i coefficients an even distribution of the points is obtained. This allows generating only points belong to the *utopia plane* and belonging to the interiority of the polygon. It must be noted that the *utopia plane points* are evenly distributed in the criterion space with respect to the *anchor points*, and not necessarily with respect to the design space X .

One of the possible algorithms for calculating the coefficients α_i is given by Das and Dennis (1998) where, in brief, the following induction procedure is used. First, a uniform distribution of coefficient α_1 is considered. The sum of the remaining coefficients $\sum_{j=2}^M \alpha_j$ equals to $1 - \alpha_1$ for each selected value of α_1 . Then, a uniform distribution of the coefficient α_2 is considered for each of these variants. The procedure is then repeated until either the last coefficient α_M is reached or the sum of the coefficients already determined equals 1. In the latter variant the remaining coefficients are equal to zero.

By following the procedure proposed by Das and Dennis (1998) the *utopia plane points* are “ordered on the basis of “nearness””. The scope of the *utopia plane points* is that of being used as reference points, in the criterion space, for reformulating the

optimisation problem. A Pareto point will be sought for each of these points. Thus the ordering of the *utopia plane points* in such a way, allows using the solution of each previously executed optimisation as starting point for the following one.

The *utopia plane points* relative to the optimisation problem of Figure 4 are depicted as black dots in Figure 5.

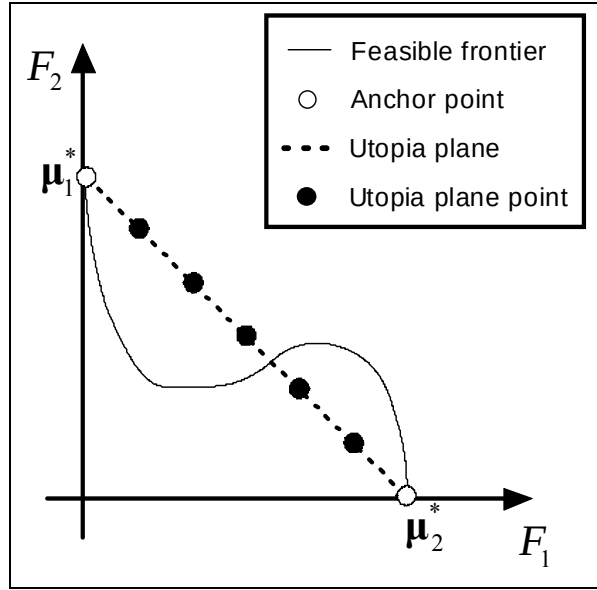


Figure 5: Anchor points, utopia plane and utopia plane points (diagram).

When the number of objectives M is greater than two an additional problem has to be solved. This is given by the possible existence of what will be referred to as *peripheral region*. The *peripheral region* is the portion of the Pareto frontier having orthogonal projection on the *utopia plane*, external to the polygon spanned by the M anchor points μ_i^* (Das and Dennis, 1998; Messac and Mattson, 2004). In Figure 6, taken from Messac and Mattson, (2004), a representation of the *peripheral region* for a hypothetical three objective optimisation problem is shown. The *peripheral region* of the Pareto frontier of Figure 6a would corresponds to the hatched area shown in Figure 6b.

While Das and Dennis (1998) state that the importance of the *peripheral region* is limited and Messac and Mattson (2004) argue that to obtain a complete representation of the Pareto frontier this region should not be ignored, a greater problem has to be highlighted. As will be shown in section 3.1, the size of the

peripheral region depends on the relative position of the *anchor points* μ_i^* , which define the polygon.

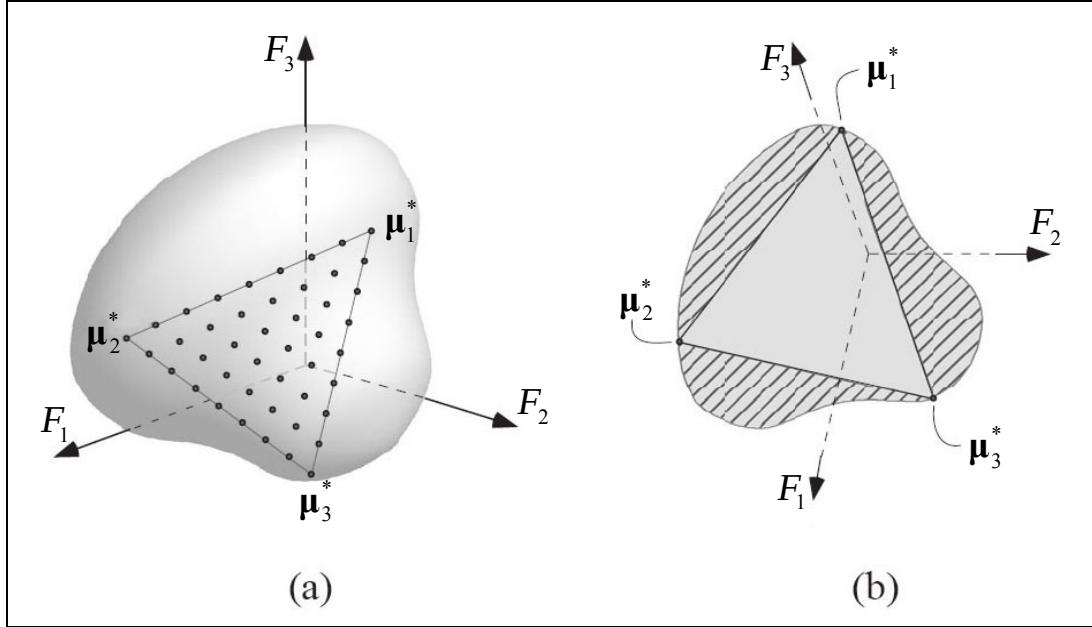


Figure 6: Peripheral region in a 3 objective problem (image taken from Messac and Mattson, 2004).

For tackling the *peripheral region* problem Messac and Mattson (2004) introduce a procedure for enlarging the polygon. For doing this they solve a set of computationally benign optimisations which allow them to obtain the lower limit α_{Li} and upper limits α_{Ui} of α_i , while still imposing $\sum_{i=1}^M \alpha_i = 1$.

The formulation of the benign optimisations for computing α_{Li} is the following:

$$\begin{aligned}
 & \alpha_{Li} = \min_{\mathbf{v}, \alpha} \alpha_i \\
 & \text{subject to } M \text{ equality constraints: } (\mu_r^* - \mu_k^*) \cdot \left(\mathbf{v} - \sum_{j=1}^M \alpha_j \mu_j^* \right) = 0, \quad \begin{matrix} k=1,2,\dots,M, \\ k \neq r \end{matrix} \\
 & \text{subject to equality constraint: } \sum_{j=1}^M \alpha_j = 1, \\
 & \text{and where } \mu_U \leq \mathbf{v} \leq \mu_N
 \end{aligned} \tag{8}$$

where r is arbitrary with the requirement that $r \in \{1, \dots, M\}$, μ_U is such that $\mu_{Ui} = \min\{\mu_{1i}^*, \dots, \mu_{Mi}^*\}$ and μ_N is such that $\mu_{Ni} = \max\{\mu_{1i}^*, \dots, \mu_{Mi}^*\}$. μ_U is termed *utopia point*, since it corresponds to the solution that would be obtained if all of the objectives could be brought simultaneously to their minimum. μ_N is instead termed *pseudo nadir point* and is the point for which all of the objective have simultaneously their worst value. For obtaining α_{Ui} an optimisation problem identical to (8) is solved, except that α_i is maximised.

By solving the M benign minimisation problems (8) and the M benign maximisation problems, Messac and Mattson expand the polygon imposing that any vector perpendicular to the *utopia plane* and passing through any point belonging to the hyper-cube, defined by all points $\mathbf{v}$ such that $\mu_U \leq \mathbf{v} \leq \mu_N$, will intersect the expanded polygon.

Finally, once *utopia plane points* have been obtained, a further benign optimisation problem is solved in order to eventually remove those *utopia plane points* which belong to irrelevant regions of the design space (Messac and Mattson, 2004).

It is to be noted that though the approach has been developed and applied to the NC method it can be applied also to the NBI as well as the PP-based method.

3. For each of the utopia plane points the optimisation problem is reformulated accordingly and an optimisation procedure is performed in order to obtain a single Pareto point.

Steps 1 and 2 allow the computation of a set of points, the *utopia plane points*, which are used as a reference for the subdivision of the criterion space, independently from the method considered. What distinguishes the various methods is how the optimisation problem is reformulated for each of the *utopia plane points*. Regardless of the peculiarities of each method, the aim is that of generating one Pareto point for each *utopia plane point*.

For the NBI method (Das and Dennis, 1998; Das, 1999) for a particular *utopia plane* point $\mathbf{p}$ the optimisation problem is reformulated as follows:

$$\begin{aligned}
 & \min_{\mathbf{x}} t(\mathbf{x}) \\
 & \text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K \\
 & \text{and } P \text{ equality constraints: } h_p(\mathbf{x}) = 0, \quad p = 1, 2, \dots, P \\
 & \text{and subject to the additional constraint: } \mathbf{p} + t\mathbf{n} = \mathbf{F}(\mathbf{x})
 \end{aligned} \tag{9}$$

where $\mathbf{n}$ is the normal to the *utopia plane*. Then for a particular *utopia plane* point $\mathbf{p}$ the additional constraint requires the solution to belong to the normal to the *utopia plane* passing through point $\mathbf{p}$. It is through the minimisation of the scalar t that the minimisation of $\mathbf{F}(\mathbf{x})$ is obtained. The solution of optimisation sub-problem (9) corresponds to the intersection of the normal passing through $\mathbf{p}$ with the boundary of the feasible region, from which the name Normal Boundary Intersection method.

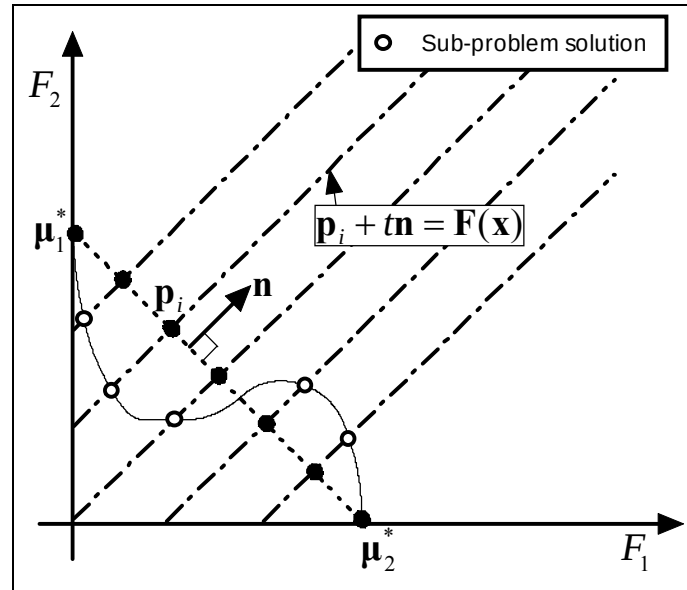


Figure 7: NBI example (diagram).

With respect to the optimisation problem of Figure 4, with *utopia plane points* positioned as in Figure 5, the NBI method would generate the solution shown in Figure 7.

The *a priori* articulation of preference method Physical Programming (Messac, 2000) (PP) is a purely AOF based method. The approach is based on associating each of the objective functions to one of the four possible soft class-functions shown in Figure 8 and four hard functions, the description of which can be found in (Messac, 2000).

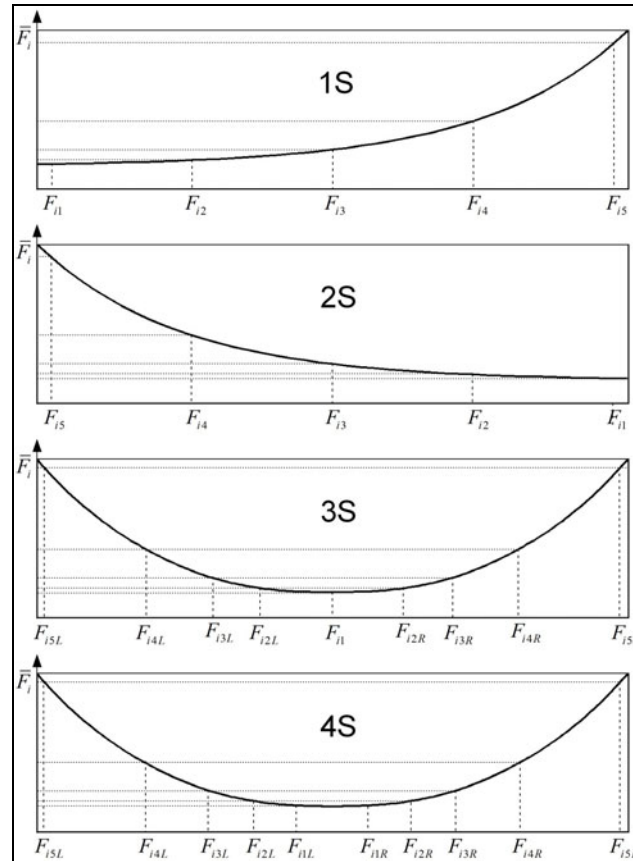


Figure 8: Physical Programming class-functions (adapted from Messac, 2000).

The four class-functions considered are: class 1S (smaller is better) where an objective is to be minimised; its mirror function, function 2S (larger is better) where an objective is to be maximised; class 3S (value is better) where an objective is to be close to a particular preferred value; and class 4S (range is better) which is similar to the 3S function, but the preferable value lies in some range rather than corresponding to some value selected *a priori*. All soft class-functions are positive and dimensionless. The argument of a soft class-function is subdivided by the DM into different preference ranges: highly desirable, desirable, tolerate, undesirable, highly

undesirable and unacceptable through the values F_{ij} . Such subdivision is a part of the approximation of the class-functions and allows the DM to exploit her/his own experience. The values of the class-functions at the boundaries of the ranges are fixed. Therefore, scaling between different objective functions is automatically provided. Furthermore the class-functions are defined in order to avoid introducing additional local minima in the optimisation problem. To guarantee this, each class-function must be a strictly convex function (Messac, 1996), that is: $\frac{d\bar{F}}{dF_i} \in C^1$, $\frac{d^2\bar{F}}{dF_i^2} > 0$.

The problem is then reduced to the following optimisation problem under constraints:

$$\begin{aligned} \min_{\mathbf{x}} AOF(\mathbf{x}) = \min_{\mathbf{x}} & \left[\log_{10} \left(\frac{1}{M} \sum_{i=1}^M \bar{F}_i(F_i(\mathbf{x})) \right) \right] \\ \text{subject to } K \text{ inequality constraints: } & g_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K \\ \text{and } P \text{ equality constraints: } & h_p(\mathbf{x}) = 0, \quad p = 1, 2, \dots, P \\ \text{and subject to the additional constraints:} & \quad \quad \quad (10) \\ & F_i(\mathbf{x}) \leq F_{i5} \quad \text{for 1S} \\ & F_i(\mathbf{x}) \geq F_{i5} \quad \text{for 2S} \\ & F_{i5L} \leq F_i(\mathbf{x}) \leq F_{i5R} \quad \text{for 3S and 4S} \\ \text{and subject to other additional constraints for the hard functions as} & \\ \text{shown in (Messac, 2000).} & \end{aligned}$$

It must be noted that the logarithm is used to diminish the difference between the maximal and minimal values of the AOF.

In (Messac and Mattson, 2002) Messac extends the PP method for *a posteriori articulation of preferences*. For distinguishing it from the PP method it will be called the PP-based method. While for the PP method for each of the objectives the class-function and the preferences are chosen by the DM *a priori*, in the PP-based method the DM chooses only to which of the class-functions each of the objectives belongs. Defining a particular set of preferences corresponds to defining a particular parallelepiped H (Figure 9) of edges $[F_{i1}, F_{i5}]$ or $[F_{i5L}, F_{i5R}] \forall i$ according to the class-functions considered for each of the objectives. The additional constraints are

used for obliging the solution to belong to parallelepiped H . To obtain solutions in various regions of the criterion space the parallelepiped H is shifted, by changing the preferences. For each of the sub-problems, thus for each of the *utopia plane points*, the vertex position is $\mathbf{v} = \mathbf{p}_i + o\mathbf{n}$ where $\mathbf{p}_i$ is the *utopia plane point*, $\mathbf{n}$ is the normal to the *utopia plane* and o is an offset coefficient.

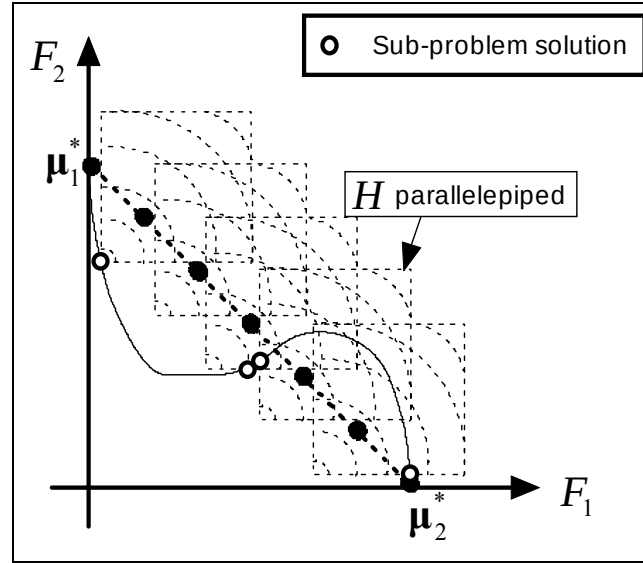


Figure 9: The PP-based method with no offsetting applied (diagram).

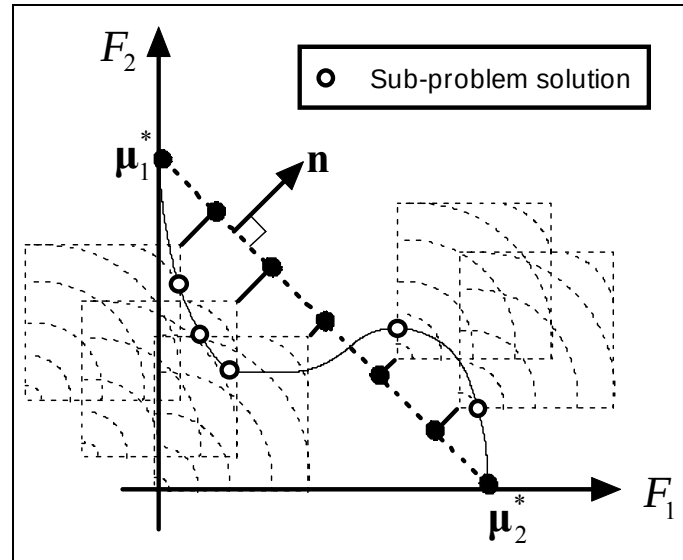


Figure 10: The PP-based method when varying offsetting is applied (diagram).

With respect to the optimisation problem of Figure 4, with *utopia plane points* positioned as in Figure 5, the PP-based method would generate a solution similar to the one shown in Figure 9; this if no offsetting or varying size of H is applied. The solution would obviously be different for different sizes of H or offset. In Figure 10 it is shown that the solutions obtained would be better distributed if applying the offsetting. It is to be noted that the ideal size of parallelepiped H and offset coefficient o for each of the *utopia plane points* are initially unknown and depend on the shape of the Pareto frontier.

The thin dotted lines inside the parallelepipeds in Figure 9 and Figure 10 represent the contour lines of the AOF of the PP-based method, as shown by Messac and Mattson (2002).

The NC method is the most recent method and has also been developed by Messac (Messac et al., 2003; Messac and Mattson, 2004). The NC method is similar to the ε -inequality constraint method (Palli et al., 1998), in fact in both, $M - 1$ objectives are transformed in constraints. Thus the NC method is not AOF based. For the NC method, the reformulated optimisation problem is:

$$\begin{aligned}
& \min_{\mathbf{x}} F_l(\mathbf{x}) \\
& \text{subject to } K \text{ inequality constraints:} & g_k(\mathbf{x}) \leq 0, & k = 1, 2, \dots, K \\
& P \text{ equality constraints:} & h_p(\mathbf{x}) = 0, & p = 1, 2, \dots, P \\
& \text{and subject to the additional } M - 1 \text{ constraints:} & \mathbf{q}_j \bullet (\mathbf{F} - \mathbf{p}_i) \leq 0, & \forall j \in \{1, 2, \dots, M\}, j \neq l
\end{aligned} \tag{11}$$

where $\mathbf{q}_j = \boldsymbol{\mu}_l^* - \boldsymbol{\mu}_j^*$.

To give a better explanation of the NC method we can refer to the example of Figure 4, with *utopia plane points* positioned as in Figure 5. Figure 11 shows a representation of the solutions that the method would yield. In this case objective F_2 is chosen for minimisation, while objective F_1 is treated as constraint. For the sub-problem i , related to *utopia plane point* $\mathbf{p}_i$, the feasible region of the criterion space

corresponds to the shaded area. For this particular sub-problem it is shown that the minima of objective F_2 is not unique since the minima coincides with the weak-Pareto region.

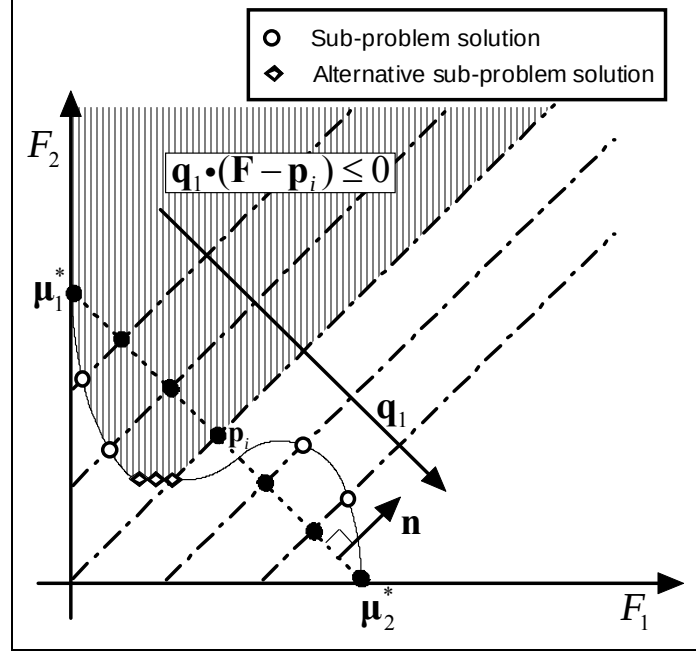


Figure 11: NC example (diagram).

An evaluation of the reported methods for multiobjective optimisation is presented in chapter 3. Through an in-depth evaluation of the characteristics of the methods and some simple test cases, it was possible to identify their respective advantages and shortcomings; the latter forming part of the subject of the research methodology presented in chapter 3.

2.4 Automatic differentiation

Sensitivities can be used for different purposes spanning from application to MDO algorithms, such as the CSSO and BLISS, as well as to be given as input to a gradient based optimiser. Therefore sensitivity is particularly interesting for application in the context of the current research.

Historically the most common approach for obtaining sensitivities is through an approximation of the derivatives by means of finite differences. For the computation of first order derivatives three of the most commonly used forms are the: forward,

backward and central differences. These three forms, for the computation of partial derivatives with respect to x_i , are:

Forward

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) \cong \frac{f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x})}{\|\mathbf{h}\|}$$

Backward

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) \cong \frac{f(\mathbf{x}) - f(\mathbf{x} - \mathbf{h})}{\|\mathbf{h}\|} \quad (12)$$

Central

$$\frac{\partial f}{\partial x_i}(\mathbf{x}) \cong \frac{f(\mathbf{x} + \frac{1}{2}\mathbf{h}) - f(\mathbf{x} - \frac{1}{2}\mathbf{h})}{\|\mathbf{h}\|}$$

with $h_j = 0 \ \forall j \neq i$ and $h_i \rightarrow 0$.

The finite difference approach has two main drawbacks. The first is that, in addition to the round-off error obtained when computing each of the values of f and computing the finite difference itself, a truncation error is introduced. The cause of the truncation error is due to the substitution of the infinitesimal differences with the finite differences. Secondly a single analysis has to be performed for obtaining each of the values of f required. Thus, for computing the gradient of $f(\mathbf{x})$ for $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ the forward and backward forms require $N + 1$ analysis, while the central form will require $2N + 1$ analysis.

An alternative approach to finite differences is provided by Automatic Differentiation (AD), which allows obtaining the derivatives analytically. AD makes use of the chain rule of differentiation, thus the derivatives obtained are exact. Therefore, the truncation error problem of finite differences is removed.

Two main approaches exist for implementing AD: *source transformation* and *operator overloading* (Forth, 2006). With *source transformation* AD software tools are applied to a source code. The source code is then derived and rewritten in order to give as output the partial derivatives of the original output in addition to the original output itself. With *operator overloading* instead, it is not the code which is manipulated, but the data used for computation. This approach is applicable only to

object oriented programming languages, for which new classes of objects together with their relative functions can be built. With *operator overloading*, in fact, a new AD class is defined. When the AD objects are substituted to the traditional input of the code, the operators which would otherwise be used for computation are overloaded by the operators of the AD class. Thus, with such approach the code does not need any modification (Forth, 2006).

Furthermore, two AD modes exist: *forward* and *backward* mode (Barthelemy and Hall, 1992; Bischof and Griewank, 1994; Forth, 2006). *Forward* mode, also known as *bottom up* mode, makes use of the original code in order to compute for each of the instructions the derivatives together with the originally computed values. In *backward* mode, also known as *top down* or *reverse* mode, the code is executed twice. The first execution allows obtaining the data needed for executing the code a second time in reverse mode. The reverse execution allows the output sensitivities to be propagated in reverse through the code. The sensitivities so obtained are also referred to as adjoints.

The advantage of AD, with respect to finite differences, is not only that of removing the truncation error introduced by the finite difference approach, but also computational. In fact, for computing the Jacobian of $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))^T$ for $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ the cost of AD in forward mode is approximately N , while the cost of AD in backward mode is approximately M (Barthelemy and Hall, 1992).

Therefore, making a comparison with the case presented earlier for finite differences, the cost of computing the gradient of scalar function $f(\mathbf{x})$ with $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ with AD would be approximately 1, while with finite differences the cost would be $N + 1$ or $2N + 1$ according to the approach followed.

In conceptual design most of the equations to be dealt with are continuous and differentiable. Considering the case for which discrete variables are fixed, the application of AD becomes an interesting alternative to the finite differences approach.

2.5 Summary

In this chapter the literature review conducted as part of this research is presented. It allowed the identification of a set of suitable methods for performing Multiobjective Multidisciplinary Constraint Optimisation.

The investigation of the existing methods for multiobjective optimisation, of interest for application in the context of aircraft conceptual design, allowed the identification of the NBI, PP-based and NC method. All of these three methods are based on the reformulation approach and allow the determination of well-distributed Pareto points.

For taking care of the inherent multidisciplinary of aircraft conceptual design, five MDO algorithms were identified. The NAND, SAND, CO, CSSO and BLISS algorithms were analysed and compared.

Finally AD was briefly reviewed. AD is, in fact, a valuable tool which allows the determination of partial derivatives at a very low computational cost.

Having defined the methodologies of interest, which are considered promising for application to conceptual design, in the following chapters an in-depth analysis of the various approaches, together with the definition of possible schemas for their integration is presented.

3 From an analysis of the multiobjective reformulation methods to the development of the NC+ method

Through the literature review, the NBI, PP-based and NC method were identified as methods of interest for obtaining the Pareto set. These methods follow a similar approach for obtaining well-distributed Pareto points. The *anchor points*, which are the minima relative to each objective, are obtained first. Subsequently a number of evenly distributed points, the *utopia plane points*, belonging to the criterion space are sought. These are used as reference points allowing the reformulation of the optimisation problem. Finally, for each *utopia plane point* an optimisation is executed in order to obtain a Pareto point.

Each of these three steps is fundamental in order to obtain a complete representation of the Pareto frontier. As will be shown in the next sections, analysing the three methods allowed the identification of some areas of potential improvements, which have led to the development of the modified PP-based method first, then of the Double Hyper-Cone Boundary Intersection method and finally of the NC+ method.

3.1 The peripheral region problem

An even distribution of the *utopia plane points* in the *criterion space* is fundamental for obtaining a complete representation of the Pareto frontier.

When dealing with multiobjective optimisation problems, for which the number of objectives is greater than two, a *peripheral region* exists. The *peripheral region* is that region of the criterion space, for which the orthogonal projection of the Pareto frontier on the *utopia plane* is external to the polygon spanned by the M anchor points μ_i^* (Das and Dennis, 1998; Messac and Mattson, 2004). In their work Das and Dennis (1998) limit the importance of the *peripheral region*, stating that such region will be of no interest to the designer, while Messac and Mattson (2004) are interested in obtaining a complete representation of the Pareto frontier.

Even though Das and Dennis believe that such points might be of no interest for the designer, it must be noted that the size of the *peripheral region* of the Pareto frontier is dependent on the positioning of the *anchor points* and can be significant.

This can be demonstrated by means of the following simple test case:

Problem SPHR_ccv:

$$\begin{aligned} & \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \\ \text{subject to:} & \quad x_1^2 + x_2^2 + x_3^2 \geq 1 \\ \text{and subject to:} & \quad 0 \leq x_i \leq 1 \quad i = 1, 2, 3 \\ \text{where } \mathbf{F}(\mathbf{x}) = \mathbf{x} = & [x_1, x_2, x_3]^T. \end{aligned}$$

The Pareto surface of test case SPHR_ccv, corresponds to the portion of sphere of radius 1 shown in Figure 12, Figure 13 and Figure 14. By following the NBI approach, three single objective optimisations are to be performed in order to determine the *anchor points* μ_1^* , μ_2^* and μ_3^* . The *anchor points* for this test case are non-unique, in fact when minimising objective F_i , any point external to the sphere of radius 1 belonging to the plane defined by axes F_j and F_k , where $i \neq j \neq k$, will be a minima of F_i .

The SPHR_ccv test case allows us to have an immediate visual representation of how the size of the *peripheral region* can change according to the relative positioning of the *anchor points*. Given the fact that the *anchor points* are not unique for this test case we chose three different sets of *anchor points*.

The sets of *anchor points* chosen are the following:

SPHR_ccv-Case1:

$$\mu_1^* = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \mu_2^* = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \text{ and } \mu_3^* = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

SPHR_ccv-Case2:

$$\mu_1^* = \begin{bmatrix} 0 \\ 0.8 \\ 0.9 \end{bmatrix}, \mu_2^* = \begin{bmatrix} 0.9 \\ 0 \\ 0.6 \end{bmatrix} \text{ and } \mu_3^* = \begin{bmatrix} 0.8 \\ 0.6 \\ 0 \end{bmatrix}$$

SPHR_ccv-Case3:

$$\mu_1^* = \begin{bmatrix} 0 \\ 1 \\ 0.3 \end{bmatrix}, \mu_2^* = \begin{bmatrix} 1 \\ 0 \\ 0.3 \end{bmatrix} \text{ and } \mu_3^* = \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix}$$

We then follow the NC approach in order to determine the *utopia plane points*. The *utopia plane* points for Case1, Case2 and Case3 are shown in Figure 12, Figure 13 and Figure 14, respectively. In all figures the polygons corresponds to a triangle and are represented with a dotted line. The view angle of the images is orthogonal to each of the *utopia planes* to which each polygon belongs. The *peripheral region* is shown in each of the three images.

As shown with this test case, the *peripheral region* can be extensive and its size can vary according to the relative position of the *anchor points*. This is particularly clear in Figure 14 for Case3.

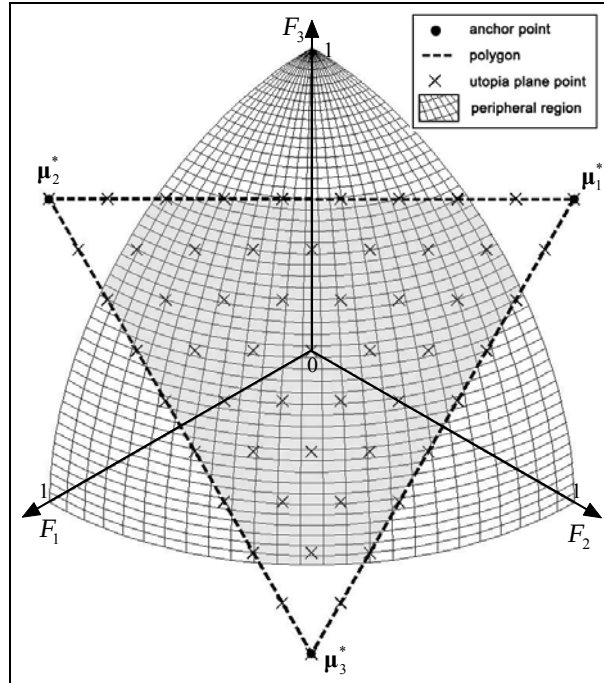


Figure 12: Utopia plane points for problem SPHR_ccv-Case1

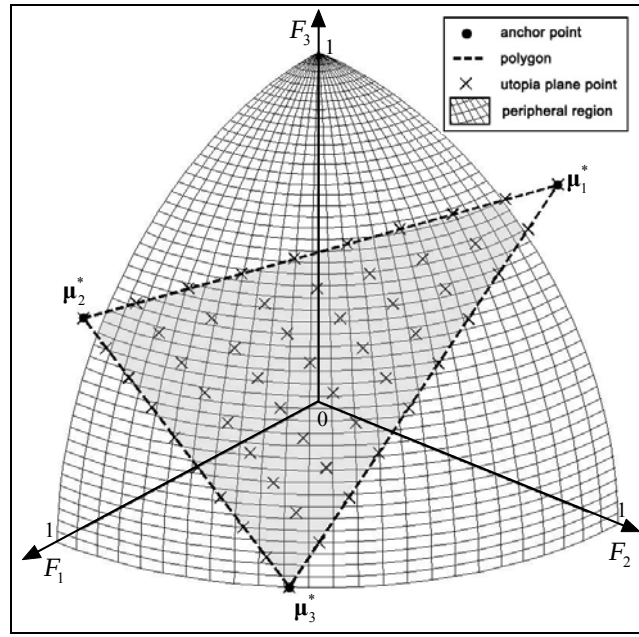


Figure 13: *Utopia plane points* for problem SPHR_ccv-Case2

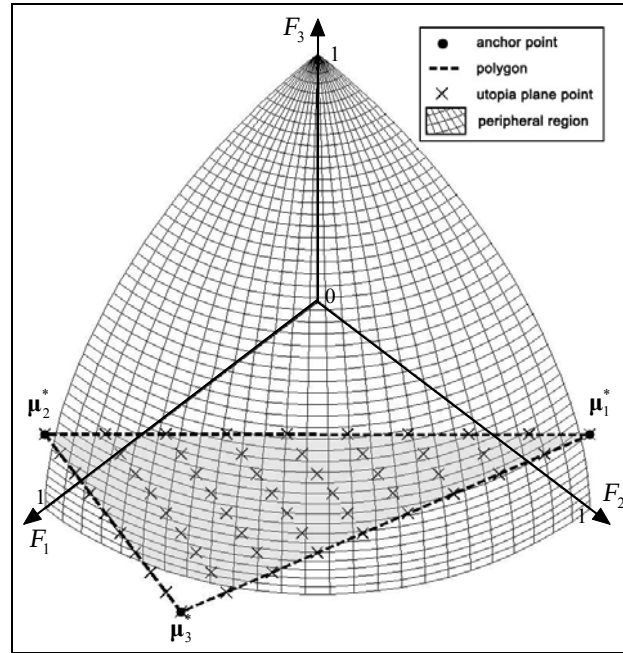


Figure 14: *Utopia plane points* for problem SPHR_ccv-Case3

Even though at first this might seem a particular case, given the non-uniqueness of the *anchor points*, the aim was that of demonstrating how the relative positioning of the *anchor points*, together with the Pareto frontier, can determine the size of the *peripheral region*. In fact, it must be considered that even for a test case for which the

anchor points are unique, their relative positioning could be such that it may prevent obtaining Pareto points on a considerable portion of the Pareto frontier.

As shown in section 2.3.3, Messac and Mattson (2004) proposed a two step procedure for tackling the *peripheral region* problem. The first step allows increasing the size of the polygon, so that any vector perpendicular to the *utopia plane* and passing through any point belonging to the hyper-cube, enveloping the original polygon, intersects the expanded polygon. The second step is the *feasibility test* which allows removing unnecessary *utopia plane points*.

Through the implementation of the method and a number of tests, it was possible to see that the approach suffers from the problem that the number of feasible *utopia plane points* obtained is sensitive to the relative positioning the *anchor points*. This can be demonstrated by considering a generic three objective optimisation problem, for which the Pareto front is contained in the cube for which the three objectives $F_i(\mathbf{x})$ for $i=1,2,3$ are such that $0 \leq F_i(\mathbf{x}) \leq 1$. This test case will be referred to as **GENERIC**.

Once again let's consider the following 3 sets of *anchor points*:

GENERIC-Case1:

$$\boldsymbol{\mu}_1^* = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \boldsymbol{\mu}_2^* = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \text{ and } \boldsymbol{\mu}_3^* = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

GENERIC-Case2:

$$\boldsymbol{\mu}_1^* = \begin{bmatrix} 0 \\ 0.8 \\ 0.9 \end{bmatrix}, \boldsymbol{\mu}_2^* = \begin{bmatrix} 0.9 \\ 0 \\ 0.6 \end{bmatrix} \text{ and } \boldsymbol{\mu}_3^* = \begin{bmatrix} 0.7 \\ 0.4 \\ 0 \end{bmatrix}.$$

GENERIC-Case3:

$$\boldsymbol{\mu}_1^* = \begin{bmatrix} 0 \\ 0.9 \\ 0.8 \end{bmatrix}, \boldsymbol{\mu}_2^* = \begin{bmatrix} 0.7 \\ 0 \\ 0.2 \end{bmatrix} \text{ and } \boldsymbol{\mu}_3^* = \begin{bmatrix} 0.6 \\ 0.3 \\ 0 \end{bmatrix}$$

The “feasible” *utopia plane points* obtained by applying the approach proposed by Messac and Mattson for the three generic cases are shown in Figure 15, Figure 16 and Figure 17, respectively. The figures show the polygon (dotted line) defined by the

three *anchor points*, the cube for which $\mu_U \leq \mathbf{v} \leq \mu_N$ and the enlarged polygon (solid line) which encloses the cube, obtained applying (8). The view angle of the image is orthogonal to the *utopia plane*, to which both polygons belong. For Case2 and Case3 the anchor points were normalized (due to the normalization of the criterion space we obtain a cube rather than a cuboid).

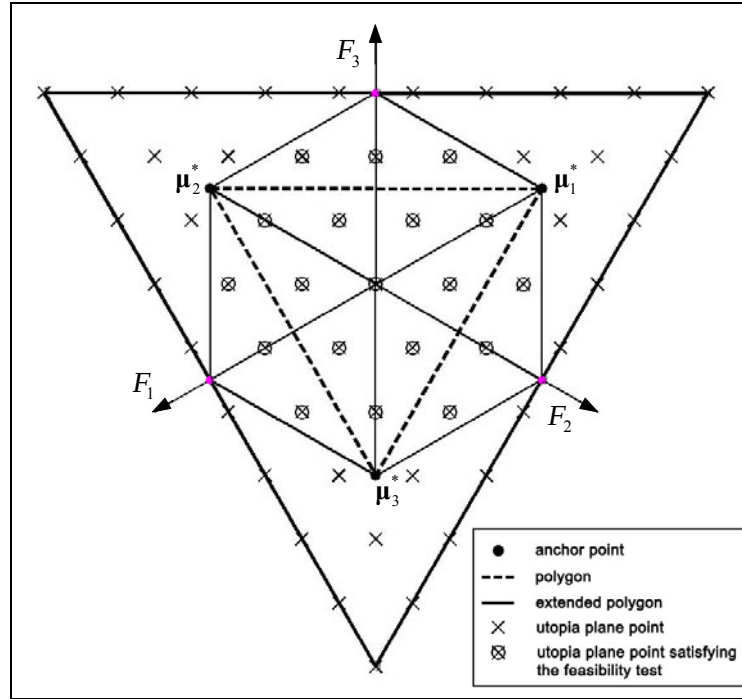


Figure 15: *Peripheral region* for problem GENERIC-Case1

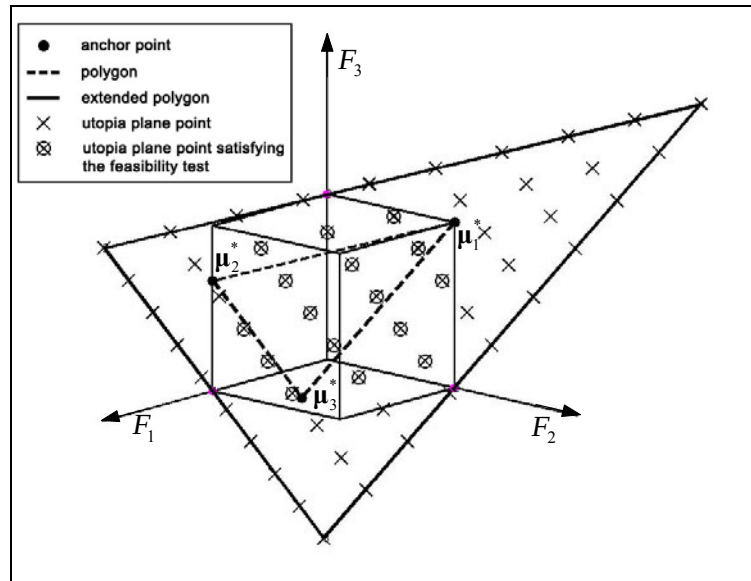


Figure 16: *Peripheral region* for problem GENERIC-Case2

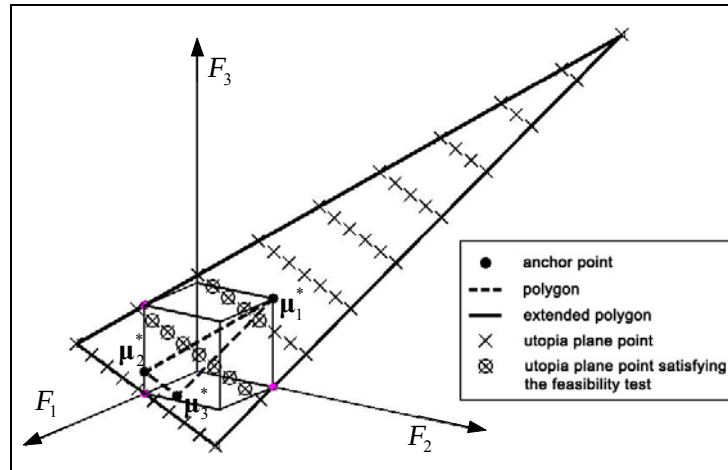


Figure 17: *Peripheral region* for problem GENERIC-Case3

The approach proposed by Messac and Mattson was applied for each of the three cases. The number of *utopia plane points* initially requested was 55. The number of feasible *utopia plane points* obtained for each of the three test cases is shown in Table 1. As shown in the table, the main problem of the approach proposed by Messac and Mattson is that, according to the relative position of the anchor points there can be cases for which only a few *utopia plane points* will satisfy the *feasibility test*. As demonstrated, this is the case even if normalizing the *anchor points*. Furthermore, it must be considered that such problem can be aggravated by an increase in the number of objectives considered; where a “favourable” distribution of the *anchor points* as the one of Case1 (Figure 15) might be rare. Finally, it must be also emphasised that there is no guarantee that a Pareto point will exist for each of the feasible *utopia plane points*. This is due to the fact that some of the feasible *utopia plane points* might be external to the orthogonal projection of the Pareto frontier on the *utopia plane*.

	Requested <i>utopia plane points</i>	<i>Utopia plane points</i> satisfying Messac’s <i>feasibility test</i>	Percentage of feasible <i>utopia plane points</i>
Generic–Case1	55	19	38%
Generic–Case2	55	16	32%
Generic–Case3	55	11	22%

Table 1: Utopia plane points for a generic 3 objective optimisation problem for three different sets of anchor points

3.2 Optimisation problem reformulation

When performing multiobjective optimisation, once a method for reformulating the optimisation problem has been chosen it is applied to each of the *utopia plane points*. For each such point the optimisation problem is reformulated accordingly and an optimisation algorithm is applied in order to obtain a Pareto point. Thus, it is important for the reformulation adopted, to enable the optimiser to be effective in obtaining the Pareto points as a solution.

For allowing the optimiser to work effectively, it is important that the reformulation of the problem does not introduce any additional minima to the original optimisation problem. For what concerns efficiency, it is important that the reformulation minimises the number of optimisations to be performed in order to obtaining the entire Pareto set.

3.2.1 Evaluation of the NBI, PP-based and NC method

For evaluating the NBI, PP-based and NC method the following simple test cases have been considered:

Problem CRC_cvx:

$$\begin{aligned} & \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \\ \text{subject to:} & \quad x_1^2 + x_2^2 \leq 1 \\ \text{and subject to:} & \quad -1 \leq x_i \leq 0, \quad i = 1, 2 \\ \text{where } \mathbf{F}(\mathbf{x}) = \mathbf{x} &= [x_1, x_2]^T. \end{aligned}$$

Problem CRC_ccv:

$$\begin{aligned} & \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \\ \text{subject to:} & \quad x_1^2 + x_2^2 \geq 1 \\ \text{and subject to:} & \quad 0 \leq x_i \leq 1, \quad i = 1, 2 \\ \text{where } \mathbf{F}(\mathbf{x}) = \mathbf{x} &= [x_1, x_2]^T. \end{aligned}$$

Because of the constraints, the Pareto front corresponds to a portion of a circle with radius equal to 1 for both test cases. For the test case CRC_cvx the Pareto front is shown with a dotted line in Figure 18, Figure 20 and Figure 22. The Pareto front for such test case is convex. The Pareto front of test case CRC_ccv is concave and is represented with a dotted line in Figure 19, Figure 21 and Figure 23. For both test cases the number of *utopia plane points* was 11. For each of these points one unique optimisation was performed in order to determine a Pareto point.

For all three methods the optimisation was performed by means of the Sequential Quadratic Programming (SQP) algorithm *fmincon*, which is part of the MATLAB optimisation toolbox (The MathWorks, Inc., 2007a).

- **NBI:**

The reformulation for the NBI method is given in (9). The results for the two test cases for the NBI method are shown in Figure 18 and Figure 19, respectively. As can be seen not all of the solutions found (red points) overlap the dotted line representing the Pareto front. Thus, the optimiser was unable to obtain all the Pareto points. An explanation for such results is given in section 3.2.2.

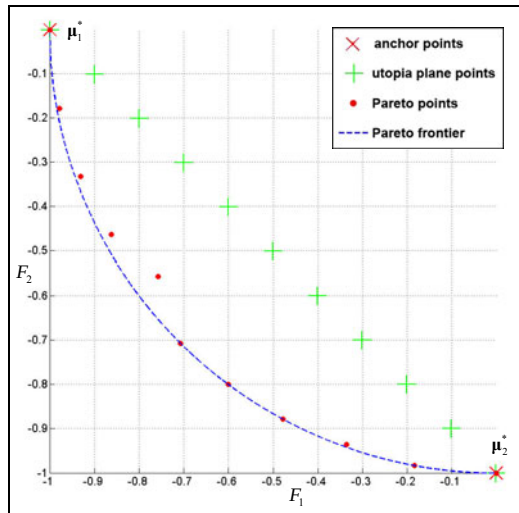


Figure 18: CRC_cvx solution obtained with the NBI method

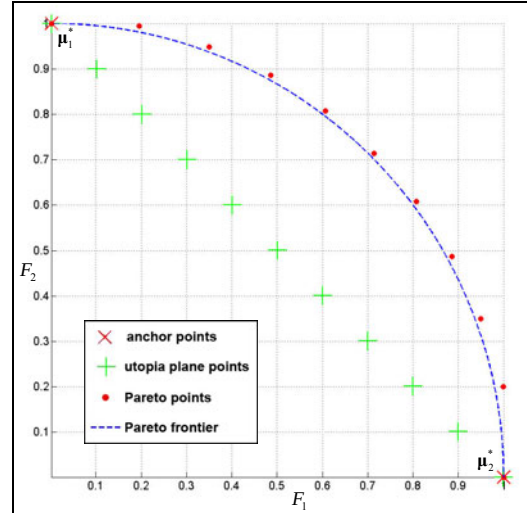


Figure 19: CRC_ccv solution obtained with the NBI method

- **PP-based:**

The results for test cases CRC_cvx and CRC_ccv for the PP-based method are shown in Figure 20 and Figure 21, respectively. The reformulation for the PP-based method is given in (10). For both test cases the class-functions considered were 1S with preferences $[F_{i1}, \dots, F_{i5}] = [0, 0.25, 0.5, 0.75, 1]$ for $i = \{1, 2\}$.

The results for the CRC_cvx test case, shown in Figure 20, were obtained for an offset coefficient $o = 0$. In the figure parallelepiped H is shown, relative to the reformulation for *utopia plane point* $\mathbf{p}_4$. The contour lines show the variation of the AOF (10). The maximum value of the AOF is obtained in correspondence to $\mathbf{p}_4$, the AOF then decreases with a reduction in the values of the objectives. The figure shows that the Pareto points are not well-distributed. A different value of o or a different set of preferences can be used for improving the distribution of the Pareto points, but this would require additional optimisations to be performed.

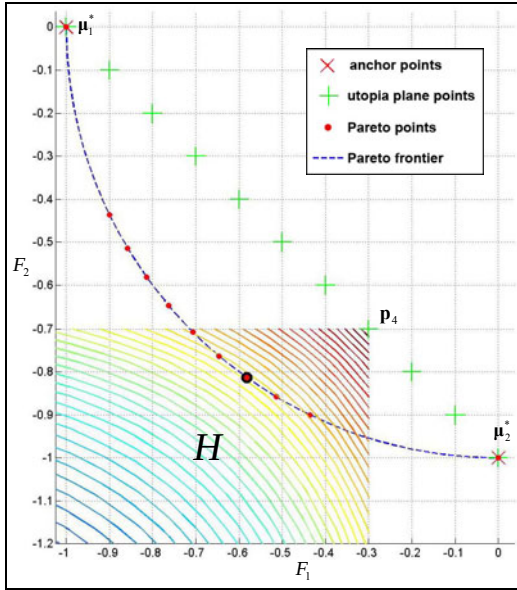


Figure 20: CRC_cvx solution obtained with the PP-based method with no offsetting

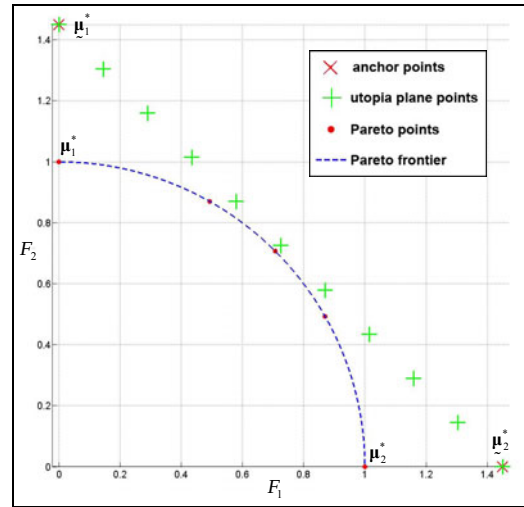


Figure 21: CRC_ccv solution obtained with the PP-based method with no offsetting

For the CRC_ccv test case, if *anchor points* μ_1^* and μ_2^* as shown in Figure 21 were to be used, no Pareto point would be found. This is because the Pareto front lies to the right of the *utopia plane*. Therefore, for testing the algorithm the *anchor points*

were substituted with points $\tilde{\mu}_1^*$ and $\tilde{\mu}_2^*$, in place of offsetting the *utopia plane points*. As shown in Figure 21 apparently only 5 Pareto points were obtained, but in truth, the solutions obtained for multiple *utopia plane points* coincided. Thus only a limited representation of the Pareto front was obtained. As stated for the CRC_cvx test case, the solution could be improved by modifying the preferences or the offset value, which would require additional optimisations.

- **NC:**

The reformulation for the NC method is given in (11). The results for test cases CRC_cvx and CRC_ccv are shown in Figure 22 and Figure 23, respectively. All solutions found are Pareto points. The figures show that the Pareto points are well distributed along the Pareto front. However, as it will be shown in the next section, according to the shape of the Pareto front, the reformulation can allow the optimiser to become trapped in a local minimum.

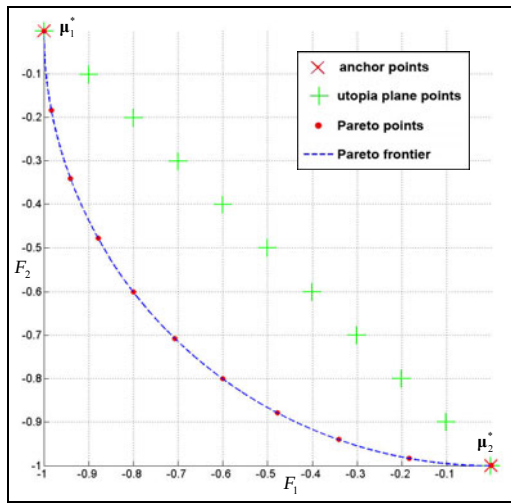


Figure 22: CRC_cvx solution obtained with the NC method

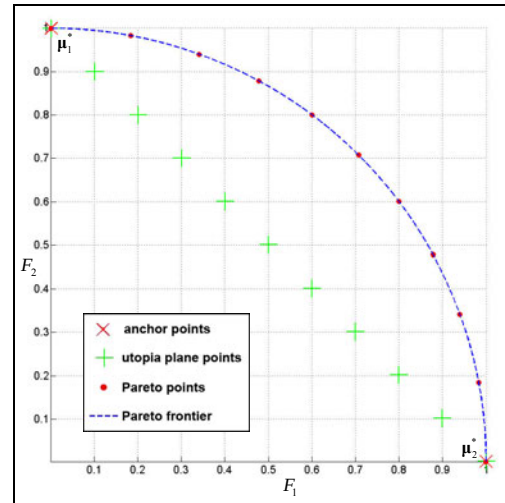


Figure 23: CRC_ccv solution obtained with the NC method

By investigating the methods for a number of different test cases it was possible to identify a major flaw in the NC method. This problem can be seen by means of the results obtained with the implemented NC method, for test case SPHR_ccv of section 3.1. The results shown in Figure 24 correspond to the following sets of *anchor points*:

SPHR_ccv-Case1: (see section 3.1)

SPHR_ccv-Case4:

$$\boldsymbol{\mu}_1^* = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \boldsymbol{\mu}_2^* = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ and } \boldsymbol{\mu}_3^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

As can be seen for Case1 (Figure 24a) the solutions (red dots) are well distributed over the Pareto frontier, represented by the portion of the sphere. For Case2 (Figure 24b) most of the solutions are not obtained on the Pareto front, except for the solutions which coincide with *anchor point* $\boldsymbol{\mu}_2^*$.

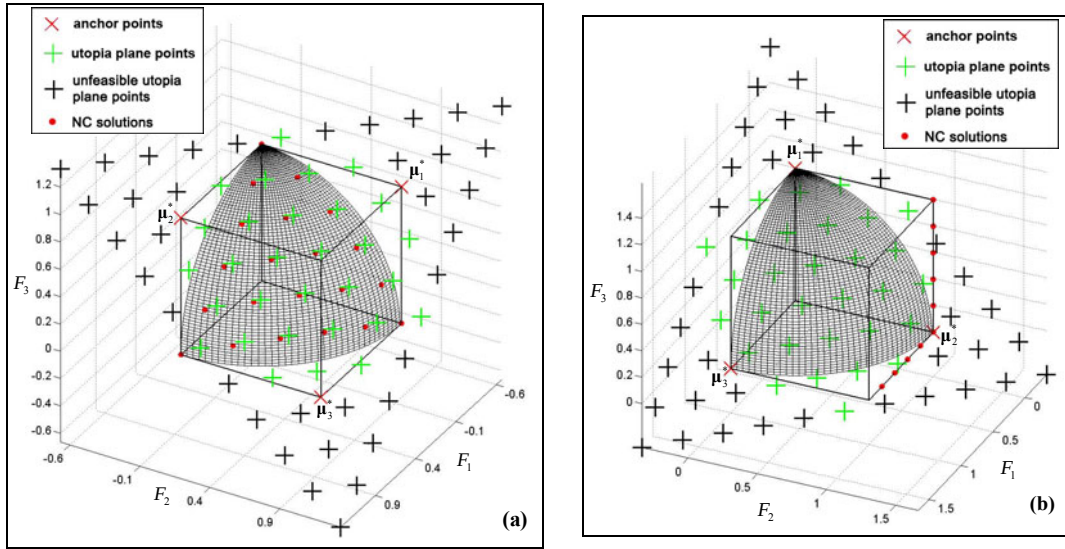


Figure 24: NC solution for problem SPHR_ccv: (a) Case1, (b) Case4

The reason for the failure of the NC method can be explained as follows. From (11) the optimisation problem for *utopia plane point* $\mathbf{p}_i$ can be formulated as follows:

$$\begin{aligned} & \min_{\mathbf{x}} F_3(\mathbf{x}) \\ & \text{subject to:} \quad x_1^2 + x_2^2 + x_3^2 \geq 1, \\ & \text{and } P \text{ equality constraints:} \quad 0 \leq x_i \leq 1, \quad i = 1, 2, 3 \\ & \text{and subject to the additional 2 constraints:} \quad \mathbf{q}_j \cdot (\mathbf{F} - \mathbf{p}_i) \leq 0, \quad j = 1, 2 \end{aligned} \tag{13}$$

where $\mathbf{q}_j = \boldsymbol{\mu}_3^* - \boldsymbol{\mu}_j^*$.

Minimising function $f = F_3$, the gradient of the optimisation problem with respect to the three objective functions is:

$$\nabla f = \left[\frac{\partial f}{\partial F_1} \quad \frac{\partial f}{\partial F_2} \quad \frac{\partial f}{\partial F_3} \right]^T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{e}_3. \quad (14)$$

If vector $\mathbf{l}$ is the unit vector orthogonal to the *utopia plane*, then the projection of ∇f on the utopia plane is:

$$\nabla f - (\nabla f \cdot \mathbf{l})\mathbf{l} = \mathbf{e}_3 - (\mathbf{e}_3 \cdot \mathbf{l})\mathbf{l} = \mathbf{e}_3 - l_3\mathbf{l}. \quad (15)$$

This is shown in Figure 25, where the dashed-dotted line is the *utopia plane*. In the figure the direction which minimises objective function F_3 is considered.

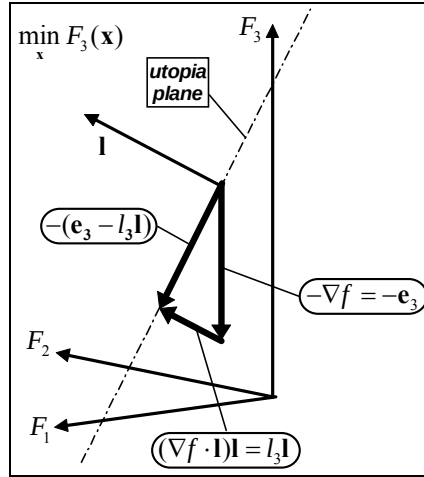


Figure 25: NC method minimisation directions

When optimising for any *utopia plane point*, the component of the gradient of the objective function parallel to the *utopia plane* allows the formulation of the NC method to maintain the solution at the intersection between the $M - 1$ planes; each corresponding to the portion of the frontier of the feasible region determined by one of the additional constraints of the NC method.

For the NC method to be effective it is fundamental for the $M - 1$ planes to be oriented correctly. This can be shown with the three diagrams in Figure 26, for *utopia plane point p*, considering three different sets of hypothetical *anchor points*. The

view of all figures is orthogonal with respect to the *utopia plane*, thus the two planes for which $\mathbf{q}_j \cdot (\mathbf{F} - \mathbf{p}) = 0$ correspond to the dashed lines.

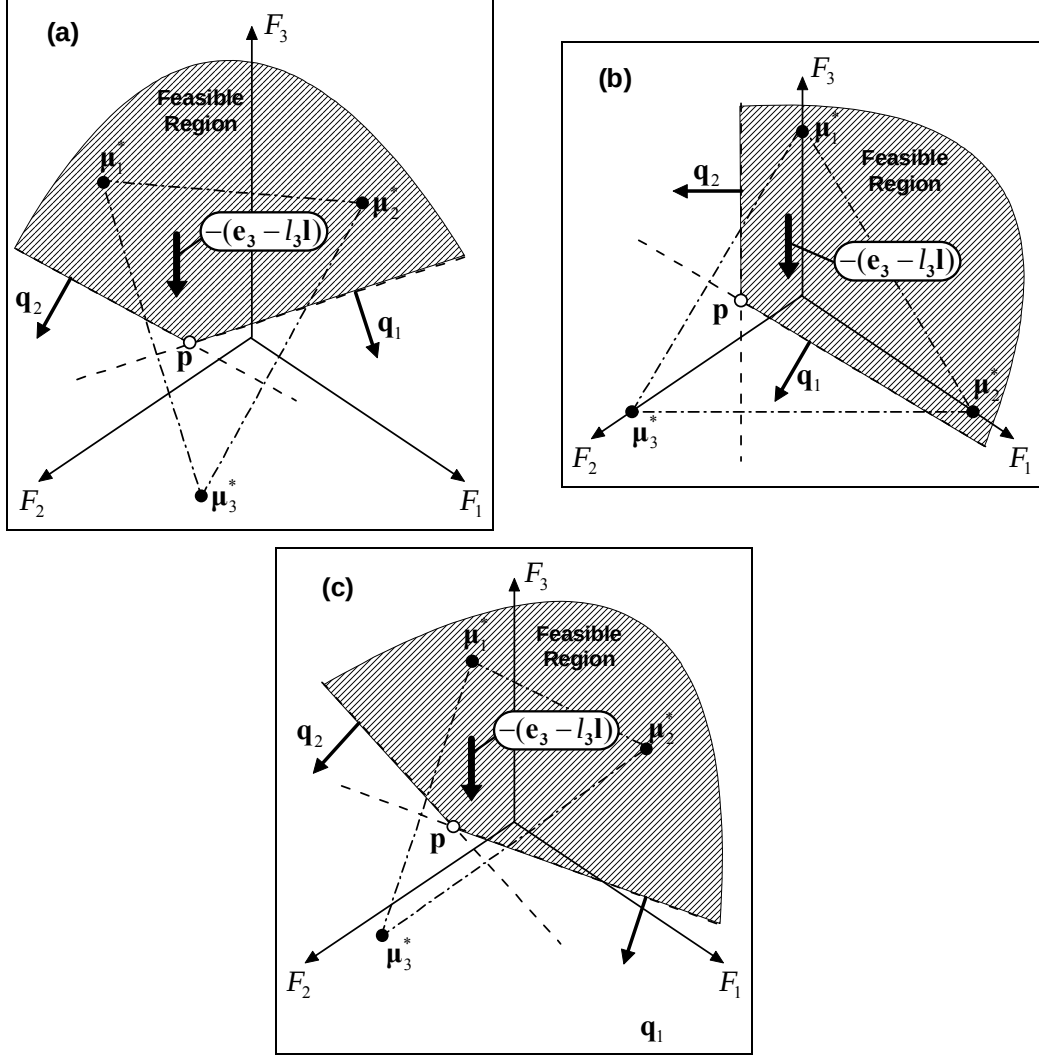


Figure 26: Formulation of the NC method for *utopia plane point p* (viewed orthogonally with respect to the utopia plane), for three different sets of anchor points.

For the *anchor points* of Figure 26a, minimising F_3 , $-(\mathbf{e}_3 - l_3 \mathbf{I})$ (the component of $-\nabla f$ in the plane) will correctly “push” the solution to the intersection of the planes $\mathbf{q}_1 \cdot (\mathbf{F} - \mathbf{p}) = 0$ and $\mathbf{q}_2 \cdot (\mathbf{F} - \mathbf{p}) = 0$, thus the solution will belong to the line passing through $\mathbf{p}$ orthogonal to the *utopia plane*. This case is similar to SPHR_ccv-Case1 (Figure 24a), for which all solutions are obtained correctly.

In Figure 26b the anchor points are positioned as for SPHR_ccv-Case4 (Figure 24b). As can be seen, because of the position of the *anchor points* the two planes $\mathbf{q}_j \cdot (\mathbf{F} - \mathbf{p}) = 0$ are rotated with respect to the case depicted in Figure 24a. Thus the two constraints do not act as required and the solution will not belong to the line passing through $\mathbf{p}$ orthogonal to the *utopia plane*.

Although the case of Figure 26b and that of SPHR_ccv-Case4 (Figure 24b) are particular in that the *anchor points* belong to the axes, (i.e. each minimises two objective functions at once) in Figure 26c we show that this problem can still occur even for a more general case.

As a result, for multiobjective optimisation problems for which the number of objectives is greater than two, the successful application of the NC method is once again, as for the *peripheral region* problem, dependant on the relative positioning of the *anchor points*.

3.2.2 Some considerations with regard to the NBI, PP-based and NC method

The following considerations can be made with respect to the NBI, PP-based and NC method:

- **NBI:**

The only apparent drawback of the NBI method is the tightness of the equality constraints of the reformulated optimisation problem. As shown for cases Problem CRC_cvx (Figure 18) and Problem CRC_ccv (Figure 19), the constraint tightness is such that the optimiser fails to obtain Pareto solution for some of the sub-problems. The feasible region for the sub-problem is reduced in fact to a line. Thus if there are ∞^M points in the global feasible space, there will be ∞^1 points for each of the sub-problems. This reduces the effectiveness of any type of optimiser in obtaining a feasible solution.

- **PP-based**

As discussed in section 2.3.3 for the PP-based method, the ideal size of parallelepiped H , as well as the value for the offsetting coefficient o are unknowns. Therefore, in order to determine well-distributed Pareto points, for each sub-problem

a number of optimisations are necessary in order to determine the optimal size of H and the value of coefficient α . As a result, for each sub-problem several optimisations need to be performed.

- NC

In the NC method, as shown in section 3.2.1, for more than two objectives the method can fail to obtain Pareto solutions, for cases in which the *anchor points* are not “favourably” positioned. Furthermore transforming $M - 1$ objectives in constraints, while minimising only one objective, exposes the optimiser to the possibility of obtaining a solution which does not belong to a line orthogonal to the *utopia plane* and passing through the *utopia plane point* considered. This, in the case of the existence of a local minimum with respect to the single objective minimised. For the example of Figure 4, this possibility is illustrated in Figure 27.

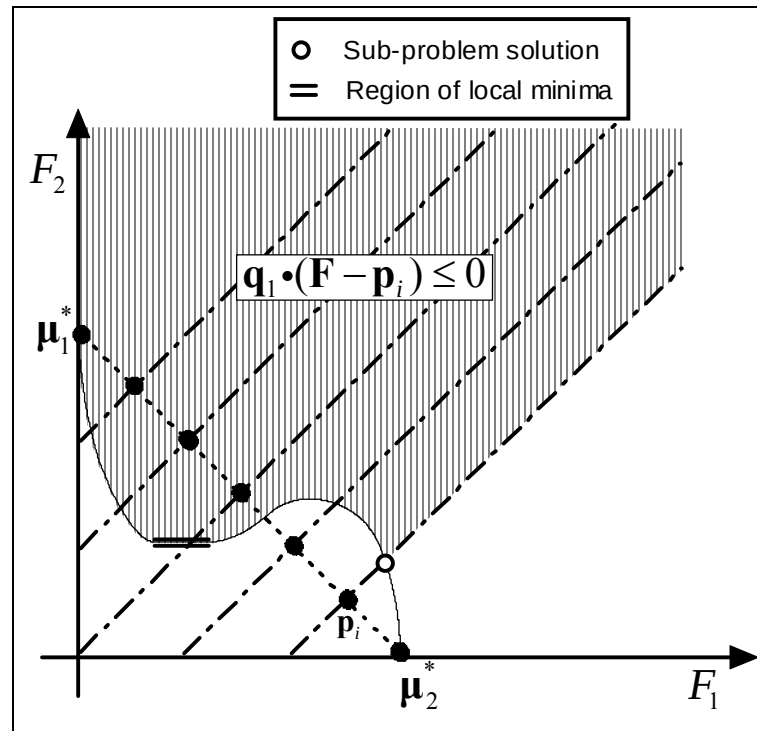


Figure 27: Introduction of the local minima by the NC method.

In Figure 27, for the sub-problem relative to *utopia plane point* p_i , when the reformulation of the NC method requires the minimisation of the single objective F_2 , the feasible region corresponds to the shaded area. In such a case any point of the

weak-Pareto region (double solid line) is a local minima of the reformulated optimisation problem for *utopia plane point* $\mathbf{p}_i$.

3.3 The Global Optimisation problem

For a multiobjective optimisation algorithm to be effective the global optimisation problem has to be taken into account. In fact, in real life optimisation problems it is possible for multiple local minima to exist. As multiple local minima can exist in single objective optimisation, so do multiple local Pareto frontiers in multiobjective optimisation.

Deb (1998) proposes the following definitions for local and global Pareto sets:

Local Pareto-optimal Set: If for every member $\mathbf{x}$ in a set P_{local} there does not exist any solution $\hat{\mathbf{x}}$ satisfying $\|\hat{\mathbf{x}} - \mathbf{x}\|_{\infty} \leq \varepsilon$, where ε is a small positive number (in principle, $\hat{\mathbf{x}}$ is obtained by perturbing $\mathbf{x}$ in a small neighbourhood) dominating any member in the set P_{local} , then the solutions belonging to the set P constitute a local Pareto-optimal set.

Global Pareto-optimal Set: If there does not exist any solution in the search space that dominates any member in the set P_{global} , then the solution belonging to set P_{global} constitutes a global Pareto-optimal set.

As a result, it is possible that in some cases there can exist multiple Pareto fronts, each belonging to a different region of the design space. We can assume that in general the following possible cases can be encountered:

- the global Pareto front coincides with one of the local Pareto fronts,
- the global Pareto front is a composition of portions of different local Pareto fronts.

These two cases are depicted for a hypothetical optimisation problem in Figure 28. In both diagrams, the two differently shaded areas (Y_1^* and Y_2^*) correspond to the mapping on the criterion space of two feasible design regions (X_1^* and X_2^* , respectively) by $\mathbf{F}(\mathbf{x})$. Thus X_i^* for $i=1,2$ is mapped by $\mathbf{F}(\mathbf{x})$ onto Y_i^* . In such case the feasible design space would correspond to $X^* = X_1^* \cup X_2^*$, while the feasible criterion space would be $Y^* = Y_1^* \cup Y_2^*$.

As shown in the diagrams, while in Figure 28a there are two distinctive Pareto fronts in Figure 28b the global Pareto front is a composition of two different portions of the two local Pareto fronts relative to the two feasible regions Y_1^* and Y_2^* .

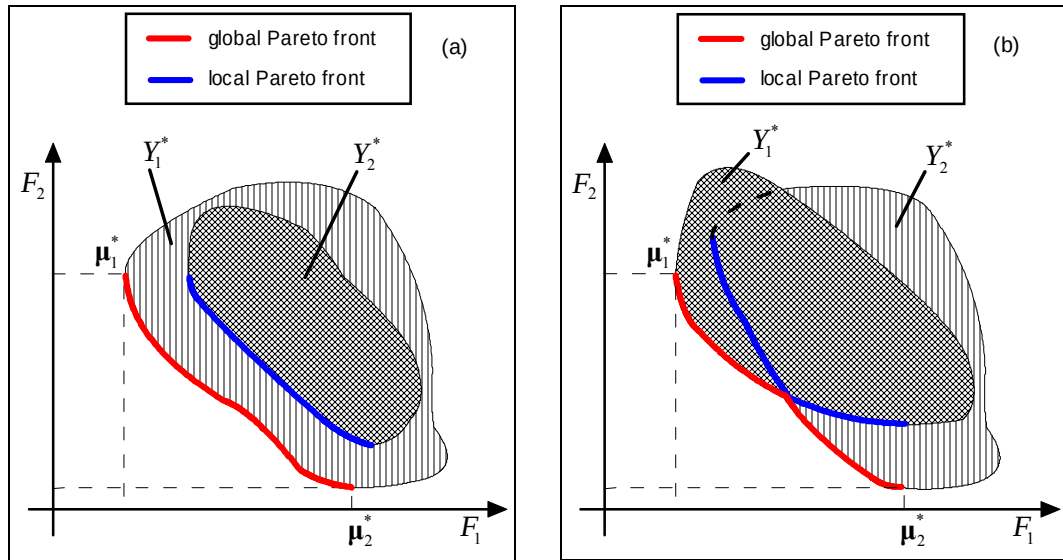


Figure 28: Local and global Pareto optimal sets cases: (a) distinct local and global Pareto-optimal sets, (b) composed local and global Pareto-optimal sets

Methods such as the NBI, PP-based and NC method, which make use of reference points (*utopia plane points*) for reformulating the optimisation problem, are not suitable for multiobjective optimisation problems for which multiple Pareto fronts exist. For all such methods, once the *anchor points* are obtained and the *utopia plane points* are computed, the optimisations are performed sequentially using as starting point the solution of the previous optimisation. As described in section 2.3.3, in fact, the *utopia plane points* are ordered by minimising the distance between each subsequent one and its predecessor. Thus if a local optimiser is used and the solution

of the previously executed optimisation problem corresponds to a point belonging to a local Pareto front, there is a high probability that the solution obtained will still belong to the same local Pareto front. Furthermore, except for the NC method, in case of more than two objectives where the polygon enlargement is performed, the first *utopia plane point* coincides with one of the *anchor points*. Thus if the solution relative to the first *utopia plane point* coincides with an *anchor point*, which coincides with a local minima, there is a high probability that the next solution obtained will belong to the local Pareto optimal front to which also the *anchor point* belongs.

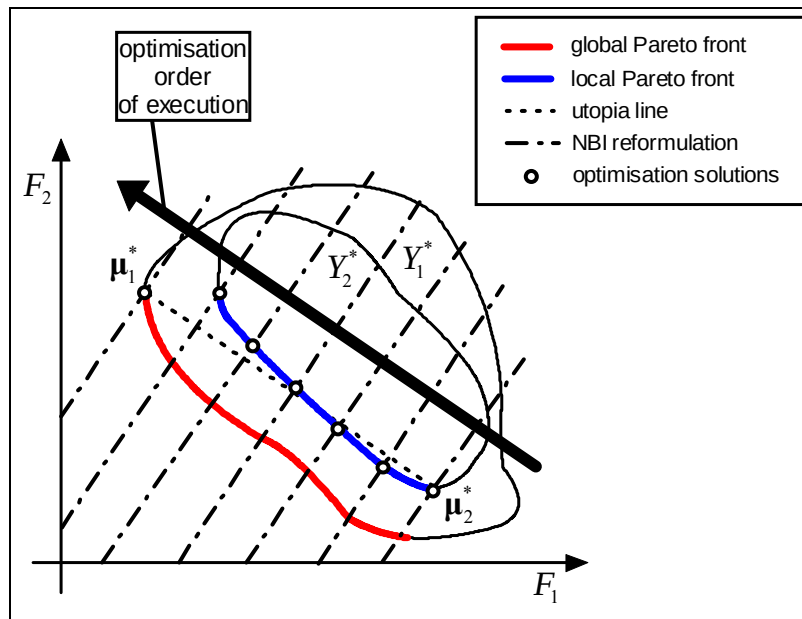


Figure 29: Hypothetical solution for “local” *anchor point* μ_2^*

To give a better understanding of this problem, we give the following example (Figure 29) relative to the hypothetical case depicted in Figure 28a. Let us consider the case in which the *anchor point* μ_2^* , determined for objective function F_2 , coincided with the local minima of Y_2^* as shown in Figure 29. Then the *utopia plane* would correspond to the dotted line connecting the two anchor points. Without loss of generality we represent with the dash-dotted line the additional equality constraints of the NBI method, passing through each *utopia plane point*. By solving each subsequent optimisation problem by means of a local optimiser, in the order shown by the black

arrow, it is possible that the solutions found would correspond to those shown in the figure.

This phenomenon can be demonstrated by means of the following test case.

Problem Deb1:

$$\min[F_1(x_1, x_2), F_2(x_1, x_2)]$$

where: $F_1 = 4x_1$ and $F_2 = g_1(\mathbf{x}) \cdot g_2(\mathbf{x})$

where:

$$g_1 = \begin{cases} 4 - 3e^{-\left(\frac{x_2 - 0.2}{0.02}\right)^2} & (x_2 < 0.4) \\ 4 - 3e^{-\left(\frac{x_2 - 0.7}{0.02}\right)^2} & (x_2 \geq 0.4) \end{cases}, \quad g_2 = \begin{cases} 1 - \left(\frac{F_1}{g_1}\right)^{0.25 + 3.75(g_1 - 1)} & F_1 \leq g_1 \\ 0 & F_1 > g_1 \end{cases}$$

and where $0 \leq x_i \leq 1, i = 1, 2$

This test case was first presented by Deb (1998) for evaluating Evolutionary algorithms. The test case is a particularly complex in that it has multiple local Pareto fronts. Furthermore, the fronts are of mixed type, while the global Pareto front is convex the local Pareto fronts are non-convex. An additional difficulty, when applying methods as those considered (NBI, PP-based and NC), is given by the fact that the single objective minima are non-unique. Thus, the *anchor points* will also be non-unique. Each of the non-unique minima can be considered both as local minima as well as global minima.

The non-uniqueness of the minima of the two objective functions F_1 and F_2 is shown in Figure 30. The figures show the overlapping surfaces relative to the two functions. As shown in Figure 30a the solutions for function F_1 correspond to all the points belonging to the x_2 axis for which $0 \leq x_2 \leq 1$. Figure 30b shows the region for which objective function F_1 is a minimum.

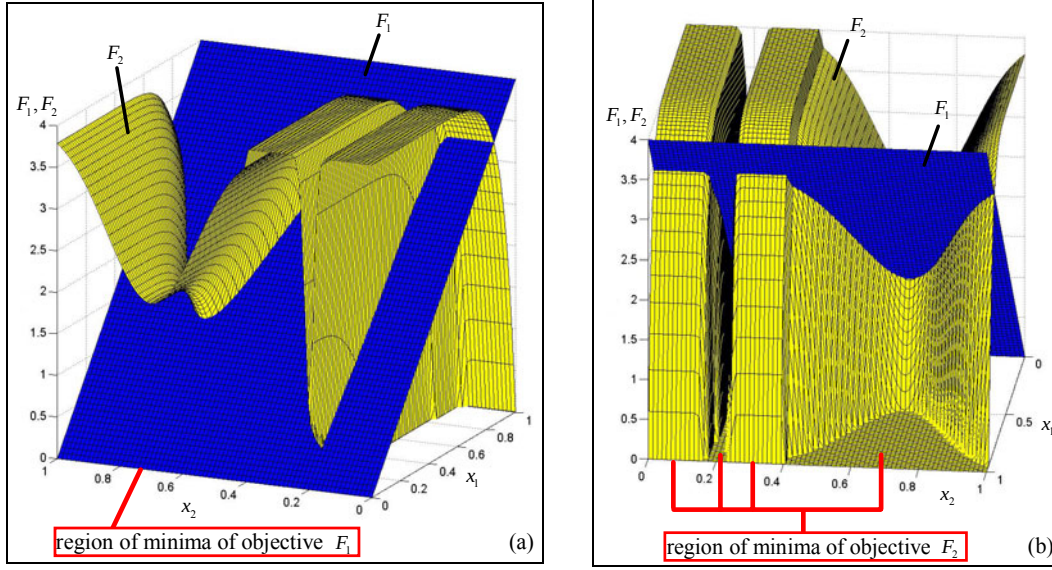


Figure 30: Two views of the surfaces corresponding to objectives F_1 and F_2 with respect to the independent variables for test case Deb1.

As previously stated, when minimising for the single objective functions, in order to obtain an *anchor point*, any points of the region of minima shown in Figure 30 could correspond to an *anchor point*. Let us consider the three following cases:

Deb1–Case1:

$$\mathbf{x}_2^* = \begin{bmatrix} 0.9 \\ 0.6 \end{bmatrix} \rightarrow \boldsymbol{\mu}_2^* = \begin{bmatrix} 3.6 \\ 0 \end{bmatrix} \text{ and } \mathbf{x}_1^* = \begin{bmatrix} 0 \\ 0.28 \end{bmatrix} \rightarrow \boldsymbol{\mu}_1^* = \begin{bmatrix} 0 \\ 3.575 \end{bmatrix}$$

Deb1–Case2:

$$\mathbf{x}_2^* = \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix} \rightarrow \boldsymbol{\mu}_2^* = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \text{ and } \mathbf{x}_1^* = \begin{bmatrix} 0 \\ 0.364 \end{bmatrix} \rightarrow \boldsymbol{\mu}_1^* = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Deb1–Case3:

$$\mathbf{x}_2^* = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix} \rightarrow \boldsymbol{\mu}_2^* = \begin{bmatrix} 4 \\ 0 \end{bmatrix} \text{ and } \mathbf{x}_1^* = \begin{bmatrix} 0 \\ 0.931 \end{bmatrix} \rightarrow \boldsymbol{\mu}_1^* = \begin{bmatrix} 0 \\ 3.47 \end{bmatrix}$$

where $\mathbf{x}_i^*$ for $i=1,2$ correspond to the value of the design variables which yield *anchor point* $\boldsymbol{\mu}_i^*$, thus $\boldsymbol{\mu}_i^* = \mathbf{F}(\mathbf{x}_i^*)$ for $i=1,2$.

For all three test cases the value of $\mathbf{x}_2^*$ was chosen, while $\mathbf{x}_1^*$ was determined through the minimisation of objective function F_1 , with a random point as starting

value for the optimiser. In all three cases the *anchor point* $\mathbf{x}_2^*$ were chosen in order to belong to the regions for which function F_1 is minimum (Figure 30b).

The results obtained by applying the NC method, with respect to the three cases, are shown in Figure 31, Figure 32 and Figure 33, respectively. Once again the optimiser used for solving, for each *utopia plane point*, was the Sequential Quadratic Programming (SQP) algorithm *fmincon* of MATLAB. The two *anchor points* are shown in all figures. The values of x_1 and x_2 of each red vertical line correspond to the values of $\mathbf{x}_i^*$ of each *anchor point* i , while the values of x_1 and x_2 of each black vertical line correspond to the values of the design variables of each solution obtained. The green dots correspond to the intersection of the vertical lines with the two surfaces. Hence they identify the values of F_1 and F_2 for each solution. The solutions were obtained in the order shown with the purple arrow (note that the first solution coincides with *anchor point* $\boldsymbol{\mu}_2^* = \mathbf{F}(\mathbf{x}_2^*)$, according to the ordering of the α_i coefficients (7)).

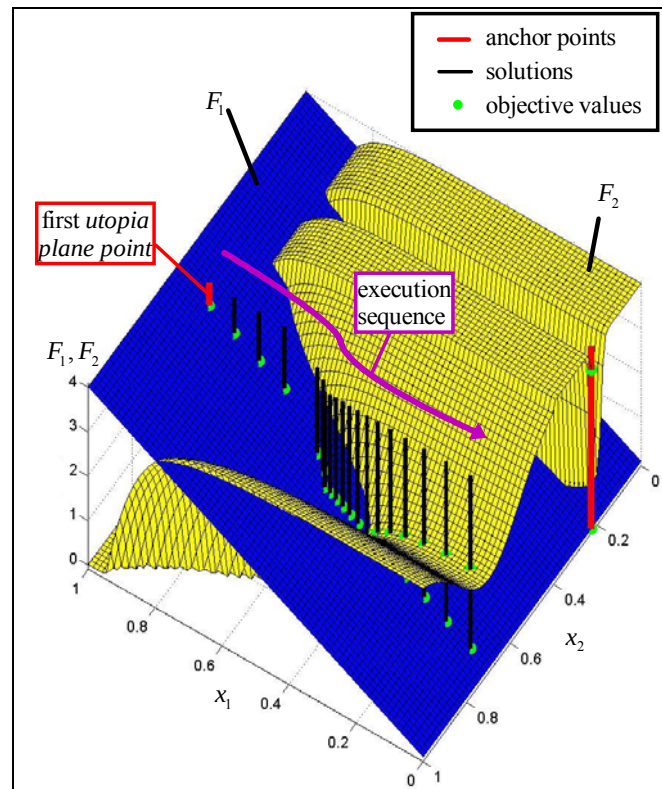


Figure 31: NC solution of test case Deb1–Case1

The three test cases illustrate the global optimisation problem depicted in Figure 29. As shown in Figure 31, Figure 32 and Figure 33, each subsequent solution obtained is in the neighbourhood of the previous one, with respect to the design variables. Thus, because a local optimiser is being used, the subsequent solutions obtained tend to belong to the same local Pareto optimal set to which *anchor point* μ_2^* and therefore the initial *utopia plane point* belongs. The global Pareto-optimal set is obtained for Case2 (Figure 32) where *anchor point* μ_2^* belongs to the global Pareto front. For completeness it must be said that some of the points found are weak Pareto points, nevertheless, these points were in the proximity of a local Pareto front.

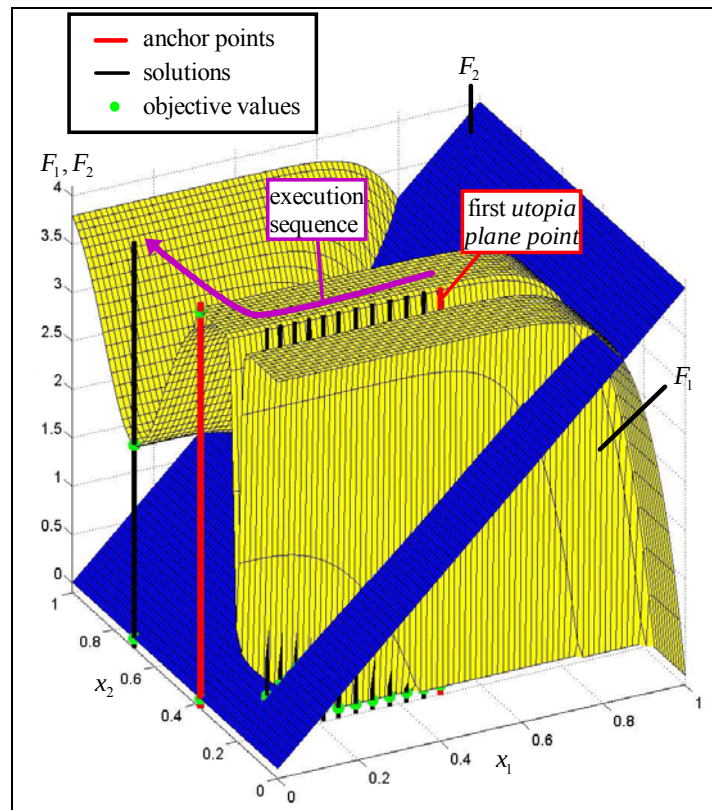


Figure 32: NC solution of test case Deb1-Case2

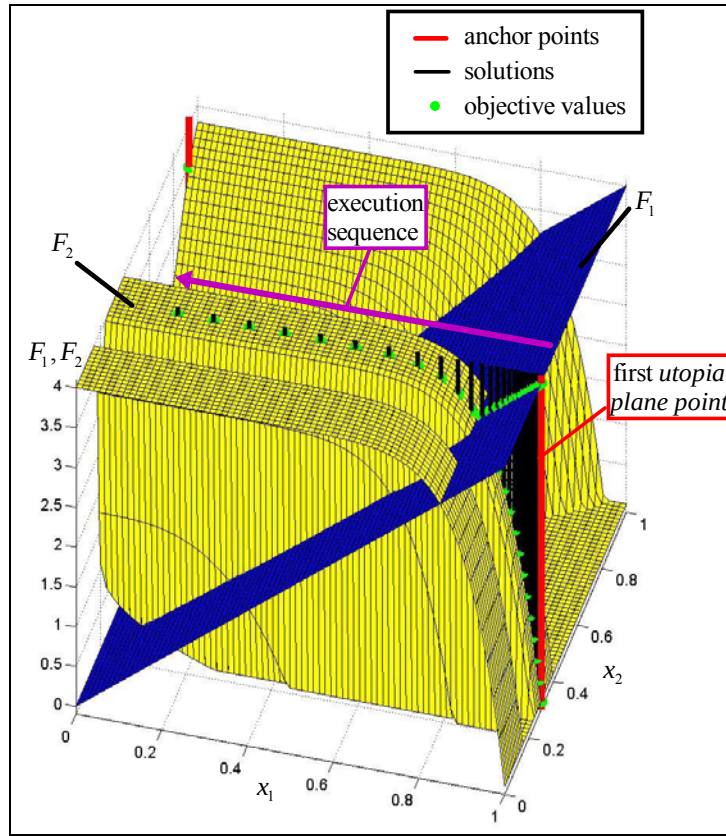


Figure 33: NC solution of test case Deb1-Case3

For a multiobjective optimisation method to be effective it should be applicable to real life design problems, where multiple local Pareto-optimal frontiers can exist. As has been shown such problem is taken care of by neither the NBI, PP-based nor NC method. Thus it is of interest of determining how it could be solved.

Furthermore, in the context of Conceptual design it is of interest for the DM to investigate the design space in search for solutions of interest. The advantage of the *a posteriori articulation of preference* approach is that of enabling an investigation of solutions obtained, allowing the DM to incorporate additional considerations when selecting a solution. For the scope of this research we consider local Pareto fronts as being part of the solutions of interest, we assume that additional considerations (e.g. robustness) might even lead the DM to discard solutions belonging to the global Pareto-optimal front.

3.4 Requirements for an effective multiobjective optimisation method

The previous considerations lead to the definition of a set of requirements which an effective method for generating well-distributed Pareto points should satisfy:

- 1) The reformulation should avoid the introduction of local minima.
- 2) The reformulation should minimise the possibility of the optimiser to fail.
- 3) A unique optimisation should be performed for each of the sub-problems minimising the risk of failure.
- 4) The method should be independent of the number of objectives.

The second point is particularly important for efficiency. In fact, when performing an optimisation relative to a particular sub-problem a set of termination criterion have to be defined; one of these will be a maximum number of iterations. This termination criterion interrupts the optimiser in case a solution is not found within a maximum number of iterations.

The forth point, in conjunction with the second one, is important in relation to the proper determination of the *peripheral region*. The following two sections present the author's effort in developing an approach to satisfy the above requirements.

3.5 The path leading to a new multiobjective optimisation method

The attempt at improving the reformulation of the optimisation problems, in order to remove the limitations associated with the existing methods, has followed three consecutive stages. The result of the first one is the modified PP-based method, developed through modification of the PP-based method (Utyuzhnikov et al., 2005a; Utyuzhnikov et al., 2005b). The second one has taken advantage of the experience gained from the development of the first one and has led to the development of the Double Hyper-cone Boundary Intersection (DHCBI) method (Fantini et al., 2007). The third and last method developed is the NC+ method. The methods follow the

tracks laid by the NBI, PP-based and NC method, combining the various approaches and the knowledge gained from them.

3.5.1 The modified PP based method

As shown above, the NBI method shrinks the search domain to a line, which reduces the feasible space and thus limits the effectiveness of the optimiser. In contrast, the search domains of the PP-based and the NC method are so wide that while in the former method it has to be readjusted, in the latter it may allow the existence of local minima.

Following is a description of the modified PP-based method, which was developed first. The PP-based method has been extensively described in two conference papers: Utyuzhnikov, Fantini and Guenov (2005) and Utyuzhnikov, Guenov and Fantini (2005).

In order to shrink the search domain defined by the parallelepiped H (section 2.3.3), the following generalized class-function is introduced:

$$\bar{F}(\tilde{F}_i), \quad i = 1, 2, \dots, M \quad (16)$$

where $\tilde{F}_i$ is defined by the following affine transform

$$\tilde{F}_i = F_j B_{ji}, \quad i, j = 1, 2, \dots, M \quad (17)$$

in the criterion space Y . This is equivalent to introducing a new coordinate system with the basis vectors

$$\mathbf{a}_i = A_{ij} \mathbf{e}_j, \quad i, j = 1, 2, \dots, M \quad (18)$$

where $\mathbf{e}_j$ for $j = 1, 2, \dots, M$ are the base vectors of the original coordinate system. As a result $B = A^{-1}$.

In particular, the basis vectors $\mathbf{a}_i$ for $i = 1, \dots, M$ can be chosen in such a way to form an angle γ_c with respect to a particular unit vector $\mathbf{l}$. For a 2D case, matrices A and B will be:

$$A = \begin{bmatrix} \cos \gamma_- & \sin \gamma_- \\ \cos \gamma_+ & \sin \gamma_+ \end{bmatrix}, \quad B = \frac{1}{\sin 2\gamma_c} \begin{bmatrix} \sin \gamma_+ & -\sin \gamma_- \\ -\cos \gamma_+ & \cos \gamma_- \end{bmatrix} \quad (19)$$

where $\gamma_+ = \gamma_n + \gamma_c$, $\gamma_- = \gamma_n - \gamma_c$ and $\mathbf{l} = [\cos \gamma_n, \sin \gamma_n]^T$.

A two objective case comparison between the PP-based method and the modified PP-based method is shown in Figure 34. The figure shows how performing the affine

transform, parallelepiped H (Figure 34a) is distorted into the rhombus $\tilde{H}$ (Figure 34b.).

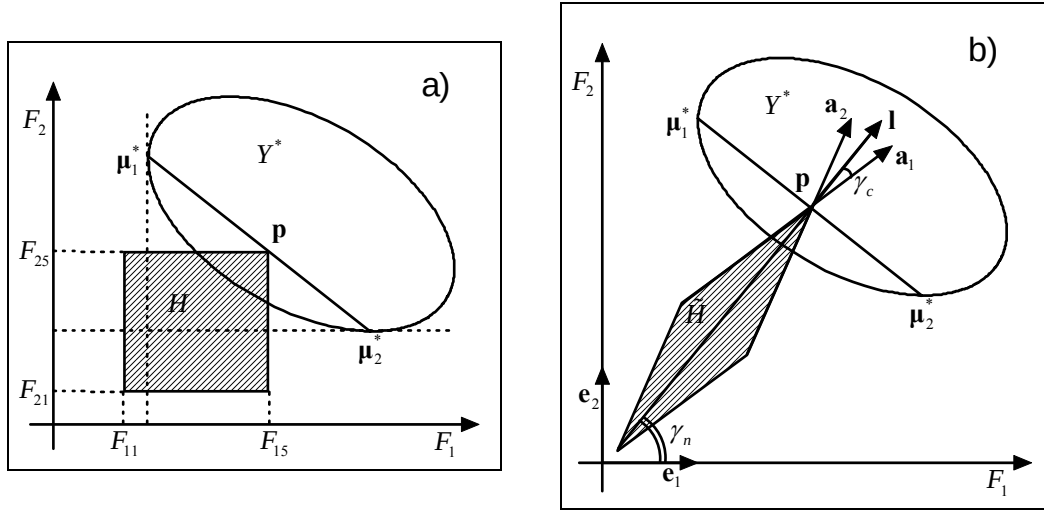


Figure 34: Criterion space: Comparison between the PP-based (a) and the modified PP-based (b) approach

In the general case for, M objectives, the following algorithm is proposed. Suppose that all basis vectors $\mathbf{a}_i$ of the new coordinate system form an angle γ_c with a unit vector $\mathbf{l}$. Thus, all vectors $\mathbf{a}_i$ must satisfy the following conditions:

$$\mathbf{a}_i \cdot \mathbf{l} = \cos \gamma_c, \quad i = 1, 2, \dots, M \quad (20)$$

As a result, all vectors $\mathbf{a}_i$ belong to the lateral area of a hyper-cone with the apex corresponding to a point $\mathbf{p}$, angle γ_c and axis along vector $\mathbf{l}$ (Figure 35).

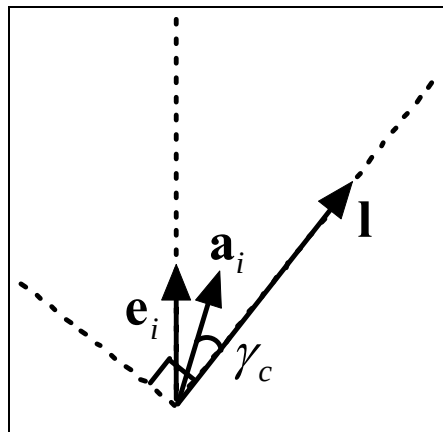


Figure 35: Local basis vectors in an i -th hyperplane

In order to obtain a uniform distribution of the vectors $\mathbf{a}_i$, as in the 2D case, the following procedure is followed. First, we suppose that $\mathbf{l}$ is directed in such a way that it has equal coordinates in the original Cartesian coordinate system $\{\mathbf{e}_1, \dots, \mathbf{e}_M\}$:

$$\mathbf{l} = \mathbf{l}_0 = [l_0, l_0, \dots, l_0]^T \quad (21)$$

Since $\mathbf{l}$ is a unit vector, we obtain its coordinates as:

$$l_0 \equiv \cos \gamma_0 = \frac{1}{\sqrt{M}} \quad (22)$$

Formula (22) immediately follows from the fact that $\mathbf{l} \cdot \mathbf{l} = 1$. Each basis vector $\mathbf{a}_i$ can be determined in the plane created by the vectors $\mathbf{e}_i$ and $\mathbf{l}_0$ as shown in Figure 35. It is possible to show that

$$\mathbf{a}_i = \frac{\sin \gamma_c}{\sin \gamma_0} \mathbf{e}_i + \frac{\sin(\gamma_0 - \gamma_c)}{\sin \gamma_0} \mathbf{l}_0 \quad (23)$$

From (17), (22) and (23) we obtain

$$A = A_0 \equiv \frac{\sin \gamma_c}{\sin \gamma_0} I + \frac{\sin(\gamma_0 - \gamma_c) \cos \gamma_0}{\sin \gamma_0} E \quad (24)$$

where all the elements of the matrix E are unities: $\|E_{ij}\| = 1$.

In many cases the shrinking around the lines parallel to vector $\mathbf{l}_0$ is already sufficient. Nevertheless, it is important to obtain matrix A in the general case of an arbitrary unit vector $\mathbf{l}$. For this purpose, it is sufficient to perform a rotation equivalent to that required to make vector $\mathbf{l}_0$ coincide with vector $\mathbf{l}$. This is achieved by multiplying both parts of equation (23) by an orthogonal matrix R :

$$R\mathbf{l}_0 = \mathbf{l} \quad (25)$$

This allows obtaining the basis of evenly distributed vectors $\{\mathbf{a}'_1, \dots, \mathbf{a}'_M\}$, each of which belonging the lateral area of the hyper-cone having the axis parallel to vector $\mathbf{l}$:

$$\mathbf{a}'_i \equiv \frac{\sin \gamma_c}{\sin \gamma_0} \mathbf{e}'_i + \frac{\sin(\gamma_0 - \gamma_c)}{\sin \gamma_0} \mathbf{l} \quad (26)$$

where $\mathbf{e}'_i = R\mathbf{e}_i$ are the components of the Cartesian coordinate system in which vector $\mathbf{l}$ has coinciding components. The columns of transition matrix R correspond to the coordinates of vectors $\mathbf{e}'_i$ in the basis $\{\mathbf{e}_1, \dots, \mathbf{e}_M\}$. Since the transform is orthogonal, all angles are preserved. In particular, $\mathbf{a}'_i \cdot \mathbf{l} = \cos \gamma_0$.

Matrix A can be obtained in the general form as follows:

$$A = A_0 R^T = \frac{\sin \gamma_c}{\sin \gamma_0} R^T + \frac{\sin(\gamma_0 - \gamma_c)}{\sin \gamma_0} E \quad (27)$$

where $\|E_{ij}\| = \|l_j\|$.

If $\gamma_c = \gamma_0$, then $\mathbf{a}'_i = \mathbf{e}'_i$ which means that the transform becomes orthogonal and is only reduced to a rotation of the original Cartesian coordinate system. As a consequence matrix A is orthogonal and $B = A^T$.

The general presentation requires the calculation of the orthogonal matrix R , the components of which must satisfy the following additional requirements:

$$\cos \gamma_0 \sum_{j=1}^M R_{ij} = l_i \quad (28)$$

Since matrix R is not unique, the simplest way of obtaining it is to consider a rotation from vector $\mathbf{l}_0$ to vector $\mathbf{l}$ in a Cartesian coordinate system $\{\mathbf{b}_1, \dots, \mathbf{b}_M\}$ related to these vectors. Suppose that in this coordinate system vectors $\{\mathbf{b}_3, \dots, \mathbf{b}_M\}$ are orthogonal to both $\mathbf{l}_0$ and $\mathbf{l}$. Then, in this basis vector $\mathbf{l}_0$ is mapped onto $\mathbf{l}$ by a rotation matrix T_R describing the rotation in the hyper-plane created by the vectors $\mathbf{l}_0$ and $\mathbf{l}$. It can then be obtained that

$$T_R = \begin{bmatrix} \mathbf{l}_0 \cdot \mathbf{l} & -\sqrt{1 - \mathbf{l}_0 \cdot \mathbf{l}} & 0 & \dots & 0 \\ \sqrt{1 - \mathbf{l}_0 \cdot \mathbf{l}} & \mathbf{l}_0 \cdot \mathbf{l} & 0 & \dots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & & 1 \end{bmatrix} \quad (29)$$

then, in the original basis the rotation matrix is represented as follows:

$$R = D_R T_R D_R^T \quad (30)$$

Here, matrix D_R is the orthogonal matrix of transition from the original basis to the basis $\{\mathbf{b}_i\}$: $\mathbf{b}_1 = \mathbf{l}_0$, $\mathbf{b}_2 = \frac{\mathbf{l} - (\mathbf{l} \cdot \mathbf{l}_0) \mathbf{l}_0}{\sqrt{1 - (\mathbf{l} \cdot \mathbf{l}_0)^2}}$ and where the remaining basis vectors $\mathbf{b}_3, \dots, \mathbf{b}_M$ can be obtained as follows. Let us supplement vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ to the full basis by those $M - 2$ vectors of the original basis $\{\mathbf{e}_1, \dots, \mathbf{e}_M\}$ that have the minimal value of $\mathbf{l} \cdot \mathbf{e}_k$ for $k = 1, \dots, M$. Then, the full basis $\{\mathbf{b}_1, \dots, \mathbf{b}_M\}$ is obtained through the

Gram-Schmidt orthogonalisation procedure applied to the selected $M - 2$ vectors of the original basis. According to this procedure each subsequent basis vector is to be orthogonal to all the previous ones. Thus, $D_{Rij} = b_{ij}$ for $i, j = 1, \dots, M$, where b_{ij} are coordinates of vector $\mathbf{b}_j$.

Vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_M\}$, as well as $\{\mathbf{a}'_1, \dots, \mathbf{a}'_M\}$, create a basis in Y . The basis vectors form a search cone similar to the 2D case shown in Figure 34b. In the 2D case, where $M = 2$ and $\gamma_0 = \pi/4$, we obtain formulas (19).

The boundaries of the preference ranges are mapped according to (17):

$$\tilde{F}_{ik} = F_{jk} B_{ji}, \quad i, j = 1, \dots, M; \quad k = 1, \dots, 5 \quad (31)$$

As shown in Figure 34, transformations (16) and (17) allow us to shrink the search domain through the constraints $\tilde{F}_i \leq F_{i5}$ for $i = 1, \dots, M$. This allows focusing on a much smaller area on the Pareto front compared to the original PP-based method. Furthermore no displacement of the box D is required in order to obtain well-distributed Pareto points.

The formulation of the optimisation doesn't differ from that of the PP-based method (10). The only difference is that the class-functions $\tilde{F}_i$ will be used in place of the original F_i for building the AOF. Thus the formulation of the AOF to be replaced to the original PP-based method (10) is the following:

$$AOF(\mathbf{x}) = \log_{10} \left(\frac{1}{M} \sum_{i=1}^M \tilde{F}_i(\bar{F}_i(F_i(\mathbf{x}))) \right). \quad (32)$$

A further advantage of the modified-PP-based approach is that the new set of axes can be defined with respect to a vector $\mathbf{l}$ which does not necessarily need to be orthogonal to the *utopia plane*. This is an interesting feature because it allows orienting the search domain in the criterion space. The orientation feature becomes useful when the number of objectives is greater than two, the case in which the *peripheral region* problem exists.

The points obtained are not only Pareto points but can also be weak-Pareto and local-Pareto points. The removal of the weak-Pareto and local-Pareto points is easily attained through comparison, identifying the dominated points in the Pareto set (Utyuzhnikov, Fantini and Guenov 2005).

To avoid undesirable severe skewing of the search domain, preliminary normalisation of the objective functions is performed:

$$F_i^{sc} = \frac{F_i - F_{i\min}}{F_{i\max} - F_{i\min}} \quad (33)$$

similarly to other methods.

Figure 36 shows the results obtained with the modified PP-based method, for test cases CRC_cvx. As for the PP-based method (Figure 20) a search domain relative to one of the *utopia plane points* is shown for comparison.

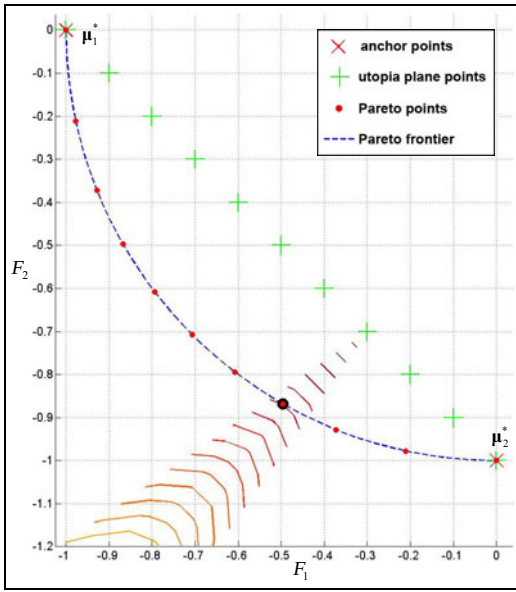


Figure 36: CRC_cvx solution obtained with the modified PP-based method

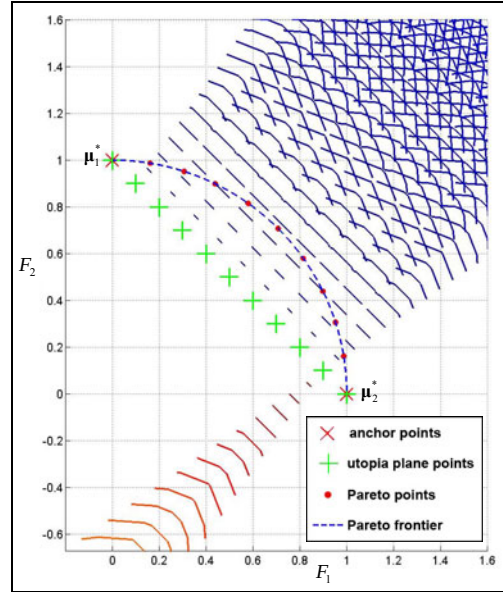


Figure 37: CRC_ccv solution obtained with the modified PP-based method

The contour lines show the values of the AOF. The gradient of the AOF is orthogonal to the contour lines, in the direction for which the objective functions increase. It is also interesting to see, with respect to the PP-method (Figure 20), that the contour lines in Figure 36 are distorted together with the shape of the search domain. For obtaining each Pareto point a single optimisation, for each of the *utopia plane points*, was performed. The modified PP-based method (Figure 36) shows a much better distribution compared to the PP-based method (Figure 20) when no offsetting or resizing is performed.

The results relative to test case CRC_ccv are shown in Figure 37. The figure shows the contour lines relative to the AOF for each of the *utopia plane points*. As

shown, after an initial search on the lower left side of the *utopia plane* (for $-\mathbf{l}$), not finding any solution, the search domain was switched to the opposite side (for $\mathbf{l}$). Comparing the results obtained by the modified PP-based method (Figure 37) and the results obtained with the PP-based method (Figure 21), when no offsetting or resizing is performed, it can be seen that once again the distribution obtained with the modified PP-based method has generated better distributed Pareto points.

One of the advantages of the modified PP-base method is the possibility of rotating the search domain by varying vector $\mathbf{l}$. Such feature is useful in tackling the *peripheral region* problem. By rotating the search domain for the *utopia plane points* which belong to the edges of the polygon of vertices μ_i^* , it is possible to investigate those regions of the criterion space for which their normal projection on the *utopia plane* is external to the polygon.

The following procedure for rotating the search domain is followed.

From (7), a *utopia plane point* $\mathbf{p}$ will belong to the k -th edge of the polygon if and only if $\alpha_m = 0$ for $m \neq k, k+1$.

We define as edge vectors of polygon, vectors (Figure 38a):

$$\mathbf{v}_i = \mu_{i-1}^* - \mu_i^*, \quad i = 1, \dots, M-1 \quad (34)$$

A unit vector which is the outer normal to the edge considered, can be obtained by means of the following linear combination of the edge vectors $\mathbf{v}_{i-1}$ and $\mathbf{v}_i$:

$$\mathbf{s}_i = \frac{\mathbf{v}_{i-1} + \beta_i \mathbf{v}_i}{|\mathbf{v}_{i-1} + \beta_i \mathbf{v}_i|}, \quad \beta_i = -\frac{\mathbf{v}_{i-1} \cdot \mathbf{v}_i}{\mathbf{v}_i \cdot \mathbf{v}_i} \quad (35)$$

Vector $\mathbf{l}$ is determined via $\mathbf{s}_i$ and the normal $\mathbf{n}$ to the *utopia hyper-plane* towards the origin as follows:

$$\mathbf{l} = -\cos \theta_r \mathbf{n} + \sin \theta_r \mathbf{s}_i, \quad 0 \leq \theta_r \leq \pi/2 \quad (36)$$

where angle θ_r is a parameter which defines the rotation of $\mathbf{l}$. Changing θ_r from 0 to $\pi/2$, vector $\mathbf{l}$ is turned from the normal vector $-\mathbf{n}$ to the vector $\mathbf{s}_i$ as shown in Figure 38b.

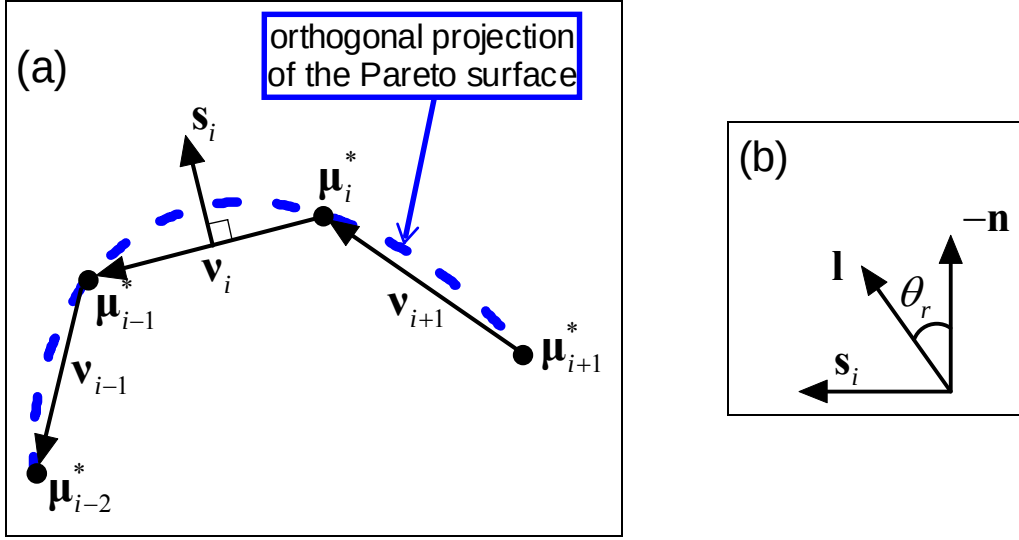


Figure 38: Rotation of the search domain to obtain the Pareto points belonging to the *peripheral region*

The following test case allows us to demonstrate the capability of the modified PP-based method to obtain those points of the Pareto front, for which their normal projection does not fall inside the polygon.

Problem half_SPHR_cvx:

$$\begin{aligned}
 & \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \\
 & \text{subject to:} \quad x_1^2 + x_2^2 + x_3^2 \leq 1, \\
 & \text{subject to:} \quad -1 \leq x_i \leq 0 \quad i=1,2 \\
 & \text{and subject to:} \quad -0.5 \leq x_3 \leq 0 \\
 & \text{where } \mathbf{F}(\mathbf{x}) = \mathbf{x} = [x_1, x_2, x_3]^T.
 \end{aligned}$$

In this case the Pareto surface is represented by the part of the unit sphere for which the coordinates are negative and $x_3 \geq -0.5$. The anchor points are the following:

$$\boldsymbol{\mu}_1^* = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\mu}_2^* = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mu}_3^* = \begin{bmatrix} -\sqrt{3}/2 \\ 0 \\ -1/2 \end{bmatrix}.$$

The results shown in Figure 39 demonstrate that the entire Pareto frontier is represented, regardless of its orthogonal image on the *utopia plane* being far beyond the triangle defined by the *anchor points*. For 66 *utopia plane points*, 127 Pareto points were obtained.

The main drawback of the modified PP-based method is that the domain $\tilde{H}$ can be oriented only on either side of the *utopia plane*. This means that a maximum of two optimisations have to be performed for each *utopia plane point*. Although the modified PP-based method is effective in reducing the search domain with respect to the PP-based method and the NC method, and enlarging the search domain with respect to the NBI method; the NBI and NC method have still the advantage of investigating both sides of the *utopia plane* at once.

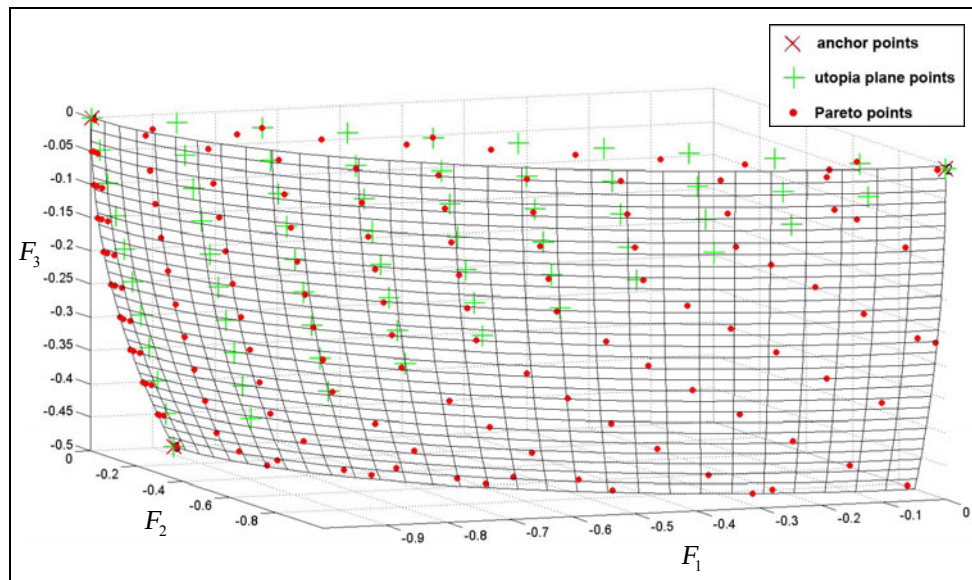


Figure 39: half_SPHR_cvx solution obtained with the modified PP-based method showing the capability of the algorithm in obtaining the solutions belonging to the peripheral region

Each of the search domains shown in Figure 37 corresponds to an optimisation performed in order to obtain a Pareto point. To attempt to avoid the double optimisation for each sub-problem the algorithm switches side making considerations over the side on which the previous Pareto points have been found. Although this reduces the number of optimisations to be performed, it has been seen that it is not effective when the method is applied to more complicated optimisation problems.

Finally, the rotation of the search domains, in order to obtain the Pareto points belonging to the *peripheral region*, would not be effective for cases in which the Pareto frontier tended to coincide with the *utopia plane*. In, fact, in the limit case in which the Pareto front coincided with the *utopia plane*, vector $\mathbf{l}$ would have to be rotated at $\pi/2$, i.e., for $\theta_r = \pi/2$ (36). Thus for all of the consecutive Pareto points sought, belonging to the *peripheral region*, the rotation would have to be the same. Hence for each such point, the reformulation of the optimisation problem and the search domain would be identical, making the rotation search strategy ineffective. Such effect is the reason for which, in Figure 39, the density of the Pareto points increases in the proximity of the edge having as vertices *anchor points* μ_1^* and μ_3^* .

3.5.2 The Double Hyper-Cone Boundary Intersection Method

To solve the limitations of the modified PP-based method the Double Hyper-cone Boundary Intersection (DHCBI) method has been developed. The PP-based method has been briefly described in a conference paper: Fantini et al. (2007). The DHCBI reproduces the modified PP-based method, but generating two opposite cones at once, for each sub-problem. Furthermore it enables to control the AOF. And last, but not least, the reformulation of the optimisation problem is much simpler.

The DHCBI method takes advantage of some of the concepts of the NBI method (9). In particular, when reformulating the optimisation problem for a *utopia plane point* $\mathbf{p}_k$, vector $\mathbf{F}(\mathbf{x})$ is decomposed in its two orthogonal components $\mathbf{t}$ and $\mathbf{n}$ with respect to a unit vector $\mathbf{l}$ orthogonal to the *utopia plane*.

As shown in Figure 40 the scalar values of the two components can be obtained as follows:

$$\begin{aligned}
 \gamma_i &= \arccos \left(\frac{\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k \cdot \mathbf{l}_i}{\|\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k\|} \right) \\
 \gamma_{\min} &= \gamma_{\min}(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) = \min\{\gamma_1, \gamma_2\} \\
 t(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) &= \|\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k\| \cos(\gamma_{\min}) \\
 n(\bar{\mathbf{F}}(\mathbf{x}), \mathbf{p}_k) &= \|\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k\| \sin(\gamma_{\min})
 \end{aligned} \tag{37}$$

where $\mathbf{l}_1$ coincides with the unit vector parallel to $\mathbf{l}$, pointing in the direction for which the objective functions increase, $\mathbf{l}_2 = -\mathbf{l}_1$ and where $\gamma_{\min}$ is the projection angle of $\|\bar{\mathbf{F}}(\mathbf{x}) - \mathbf{p}_k\|$ on $\mathbf{l}$.

This decomposition allows us to reformulate the optimisation problem for *utopia plane point* $\mathbf{p}_k$ as follows:

$$\begin{aligned} & \min_{\mathbf{x}} [f_{DHCBI}(t(\mathbf{x}), n(\mathbf{x}))] \\ & \text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}) \leq 0, \quad k = 1, 2, \dots, K \\ & \text{and } P \text{ equality constraints: } h_p(\mathbf{x}) = 0, \quad p = 1, 2, \dots, P \\ & \text{and subject to the additional constraint: } n \leq c(t) \end{aligned} \quad (38)$$

where function $f_{DHCBI}(t, n) = t + q(n)$, $t = \frac{\mathbf{t}}{\|\mathbf{t}\|}$, $n = \frac{\mathbf{n}}{\|\mathbf{n}\|}$ and $q(n)$ is a monotonically increasing function of n .

In order to obtain a domain similar to the one obtained with the modified PP-based method, we build a constraint which generates a domain as the one depicted in Figure 40b.

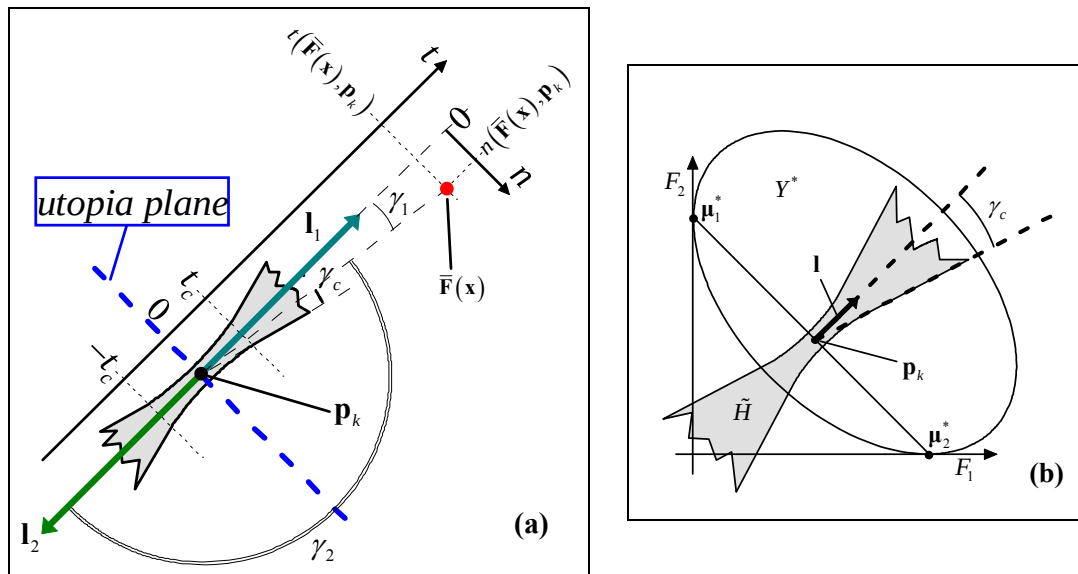


Figure 40: Detail of the DHCBI formulation (a) detail of the variables and parameters (b) constraints shape.

Through the additional constraint $n \leq c(t)$ in (38) we impose that the solution will have to be within a certain distance with respect to the line orthogonal to the *utopia plane* and passing through *utopia plane point* $\mathbf{p}_k$. Function $c(t)$ is chosen in order to generate two symmetric cones having cone angle γ_c . To avoid introducing discontinuities in the constraint, the cones are joined by means of a “smoothing” function for $t \leq t_c$.

One of the possible formulations for generating the double hyper-cone is the following:

$$c(t) = \begin{cases} t_c \frac{e^{\frac{\ln(\tan \gamma_c + 1)t}{t_c}} - 1}{\ln(\tan \gamma_c + 1)} - t + n_c & t \leq t_c \\ (t - t_c) \tan \gamma_c + t_c \left(\frac{\tan \gamma_c}{\ln(\tan \gamma_c + 1)} - 1 \right) + n_c & t > t_c \end{cases} \quad (39)$$

where $n_c = c(0)$ and t_c is the distance from $\mathbf{p}_k$ along $\pm \mathbf{l}$ at which the constraint becomes a hyper-cone. The formulation of the additional constraint (39), allows to maintain the continuity of the first derivative for $t = 0$ and $t = t_c$ as can be easily seen:

$$c'(t) = \begin{cases} e^{\frac{\ln(\tan \gamma_c + 1)t}{t_c}} - 1 & t \leq t_c \\ \tan \gamma_c & t > t_c \end{cases} \quad (40)$$

n_c , which is the minimum width of the search domain, can be chosen as a fraction of the Euclidean distance between two contiguous *utopia plane points*, in order to avoid the inclusion of other *utopia plane points* in the search domain.

The formulation for the additional constraint in (39) can be obtained as follows.

To obtain a domain as the one shown in Figure 40b the four following conditions have to be satisfied:

$$\begin{aligned} c(0) &= n_c \\ c'(0) &= 0 \\ c'(t) &= \tan \gamma_c, \quad t \geq t_c \end{aligned} \quad (41)$$

For $t \leq t_c$ we chose as first derivative for function $c(t)$

$$c'(t) = e^{\frac{A t}{t_c}} + B, \quad (42)$$

since in such way function $c(t)$ will be a monotone increasing function. Constants A and B can be determined as follows:

$$\begin{cases} c'(0) = 1 + B = 0 \\ c'(t_c) = e^A + B = \tan \gamma_c \end{cases} \Rightarrow \begin{cases} B = -1 \\ A = \ln(\tan \gamma_c + 1) \end{cases} . \quad (43)$$

$c(t)$ can then be obtained through integration as follows:

$$c(t) = \int c'(t) dt = \int (e^{\frac{A}{t_c} t} - 1) dt = \frac{t_c}{A} e^{\frac{A}{t_c} t} - t + C + D \quad (44)$$

where C and D are two constants to be determined.

From the first conditions of (41) and from (44)

$$c(0) = \frac{t_c}{A} + C + D = n_c , \quad (45)$$

which can be satisfied for $C = -\frac{t_c}{A}$ and $D = n_c$. Thus for $t \leq t_c$

$$c(t) = t_c \frac{e^{\frac{\ln(\tan \gamma_c + 1) t}{t_c}} - 1}{\ln(\tan \gamma_c + 1)} - t + n_c . \quad (46)$$

As a consequence, the value of function $c(t)$ for $t = t_c$ will be:

$$c(t_c) = \frac{t_c}{A} (\tan \gamma_c + 1) - t_c + C + D = \frac{t_c}{A} \tan \gamma_c - t_c + n_c . \quad (47)$$

For $t > t_c$ the condition to be satisfied from (41) is the following:

$$c'(t) = \tan \gamma_c . \quad (48)$$

This condition allows the search domain to become conical for $t > t_c$.

Function $c(t)$ for $t > t_c$ can be obtained as follows:

$$c(t) = \int c'(t) dt = t \tan \gamma_c + E , \quad (49)$$

where E is a constant to be determined. For maintaining the continuity of $c(t)$ in $t = t_c$ from (47) and (49) we impose

$$c(t_c) = t_c \tan \gamma_c + E = \frac{t_c}{A} \tan \gamma_c - t_c + n_c , \quad (50)$$

thus

$$E = \frac{t_c}{A} \tan \gamma_c - t_c + n_c - t_c \tan \gamma_c . \quad (51)$$

We then obtain that for $t > t_c$, $c(t)$ corresponds to the following linear function:

$$c(t) = (t - t_c) \tan \gamma_c + t_c \left(\frac{\tan \gamma_c}{\ln(\tan \gamma_c + 1)} - 1 \right) + n_c . \quad (52)$$

By choosing function $q(n)$ appropriately it is possible to orient the gradient of function $f_{DHCBI}(t, n) = t + q(n)$, in order to direct the solver towards the axis of the hyper-cone while minimising f_{DHCBI} along the axis of the cone.

One possible formulation for function q is the following:

$$q(n) = \begin{cases} \frac{n^{c+1}}{n_c^c}, & n \leq n_c \\ (1-c)n + \frac{cn^2}{n_c}, & n > n_c \end{cases}. \quad (53)$$

By computing the components of the gradients for $f_{DHCBI}(t, n) = t + q(n)$:

$$\nabla f_{DHCBI}(t, n) = \begin{cases} \begin{bmatrix} 1 \\ (c+1)\frac{n^c}{n_c^c} \end{bmatrix}, & n \leq n_c \\ \begin{bmatrix} 1 \\ (1-c) + 2\frac{cn}{n_c} \end{bmatrix}, & n > n_c \end{cases} \quad (54)$$

it can be shown that, on the axis of the cone ($n=0$) the gradient will be $\nabla f_{DHCBI}(t, 0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and will be parallel to the axis of the cone. For $0 \leq n \leq n_c$ the component of the gradient orthogonal to the axis of the cone $\frac{\partial f_{DHCBI}}{\partial t}$ will increase proportionally to n^c , while for $n \geq n_c$ it will increase linearly with respect to n .

The formulation for function $f_{DHCBI}(t, n)$ can be obtained as follows:

The following conditions are to be satisfied:

$$\begin{aligned} \nabla f_{DHCBI}(t, 0) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \nabla f_{DHCBI}(t, n_c) &= \begin{bmatrix} 1 \\ (c+1) \end{bmatrix}, & n = n_c \\ \nabla f_{DHCBI}(t, n) &= \begin{bmatrix} 1 \\ linear \end{bmatrix}, & n > n_c \end{aligned} \quad (55)$$

This corresponds to requiring the gradient to be parallel to the axis of the search domain (where $n=0$) while, moving outwards with respect to the axis, the component of the gradient orthogonal to the axis increases progressively from 0 to

$c+1$ and then linearly from thereafter. The desired behaviour of the gradient is shown in Figure 41. Parameter c allows controlling the rotation of the gradient.

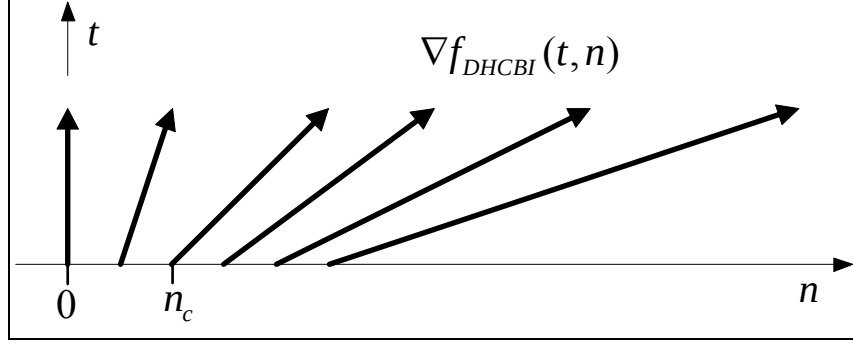


Figure 41: DHCBI gradient variation for with n

Since $f_{DHCBI}(t, n) = t + q(n)$ then the conditions can be reformulated as follows:

$$\begin{aligned} q'(0) &= 0 \\ q'(n_c) &= c+1 \\ q'(n) &= \text{linear}, \quad n > n_c \end{aligned} \tag{56}$$

by choosing

$$q'(n) = (c+1) \frac{n^c}{n_c^c}, \tag{57}$$

the second condition of (56) is satisfied. Then

$$q(n) = \int q'(n) dn = \frac{n^{c+1}}{n_c^c} \tag{58}$$

and

$$q(n_c) = n_c. \tag{59}$$

For $n > n_c$ continuity has to be satisfied. If we consider the following linear function:

$$q'(n) = A + Bn, \tag{60}$$

as required by the third condition of (56), where A and B are coefficients to be determined, then

$$q(n) = \int q'(n) dn = An + \frac{1}{2} Bn^2. \tag{61}$$

By imposing the second condition of (56) and continuity of $q(n)$ with (60) and (61), we obtain that:

$$\begin{cases} q'(n_c) = c+1 \\ q(n_c) = n_c \end{cases} \Rightarrow \begin{cases} A + Bn_c = c+1 \\ An_c + \frac{1}{2}Bn_c^2 = n_c \end{cases} \Rightarrow \begin{cases} A = 1-c \\ B = \frac{2c}{n_c} \end{cases}, \quad (62)$$

therefore

$$q(n) = (1-c)n + \frac{cn^2}{n_c}. \quad (63)$$

Figure 42 and Figure 43 show the results obtained with the implemented DHBI method for test case CRC_cvx and CRC_ccv, respectively. The figures show all of the search domains and corresponding contour lines, each of which allowed obtaining a Pareto point for each *utopia plane point*. As can be seen the Pareto points are well-distributed along the Pareto frontier.

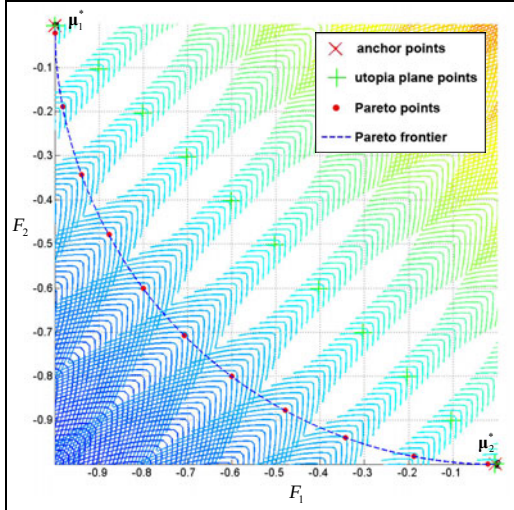


Figure 42: CRC_cvx solution obtained with the modified PP-based method

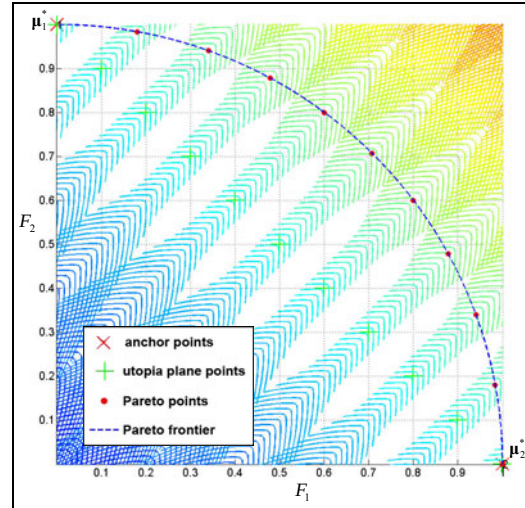


Figure 43: CRC_ccv solution obtained with the modified PP-based method

The advantages of the DHCBI method over the modified PP-based method can be summarised as follows:

- 1) Minimising of the number of optimisations to be performed.

Only one optimisation is performed for each of the *utopia plane points*.

- 2) Flexible definition of the search domain.

The search domain can be controlled through parameters n_c , γ_c and c . By modifying these parameters it is also possible to obtain search domains similar to

the ones of the NBI, PP-based (with no offsetting) and the modified PP-based method.

- 3) Better control over the AOF through the definition of function q .

Function q can be chosen in such a way to allow controlling the shape of the contour lines.

3.5.3 The NC+ method

Even though the DHCBI is capable of solving the problems encountered when performing multiobjective optimisation, analysing the results obtained for three test cases, it is evident that the NC method has still one advantage over the DHCBI method. The results for the three test cases are shown in Table 2. The test cases considered were the two 2 objective optimisation problem CRC_cvx and CRC_ccv presented in section 3.2.1 and the 3 objective SPHR_ccv-Case1 test case reported in section 3.1. For all three test cases the *utopia plane points* considered for comparing the two methods were the same, thus the number of analyses performed is only dependent on the formulation of the optimisation problem. For both CRC_cvx and CRC_ccv the number of *utopia plane points* was 11. For the SPHR_ccv-Case1 the *utopia plane points* considered were 28 and the anchor points positions were as defined by Case1 in section 3.1. The NBI method was applied in order to determine the *utopia plane points*, thus for test case SPHR_ccv-Case1 the *peripheral region problem* was not considered. This was done, once again in order to focus the attention only on the number of analyses required for obtaining the Pareto points. In all three cases the Pareto points obtained with the NC method and the DHCBI method were practically identical, the only difference being the number of iterations required for obtaining them. The parameters chosen for the DHCBI method were $c = 0.05$, $\gamma_c = 10^\circ$, $n_c = d_c/4$ and t_c was conveniently chosen as $t_c = n_c/\tan(\gamma_c)$.

As shown in Table 2 the NC method requires in all cases fewer analyses than the NC method for obtaining all Pareto points. It is speculated that this is due to the fact that while the formulation of the DHCBI method requires a complex combination of the objectives, the formulation of the NC method requires at most a linear combination of the objectives in order to compute the additional constraints.

Furthermore, the DHCBI method requires a number of parameters for reformulating the optimisation problem, while the NC method does not require any.

	NC Number of analyses	DHCBI Number of analyses
CRC_cvx	202	949
CRC_ccv	137	375
SPHR_ccv-Case1	420	1502

Table 2: Comparison of the number of analyses performed by the NC and DHCBI method

In section 3.2.1 a number of limitations of the NC method were identified. Firstly, it was demonstrated that particular positioning of the *anchor points* can bring the method to fail in obtaining well-distributed Pareto points. Furthermore, the method allows the existence of local minima for which the solution does not belong to a line orthogonal to the *utopia plane* and passing through the *utopia plane point* considered, this potentially jeopardising the capability of the method of generating well-distributed Pareto points.

By evaluating the limitations encountered by the NC method it was possible to develop the NC+ method, which is an improvement over the NC method. The formulation of the method is the following:

$$\begin{aligned}
& \min_{\mathbf{x}} F_l(\mathbf{x}) \\
& \text{subject to } K \text{ inequality constraints:} & g_k(\mathbf{x}) \leq 0, & k = 1, 2, \dots, K \\
& P \text{ equality constraints:} & h_p(\mathbf{x}) = 0, & p = 1, 2, \dots, P \\
& \text{subject to the additional } M - 1 \text{ constraints:} & \mathbf{v}_j \cdot (\mathbf{p}_i - \mathbf{F}) \leq 0 & \forall j \in \{1, 2, \dots, M\}, j \neq l \\
& \text{and subject to the additional constraints:} & \mathbf{v}_l \cdot \left(\mathbf{F} - \mathbf{p}_i - n_c \frac{\mathbf{v}_l}{\|\mathbf{v}_l\|} \right) \leq 0
\end{aligned} \tag{64}$$

where $\mathbf{v}_j = \frac{l_j}{l_i} \mathbf{e}_i - \mathbf{e}_j$ for $j \neq i$, $\mathbf{v}_i = \frac{\mathbf{m} \cdot \mathbf{l}}{l_i} \mathbf{e}_i - \mathbf{m}$, $\mathbf{e}_j \ \forall j$ are the base vectors of the coordinate system, $\mathbf{l}$ is the unit vector orthogonal to the *utopia plane*, n_c is a fraction of the Euclidean distance between two contiguous *utopia plane points* and finally $\mathbf{m} \in \mathbb{R}^M$ is a vector such that $m_i = 1 \ \forall i$ and $m_j = 0$.

The formulation is similar to the NC method, where $M - 1$ constraints are used for building $M - 1$ hyper-planes which confine the solution to a region of the criterion space. Although in the NC method these constraints are dependent on the *anchor points*, in the NC+ method they are build with respect to the coordinate system.

For the solution to belong to the line orthogonal to the *utopia plane*, passing through a particular *utopia plane point*, all hyper-planes need to be orthogonal to the *utopia plane* and have to intersect in the *utopia plane point*. In the NC method, for a *utopia plane point* $\mathbf{p}_i$, the orthogonality of the hyper-planes is obtained by defining vectors $\mathbf{q}_j$ (11). Since the *anchor points* are used for determining vectors $\mathbf{q}_j$, the vectors are automatically parallel to the *utopia plane*, thus the hyper-planes are all orthogonal to the *utopia plane*. In the NC+ method instead, the hyper-planes are determined with respect to the coordinate system thus the orthogonality condition has to be enforced. This can be simply obtained as follows.

If objective function F_i is to be minimised then we define the vector orthogonal to hyper-plane $j \neq i$ as:

$$\mathbf{v}_j = t_j \mathbf{e}_i - \mathbf{e}_j, \ \forall j \neq i \quad (65)$$

as also shown in Figure 44a for which F_3 is minimised. Then the orthogonality condition can be simply imposed as:

$$\mathbf{v}_j \cdot \mathbf{l} = (t_j \mathbf{e}_i - \mathbf{e}_j) \cdot \mathbf{l} = 0, \quad (66)$$

thus

$$t_j = \frac{l_j}{l_i}. \quad (67)$$

As a result from (65) and (67) the vector orthogonal to the hyper-plane for constraint j , $j \neq i$ is

$$\mathbf{v}_j = \frac{l_j}{l_i} \mathbf{e}_i - \mathbf{e}_j, \ \forall j \neq i \quad (68)$$

Figure 44b shows a diagram depicting the feasible region defined through constraints $\mathbf{v}_j$ for $j \neq l$. The diagram also shows the projection of the search direction on the utopia plane (15). As a result constraint j for $j \neq l$ is the following:

$$\mathbf{v}_j \cdot (\mathbf{p}_i - \mathbf{F}) \leq 0, \quad \forall j \neq l \quad (69)$$

By using the coordinate system for defining the constraints, no two hyper-planes are allowed to be parallel; this can be demonstrated by the fact that the angle between the two vectors is always greater than 0, which corresponds to demonstrating that scalar product $\mathbf{v}_j \cdot \mathbf{v}_p$ is always smaller $\|\mathbf{v}_j\| \|\mathbf{v}_p\|$. In fact, since

$$\mathbf{v}_j \cdot \mathbf{v}_p = \left(\frac{l_j}{l_l} \mathbf{e}_l - \mathbf{e}_j \right) \cdot \left(\frac{l_p}{l_l} \mathbf{e}_l - \mathbf{e}_p \right) = \frac{l_j l_p}{l_l^2}, \quad \forall j, p \neq l \quad (70)$$

and

$$\|\mathbf{v}_j\| \|\mathbf{v}_p\| = \left\| \frac{l_j}{l_l} \mathbf{e}_l - \mathbf{e}_j \right\| \left\| \frac{l_p}{l_l} \mathbf{e}_l - \mathbf{e}_p \right\| = \sqrt{\frac{l_j^2 l_p^2}{l_l^4} + \frac{l_j^2 + l_p^2}{l_l^2} + 1}, \quad \forall j, p \neq l \quad (71)$$

then verifying

$$\mathbf{v}_j \cdot \mathbf{v}_p < \|\mathbf{v}_j\| \|\mathbf{v}_p\|, \quad \forall j, p \neq l \quad (72)$$

we obtain that

$$\frac{l_j^2 l_p^2}{l_l^4} < \frac{l_j^2 l_p^2}{l_l^4} + \frac{l_j^2 + l_p^2}{l_l^2} + 1, \quad \forall j, p \neq l \quad (73)$$

which is always true.

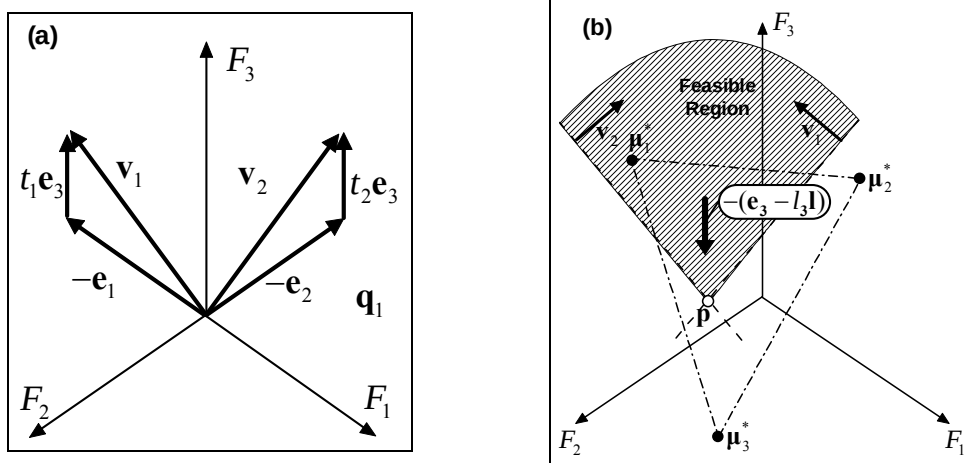


Figure 44: Formulation of the first $M-1$ additional constraints for the NC+ method (a) hyper-plane orthogonal vectors determination, (b) feasible region for utopia plane point $\mathbf{p}$ (viewed orthogonally with respect to the utopia plane).

To minimise the possibility of obtaining solutions not belonging to the line orthogonal to the *utopia plane*, passing through *utopia plane point* $\mathbf{p}_i$ we add an additional constraint. Constraint l is built in order to reduce the size of the feasible region, confining the solution in the proximity of the line passing through *utopia plane point* $\mathbf{p}_i$. As for the other $M - 1$ constraints, constraint l defines a hyper-plane orthogonal to the utopia plane, but positioned at a distance n_c in the direction of vector $\mathbf{v}_l$. Vector $\mathbf{v}_l$ is obtained as:

$$\mathbf{v}_l = t_l \mathbf{e}_l - \mathbf{m} \quad (74)$$

where $\mathbf{m} \in \mathbb{R}^M$ is a vector such that $m_i = 1 \ \forall i$ and $m_l = 0$, as shown in Figure 45a.

Once again the orthogonality condition is imposed as:

$$\mathbf{v}_l \cdot \mathbf{l} = (t_l \mathbf{e}_l - \mathbf{m}) \cdot \mathbf{l} = 0, \quad (75)$$

thus

$$t_l = \frac{\mathbf{m} \cdot \mathbf{l}}{l_l} \quad (76)$$

and as a result, from (74) and (76), we obtain:

$$\mathbf{v}_l = \frac{\mathbf{m} \cdot \mathbf{l}}{l_l} \mathbf{e}_l - \mathbf{m}. \quad (77)$$

As among vectors $\mathbf{v}_j$ for $j \neq l$, it can be demonstrated that constraint $\mathbf{v}_l$ is not parallel to any constraint $\mathbf{v}_j$ by the fact that their scalar product $\mathbf{v}_l \cdot \mathbf{v}_j$ is always smaller $\|\mathbf{v}_l\| \|\mathbf{v}_j\|$, in fact

$$\mathbf{v}_l \cdot \mathbf{v}_j = \left(\frac{\mathbf{m} \cdot \mathbf{l}}{l_l} \mathbf{e}_l - \mathbf{m} \right) \cdot \left(\frac{l_j}{l_l} \mathbf{e}_l - \mathbf{e}_j \right) = \frac{l_j}{l_l^2} \sum_{i=1}^M l_i + 1, \quad \forall j \neq l \quad (78)$$

and

$$\|\mathbf{v}_l\| \|\mathbf{v}_j\| = \sqrt{\frac{l_j^2}{l_l^4} \left(\sum_{i=1, i \neq l}^M l_i \right)^2 + (M-1) \frac{l_j^2}{l_l^2} + \frac{1}{l_l^2} \left(\sum_{i=1, i \neq l}^M l_i \right)^2 + M-1}, \quad \forall j \neq l \quad (79)$$

then verifying

$$\mathbf{v}_l \cdot \mathbf{v}_j < \|\mathbf{v}_l\| \|\mathbf{v}_j\|, \quad \forall j \neq l \quad (80)$$

we obtain that

$$(M-2)\frac{l_j^2}{l_l^2} + \left(\sum_{\substack{i=1 \\ i \neq l, j}}^M l_i \right)^2 + M-2 > 0, \quad \forall j \neq l \quad (81)$$

which is always satisfied for a number of objectives M greater than 2.

Figure 45b depicts all the three constraints for a three objective optimisation problem.

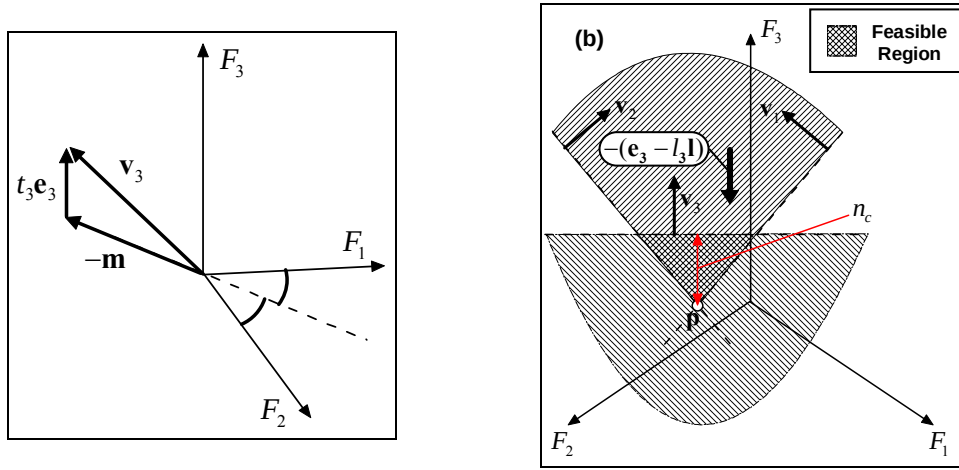


Figure 45: Formulation of additional constraints for the NC+ method (a) hyper-plane l orthogonal vectors determination, (b) feasible region for utopia plane point $\mathbf{p}$ (viewed orthogonally with respect to the utopia plane).

The positioning of the constraint at a distance n_c is simply obtained by adding $n_c \frac{\mathbf{v}_l}{\|\mathbf{v}_l\|}$ to utopia plane point $\mathbf{p}_i$ for constraint l , as also shown in Figure 45b. Thus constraint l is

$$\mathbf{v}_l \cdot \left(\mathbf{F} - \mathbf{p}_i - n_c \frac{\mathbf{v}_l}{\|\mathbf{v}_l\|} \right) \leq 0. \quad (82)$$

As for the DHCBI method, n_c can be chosen as a fraction of the Euclidean distance between two contiguous utopia plane points

The advantages of the NC+ method with respect to the DHCBI and modified PP-based method are the following:

- 1) Minimises the number of optimisations to be performed.

Only one optimisation is performed for each of the utopia plane points.

- 2) Flexible definition of the search domain with minimum number of parameters.

The search domain can be controlled through parameter n_c . Parameter n_c allows to vary the size of the feasible region.

- 3) No need of controlling the AOF.

The AOF of the modified PP-based and DHCBI method is replaced with a single objective similarly to the NC method

The results of the comparison of the NC+ method with the other investigated methods are presented in chapter 6.

3.5.4 Obtaining a complete representation of the Pareto Front

For obtaining a complete representation of the Pareto frontier it is necessary to obtain also those Pareto points that belong to the *peripheral region*. Here we propose an alternative approach for obtaining such points, with respect to the one proposed by Messac and Mattson (2004). Since the points belonging to the *peripheral region* have orthogonal projection to the *utopia plane* external to the polygon of vertices μ_i^* , a possible solution is that of generating the *peripheral region utopia plane points* $\mathbf{p}^+$ using the *utopia plane points* belonging to the edges of the polygon.

Let us consider a *utopia plane point* $\mathbf{p}^*$ belonging to the k -th edge of the polygon. Then as for the modified PP-based method, following (34) and (35), we can determine vector $\mathbf{s}_i$ parallel to the *utopia plane* and pointing towards the exterior of the polygon having as vertices the *anchor points* (Figure 38a). For each of the *utopia plane points* $\mathbf{p}^*$ belonging to an edge, the *peripheral region utopia plane points* can be then obtained as:

$$\mathbf{p}^+ = \mathbf{p}^* + q n_d \mathbf{s}_i, \quad q = 1, 2, \dots \quad (83)$$

where n_d corresponds to the distance between two adjacent *utopia plane points*. Following this approach each new *peripheral region utopia plane point* $\mathbf{p}^+$ belongs to the orthogonal projection of the *peripheral region* on the *utopia plane* and is

generated from an initial *utopia plane point* $\mathbf{p}^*$, moving orthogonally to the edge of the polygon to which $\mathbf{p}^*$ belongs and parallel to the utopia plane. For obtaining all of the Pareto points belonging to the *peripheral region*, for each of the edge *utopia plane points* $\mathbf{p}^*$, q is increased until the optimiser fails to obtain a solution.

The advantage of this approach over the modified PP-based method is that the problem concerning the rotation of the anchor points is removed. Furthermore the approach is general and can also be applied to the NBI, PP-based and NC method.

3.5.5 Gradient Based Global Multiobjective Optimisation

The implemented methods make use of a Sequential Quadratic Programming (SQP) algorithm for solving the optimisation sub-problems. SQP belongs to the family of the gradient based methods. SQP, as other gradient based methods, is conventionally used for solving local optimisation problems. The reason for this is that at each iteration gradient based methods make use of the gradients relatively to a particular point, in order to determine the next one. Thus the information used for performing the optimisation is valid only with respect to the neighbourhood to which the gradients are obtained.

The fact that SQP is a local optimiser can be used as an advantage for searching the design space for points of interest for the DM. It must be remembered that the design points obtained should be evaluated *a posteriori* so that the DM can make additional considerations on the solutions found.

For solving the global optimisation problem one of the proposed methods is applied repeatedly in order to allow the identification of all local Pareto frontiers. Then, by evaluating the solutions obtained, the global Pareto set is easily determined. For doing this it is sufficient to run one of the proposed methods as many times, as the number of local minima obtained for each of the single objectives. These are computed in a pre-processing phase of the algorithm. This gives the proposed approach the unique feature of generating local Pareto fronts.

As discussed in section 3.3, we can define as local Pareto frontier, a Pareto frontier within a particular neighbourhood of the design space. The procedure presented bellow shows how such frontiers can be obtained. The procedure has two

main advantages, the first is that a more complete investigation of the design space is performed improving the chances of obtaining global Pareto solutions. Secondly, it allows the DM to obtain and analyse a set of solutions that can be of interest from an engineering point of view, for example, when accounting for additional considerations such as design robustness.

The procedure for obtaining the local Pareto frontiers is the following:

1. A Latin-hypercube sampling procedure is used for obtaining L well distributed points in the design space.
2. L single-objective optimisations are performed for each of the M objectives, yielding *local anchor points* $\boldsymbol{\mu}_{ij}^* \in \mathbb{R}^M$, $i \in \{1, \dots, M\}$, $j \in \{1, \dots, L_i^*\}$ and relative design points $\mathbf{x}_{ij}^* \in \mathbb{R}^N$. L_i^* is the number of local optima obtained for objective i . It is to be noted that L_i^* is considered to be different from L since not all optimisations might yield a solution.
3. In order to remove multiple coinciding *anchor points*, for each objective i it is checked that $\|\bar{\mathbf{x}}_{ij}^* - \bar{\mathbf{x}}_{ik}^*\|_{\infty} > \varepsilon$ for each $j, k \in \{1, \dots, L_i^*\}$, $j \neq k$. Where each component of the design point vectors is normalised with respect to its lower and upper bounds, i.e. $\bar{x}_{ijl}^* = (x_{ijl}^* - x_{Ul}) / (x_{Ll} - x_{Ul})$ for $l = 1, \dots, N$. If for two points $\|\bar{\mathbf{x}}_{ij}^* - \bar{\mathbf{x}}_{ik}^*\|_{\infty} \leq \varepsilon$ then the μ_{iji}^* and μ_{iki}^* are compared and the point with the greater value in case the objective is to be minimised, or the smaller value in case the point is to be maximised, is removed. The value of ε is a parameter and is to be chosen for $0 < \varepsilon \ll 1$.
4. For each objective i and *anchor point* j we determine a polygon with well spread *anchor points*. This is done by evaluating the Euclidean distance between $\bar{\boldsymbol{\mu}}_{ij}^*$ and each *anchor point* of each combination of the $M-1$ *anchor points* $\bar{\boldsymbol{\mu}}_{pk_p}^*$, where $p = 1, \dots, M$, $p \neq i$ and $k_p \in \{1, \dots, L_p^*\}$, and determining the combination for which the minimum distance between $\bar{\boldsymbol{\mu}}_{ij}^*$ and one of the *anchor points* $\bar{\boldsymbol{\mu}}_{pk_p}^*$ of the

permutation is maximum. This corresponds to determining the $M-1$ anchor points $\bar{\mu}_{pk_p}^* = \bar{\mu}_{pk_p}^*$ for which

$$\max_{k_1, \dots, k_p, \dots, k_M} \left(\min \left(\left\{ \left\| \bar{\mu}_{1,k_1}^* - \bar{\mu}_{2,k_2}^* \right\|, \dots, \left\| \bar{\mu}_{i-1,k_{i-1}}^* - \bar{\mu}_{i,j}^* \right\|, \left\| \bar{\mu}_{i,j}^* - \bar{\mu}_{i+1,k_{i+1}}^* \right\|, \dots \right\} \right) \right) \quad (84)$$

It must be noted that each component of the *anchor point* vectors is normalised with respect to its lower and upper bounds, thus $\bar{\mu}_{ijl}^* = \mu_{ijl}^* / (\mu_{Ll} - \mu_{Ul})$ for $l = 1, \dots, M$.

5. For objective i and local minimum j , *local anchor points* $\mu_{1k_1}^*, \dots, \mu_{ij}^*, \dots, \mu_{Mk_M}^*$ are used for defining the K *utopia plane points* as in (7):

$$\mathbf{p}_{ijk} = \sum_{\substack{p=1 \\ p \neq i}}^M \alpha_p \bar{\mu}_{pk_p}^* + \alpha_i \mu_{ij}^*, \quad k = \{1, 2, \dots, K\} \quad (85)$$

6. For *utopia plane point* $\mathbf{p}_{ijk}$ the optimisation problem is reformulated and solved.
7. Steps 4 through 6 are repeated L_i times for each of the M objectives.

Starting the sequence of optimisations from the *utopia plane point* closest to the *local anchor point* μ_{ij}^* and moving towards the other $M-1$ *global anchor points* allows obtaining the local Pareto in the neighbourhood of μ_{ij}^* .

The procedure presented is very simple; in fact, it coincides with reiterating the Pareto front procedure, each time selecting a set of different *anchor points*. It has been developed taking advantage from the considerations made when analysing the results obtained for global multiobjective optimisation test cases such as Deb1 (section 3.3). The results obtained by applying this procedure on a set of global multiobjective optimisation problems are presented in chapter 6.

3.6 Summary

By testing the NBI, PP-based and NC method on a number of test cases it was possible to identify several shortcomings of such methods. The tightness of the equality constraint introduced in the reformulation of NBI method can limit its effectiveness in determining each Pareto point, furthermore the *peripheral region*

problem is not considered. The application of the PP-based method requires the determination of several parameters which are initially unknown and the *peripheral region* problem is not considered. Finally, even though with the NC method an approach for dealing with the *peripheral region* problem is proposed, the NC method can fail in generating Pareto points when the number of objectives is greater than two; furthermore its reformulation allows the existence of local minima.

In the attempt of solving such limitations the modified PP-based method was initially developed, this then led to the development of the DHCBI and finally the NC+ method. Through the evaluation of the results obtained by applying the reformulation techniques investigated to global multiobjective optimisation problems, a procedure which allows obtaining local-Pareto fronts, as well as increasing the chances of obtaining the global Pareto front, is proposed. The developed NC+ method solves the limitations encountered by the NBI, PP-based, NC, modified PP-based and DHCBI method and is capable of solving global multiobjective continuous optimisation problems.

In the following chapter considerations on the integration of the reformulation methods with multidisciplinary optimisation algorithms, together with considerations on the application of automatic differentiation (AD), are discussed.

4 Multiobjective MDO and application of Automatic Differentiation

For simulating the behaviour of a complex system the designer makes use of a set of CAs, each of which models a particular discipline. According to the aims of the particular study to be conducted, input and output variables of the complete system are selected. Because of the mutual interaction of the CAs, arranging these in order to satisfy the requirements of the study may lead to a non-hierarchically decomposable system. In such a case the CAs are said to be coupled and can not be executed sequentially. For performing effectively and efficiently an optimisation when the system is non-hierarchically decomposable, a number of approaches exist; the main ones are briefly introduced in section 2.2.

4.1 Multiobjective MDO for conceptual design

Although all of the MDO algorithms would be suitable for application to conceptual design, a problem exists. The current approaches have been developed for preliminary design, where the number of constituent disciplines is usually, at most, in the order of tens of CAs. This is due to the fact that in the preliminary design phase high fidelity CAE tools are used for simulating the behaviour of the system. Therefore performing an optimisation of the complete system with all of its components, as done at the conceptual phase, becomes unthinkable. Thus, for solving this problem the aircraft is subdivided in its main components, for each of which a preliminary design is conducted. In contrast with preliminary design, at the conceptual design phase the complete aircraft is taken into account and the number of CAs is in the order of hundreds, each of which corresponding to simple equations. Thus even in the case of a hierarchically decomposable system, arranging the disciplines becomes a challenge (Balachandran, Fantini, and Guenov, 2007).

Therefore, although arranging a non-hierarchically decomposable system as would be required for the application of more complex MDO algorithms, such as CO, CSSO and BLISS (discussed in section 2.2), would be an interesting challenge, it is out of the scope of the current work. As a result the MDO algorithms considered here

are the simpler and more commonly applied ones: the Nested Analysis and Design (NAND) and the Simultaneous Analysis and Design (SAND). Anyway, this does not affect the general applicability of the approach.

The general optimisation problem for a non-hierarchically decomposable system is formulated in (3). The optimisation problems for the NAND and SAND approaches discussed in section 2.2, can be formulated as follows:

NAND:

As discussed in section 2.2.1, following the NAND approach the problem of the non-hierarchically decomposable system is completely hidden from the SLO. Thus, the multiobjective optimisation problem can be formulated as if the system was hierarchically decomposable. The formulation of the optimisation problem is identical to (1), which is also reported here:

$$\begin{aligned} & \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}, \mathbf{z}) \\ & \text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}, \mathbf{z}) \leq 0, \quad k = 1, 2, \dots, K \\ & \text{and } P \text{ equality constraints: } h_p(\mathbf{x}, \mathbf{z}) = 0, \quad p = 1, 2, \dots, P \end{aligned} \quad (86)$$

Each time function $\mathbf{F}(\mathbf{x}, \mathbf{z})$ is evaluated for a design point $\mathbf{x} = \bar{\mathbf{x}}$, the non-hierarchically decomposable system $\mathbf{z} = \mathbf{S}(\bar{\mathbf{x}}, \mathbf{z})$ has to be brought to convergence. As a result the values of $\mathbf{F}(\mathbf{x}, \mathbf{z})$ are always consistent, the only exception being the case for which for some $\mathbf{x} = \bar{\mathbf{x}}$ there does not exist any $\mathbf{z}$ satisfying $\mathbf{z} = \mathbf{S}(\bar{\mathbf{x}}, \mathbf{z})$, thus $\mathbf{F}(\bar{\mathbf{x}}, ?)$ can not be determined. This case is important from a practical point of view, since the optimiser will still require some value for $\mathbf{F}(\bar{\mathbf{x}}, ?)$. A possible solution to this problem is that of substituting $\mathbf{F}(\bar{\mathbf{x}}, ?)$ with a vector containing maximal values for $\mathbf{F}(\mathbf{x}, \mathbf{z})$, while substituting $\mathbf{g}(\mathbf{x}, \mathbf{z}) \leq \mathbf{0}$ and $\mathbf{h}(\mathbf{x}, \mathbf{z}) = \mathbf{0}$ with the maximal values of $\mathbf{g}(\mathbf{x}, \mathbf{z})$ and $\mathbf{h}(\mathbf{x}, \mathbf{z})$, respectively; this in order to discourage the optimiser in investigating such region of the design space.

SAND:

The formulation of the optimisation problem for the SAND approach is the following:

$$\begin{aligned}
& \min_{\mathbf{x}, \mathbf{z}} \mathbf{F}(\mathbf{x}, \mathbf{z}) \\
& \text{subject to } K \text{ inequality constraints: } g_k(\mathbf{x}, \mathbf{z}) \leq 0, \quad k = 1, 2, \dots, K \\
& \text{and } P \text{ equality constraints: } h_p(\mathbf{x}, \mathbf{z}) = 0, \quad p = 1, 2, \dots, P \\
& \text{and with the additional equality constraints: } \mathbf{z} = \mathbf{S}(\mathbf{x}, \mathbf{z}).
\end{aligned} \tag{87}$$

In contrast to the NAND approach, for the SAND approach (section 2.2.2) the optimisation problem is augmented with the equality constraints, requiring for the system to be satisfied. As also previously reported in section 2.2, the main problem of this approach is that consistency may not be enforced until an optimal solution is obtained.

From an optimisation point of view this can mislead the optimiser in minimising objective functions $\mathbf{F}$. This can be easily demonstrated by considering two design points: point $(\mathbf{x}, \mathbf{z}) = (\bar{\mathbf{x}}, \bar{\mathbf{z}})$ for which $\bar{\mathbf{z}} \neq \mathbf{S}(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ and point $(\mathbf{x}, \mathbf{z}) = (\bar{\mathbf{x}}, \tilde{\mathbf{z}})$ for which $\tilde{\mathbf{z}} = \mathbf{S}(\bar{\mathbf{x}}, \tilde{\mathbf{z}})$; in general this implies that $\mathbf{F}(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \neq \mathbf{F}(\bar{\mathbf{x}}, \tilde{\mathbf{z}})$. As a result, during the optimisation process, if $\mathbf{z} = \mathbf{S}(\mathbf{x}, \mathbf{z})$ is not satisfied, the information used at each step by the optimiser in order to minimise the objective functions will be based on inconsistent values of $\mathbf{F}$.

Once again, the fact that the *utopia plane points* are ordered by minimising the distance between each subsequent one and its predecessor, becomes useful when performing multiobjective multidisciplinary optimisation. As the solution obtained for a *utopia plane point* can be used as starting point for the optimisation relative to the subsequent point, so can the coupling variables $\mathbf{z}$. Whichever MDO algorithm is considered, this allows using as starting point for the optimiser a solution corresponding to a converged system. Furthermore if the solution sought is in the neighbourhood of the starting point, we can reasonably assume that the values of the coupling variables will not differ greatly from the initial ones.

Figure 46 and Figure 47 show the variable flow for the NAND and SAND approaches respectively, when optimising for *utopia plane point* $\mathbf{p}_i$. In the figures, $\mathbf{x}_i$ is the initial design point, while $\mathbf{z}_i$ is the initial value of the coupling variables; these values having been determined as solution obtained relatively to *utopia plane point* $\mathbf{p}_{i-1}$. At each iteration, the optimiser will call the system block for evaluating a particular design point and will retrieve its output $\mathbf{y}$. For the NAND approach (Figure 46) the flow of the coupling variables is represented by dotted lines since, according to its formulation (86), the coupling variables are not part of the independent variables for the optimiser. In contrast, for the SAND approach (Figure 47) both the design variables and coupling variables, according to (87), are independent variables for the optimiser.

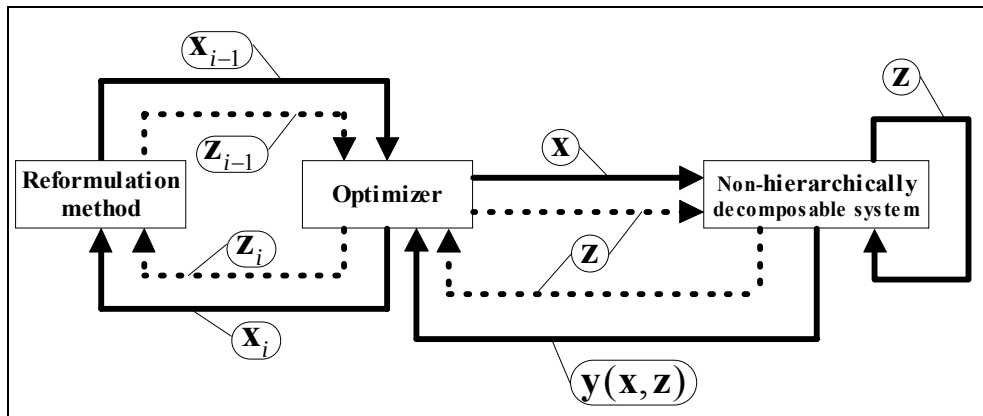


Figure 46: NAND variable flow

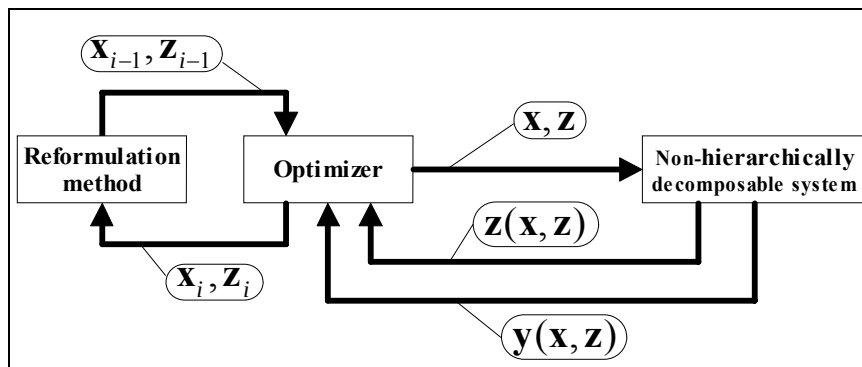


Figure 47: SAND variable flow

4.2 Automatic Differentiation and Global Sensitivity Equations

When performing multiobjective optimisation following the reformulation approach, a single objective optimisation has to be performed in order to obtain each Pareto point. According to the characteristics of conceptual design considered (section 2), for which all variables are continuous and the functions constituting the CAs are generally continuous, gradient based optimisation methods appear to be the most appropriate.

As previously stated in section 2.3.2, one of the problems encountered in applying gradient based methods, is the number of analyses required for obtaining the Jacobians. As stated in section 2.4, to solve this problem automatic differentiation (AD) can be applied.

When evaluating the solution of a system for a particular set of input variables and automatic differentiation (AD) is applied, the analytical partial derivatives of the output of the system with respect to the inputs are obtained as output of the system. The variable flow in the case of a non-hierarchically decomposable system is shown in Figure 48. In the figure the inputs to the system are $\mathbf{x} \in \mathbb{R}^N$, which is the vector of the independent variables, and $\mathbf{z}_I \in \mathbb{R}^R$, which is the vector of the coupling variables in input. The outputs to the system are: $\mathbf{y} = \mathbf{y}(\mathbf{x}, \mathbf{z}_I)$ ($\mathbf{y}: \mathbb{R}^{N+R} \rightarrow \mathbb{R}^M$) and $\mathbf{z}_O = \mathbf{z}_O(\mathbf{x}, \mathbf{z}_I)$ ($\mathbf{z}: \mathbb{R}^{N+R} \rightarrow \mathbb{R}^R$), which is the vector of the coupling variables in output. Furthermore, applying AD, the Jacobians $\mathbf{J}_y(\mathbf{x}, \mathbf{z}_I)$, which is the Jacobian of $\mathbf{y}$, and $\mathbf{J}_{z_o}(\mathbf{x}, \mathbf{z}_I)$ are obtained.

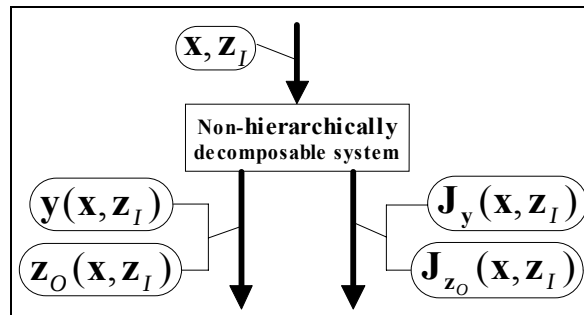


Figure 48: Variable flow with AD application.

When computing the Jacobian for a point $(\mathbf{x}, \mathbf{z}_I)$ the convergence of the non-hierarchically decomposable system $\mathbf{z}_O(\mathbf{x}, \mathbf{z}_I) = \mathbf{z}_I$ is not taken into account. Thus, in order to correct the Jacobian, global sensitivity equations GSE (Sobieszczanski-Sobieski, 1990) have to be applied. The derivation of the GSE, similarly to (Sobieszczanski-Sobieski, 1990) is reported as follows.

By considering a linearization of $\mathbf{y}$ and $\mathbf{z}$, according to the chain rule, we obtain that the partial derivative of component y_i with respect to x_j is:

$$\frac{dy_i}{dx_j} = \frac{\partial y_i}{\partial x_j} + \sum_{k=1}^R \frac{\partial y_i}{\partial z_{Ik}} \frac{dz_{Ik}}{dx_j}, \quad (88)$$

while the partial derivative of z_{Or} with respect to x_j is:

$$\frac{dz_{Or}}{dx_j} = \frac{\partial z_{Or}}{\partial x_j} + \sum_{k=1}^R \frac{\partial z_{Or}}{\partial z_{Ik}} \frac{dz_{Ik}}{dx_j}. \quad (89)$$

Which can also be written as:

$$\frac{dz_{Or}}{dx_j} = \frac{\partial z_{Or}}{\partial x_j} + \frac{\partial z_{Or}}{\partial z_{Ir}} \frac{dz_{Ir}}{dx_j} + \sum_{\substack{k=1 \\ k \neq r}}^R \frac{\partial z_{Or}}{\partial z_{Ik}} \frac{dz_{Ik}}{dx_j}. \quad (90)$$

All values $\frac{\partial y_i}{\partial x_j}$, $\frac{\partial y_i}{\partial z_{Ir}}$, $\frac{\partial z_{Or}}{\partial x_j}$ and $\frac{\partial z_{Or}}{\partial z_{Ir}}$, for $i = 1, \dots, M$, $j = 1, \dots, N$ and $r = 1, \dots, R$

are readily available from the Jacobian computed by means of AD.

If the infinitesimal variation of each coupling variable z_{Ir} ($r = 1, \dots, R$) is such that the consistency of the system is maintained then we have that

$$\frac{dz_{Or}}{dx_j} = \frac{dz_{Ir}}{dx_j} \Rightarrow dz_{Or} = dz_{Ir} = dz_r. \quad (91)$$

As a result (90) can also be written as:

$$\left(1 - \frac{\partial z_{Or}}{\partial z_{Ir}}\right) \frac{dz_{Or}}{dx_j} = \frac{\partial z_{Or}}{\partial x_j} + \sum_{\substack{k=1 \\ k \neq r}}^R \frac{\partial z_{Or}}{\partial z_{Ik}} \frac{dz_{Ik}}{dx_j}. \quad (92)$$

From (88) and (92) we then obtain the systems of $N(R+M)$ equations and

$N(R+M)$ unknowns $\frac{dz_r}{dx_j}$, for $r = 1, \dots, R$, $j = 1, \dots, N$ and $\frac{dy_i}{dx_j}$, for $i = 1, \dots, M$,

$j = 1, \dots, N$:

$$\begin{cases} \left(1 - \frac{\partial z_{Or}}{\partial z_{Ir}}\right) \frac{dz_r}{dx_j} - \sum_{\substack{k=1 \\ k \neq r}}^R \frac{\partial z_{Or}}{\partial z_{Ik}} \frac{dz_k}{dx_j} = \frac{\partial z_{Or}}{\partial x_j}, & r = 1, \dots, R, j = 1, \dots, N \\ \frac{dy_i}{dx_j} - \sum_{k=1}^R \frac{\partial y_i}{\partial z_{Ik}} \frac{dz_k}{dx_j} = \frac{\partial y_i}{\partial x_j}, & i = 1, \dots, M, j = 1, \dots, N \end{cases} \quad (93)$$

Or equivalently N independent systems, where system j of $R+M$ equations and $R+M$ unknowns $\frac{dz_r}{dx_j}$, for $r = 1, \dots, R$ and $\frac{dy_i}{dx_j}$, for $i = 1, \dots, M$ is:

$$\begin{cases} \left(1 - \frac{\partial z_{Or}}{\partial z_{Ir}}\right) \frac{dz_r}{dx_j} - \sum_{\substack{k=1 \\ k \neq r}}^R \frac{\partial z_{Or}}{\partial z_{Ik}} \frac{dz_k}{dx_j} = \frac{\partial z_{Or}}{\partial x_j}, & r = 1, \dots, R \\ \frac{dy_i}{dx_j} - \sum_{k=1}^R \frac{\partial y_i}{\partial z_{Ik}} \frac{dz_k}{dx_j} = \frac{\partial y_i}{\partial x_j}, & i = 1, \dots, M \end{cases} \quad (94)$$

System (94) can be rewritten using matrices as follows:

$$\begin{bmatrix} \mathbf{A} & \mathbf{Z} \\ \mathbf{B} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{z}}{dx_j} \\ \frac{d\mathbf{y}}{dx_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{z}}{\partial x_j} \\ \frac{\partial \mathbf{y}}{\partial x_j} \end{bmatrix} \quad (95)$$

where $\mathbf{A}$ is the $R \times R$ matrix

$$\mathbf{A} = \begin{bmatrix} \left(1 - \frac{\partial z_{O1}}{\partial z_{I1}}\right) & -\frac{\partial z_{O1}}{\partial z_{I2}} & \dots & -\frac{\partial z_{O1}}{\partial z_{IR}} \\ -\frac{\partial z_{O2}}{\partial z_{I1}} & \left(1 - \frac{\partial z_{O2}}{\partial z_{I2}}\right) & & \vdots \\ \vdots & & \ddots & \vdots \\ -\frac{\partial z_{OR}}{\partial z_{I1}} & \dots & \dots & \left(1 - \frac{\partial z_{OR}}{\partial z_{IR}}\right) \end{bmatrix}, \quad (96)$$

$\mathbf{B}$ is the $M \times R$ matrix

$$\mathbf{B} = \begin{bmatrix} -\frac{\partial y_1}{\partial z_{I1}} & -\frac{\partial y_1}{\partial z_{I2}} & \dots & -\frac{\partial y_1}{\partial z_{IR}} \\ -\frac{\partial y_2}{\partial z_{I1}} & -\frac{\partial y_2}{\partial z_{I2}} & & -\frac{\partial y_2}{\partial z_{IR}} \\ \vdots & & \ddots & \vdots \\ -\frac{\partial y_M}{\partial z_{I1}} & -\frac{\partial y_M}{\partial z_{I2}} & \dots & -\frac{\partial y_M}{\partial z_{IR}} \end{bmatrix}, \quad (97)$$

$\mathbf{Z}$ is an $R \times M$ zero matrix and $\mathbf{I}$ is an $M \times M$ identity matrix.

From (95) we can then obtain that for a converged system, if matrix $\begin{bmatrix} \mathbf{A} & \mathbf{Z} \\ \mathbf{B} & \mathbf{I} \end{bmatrix}$ is not singular

$$\begin{bmatrix} \frac{d\mathbf{z}}{dx_j} \\ \frac{d\mathbf{y}}{dx_j} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{Z} \\ \mathbf{B} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathbf{z}}{\partial x_j} \\ \frac{\partial \mathbf{y}}{\partial x_j} \end{bmatrix}, \quad (98)$$

thus the corrected Jacobians for converged system are:

$$\begin{aligned} \bar{\mathbf{J}}_y(\mathbf{x}, \mathbf{z}) &= \left[\frac{d\mathbf{y}}{dx_1}, \dots, \frac{d\mathbf{y}}{dx_N}, \frac{d\mathbf{y}}{dz_1}, \dots, \frac{d\mathbf{y}}{dz_R} \right] \text{ and} \\ \bar{\mathbf{J}}_{z_o}(\mathbf{x}, \mathbf{z}) &= \left[\frac{dz_o}{dx_1}, \dots, \frac{dz_o}{dx_N}, \frac{dz_o}{dz_1}, \dots, \frac{dz_o}{dz_R} \right] \end{aligned} \quad (99)$$

Where the components of y_i with respect to z_j can be computed as

$$\frac{dy_i}{dz_j} = \frac{\partial y_i}{\partial z_{Ij}} + \sum_{k=1}^N \frac{\partial y_i}{\partial x_k} \frac{dx_k}{dz_j}, \quad (100)$$

while the partial derivative of z_{or} with respect to z_j are

$$\frac{dz_i}{dz_j} = \frac{\partial z_{oi}}{\partial z_{Ij}} + \sum_{k=1}^N \frac{\partial z_{oi}}{\partial x_k} \frac{dx_k}{dz_j}. \quad (101)$$

Since it is obviously the gradients which allow a gradient based optimiser to obtain a solution, it is paramount to apply GSE. This is necessary in order to provide the optimiser with the correct partial derivatives of all the outputs with respect to all of the inputs.

From an implementation point of view it must be considered that, although AD is effective in reducing the computational cost of obtaining Jacobians, its application is effective only if the functions are continuous in the point $(\mathbf{x}, \mathbf{z}_I)$ where the Jacobians are computed. Thus, in the context of an optimisation process the values of the Jacobians need to be checked, in order to avoid the failure of the optimisation process. Although finite difference are affected by the same problem, they are more robust in that a value can still be computed, even if not correct, avoiding the complete failure of the optimisation process. As a result the two approaches can be used complementarily, applying finite differences when AD fails. Finally, when computing

the correct Jacobians by means of GSE it is important for the matrix $\begin{bmatrix} \mathbf{A} & \mathbf{Z} \\ \mathbf{B} & \mathbf{I} \end{bmatrix}$ to be invertible. When this is not the case, then GSE can not be applied. Thus, the Jacobians needed by the optimiser will have to be determined in a different way, for example through finite differences.

4.3 Summary

In this chapter the methodology for applying multiobjective optimisation algorithms to multidisciplinary design problems was presented. Automatic differentiation is considered for determining the partial derivatives at a low computational cost, in order to enhance the performance of gradient based optimisers. The problem of determining the correct Jacobians in case of a converged system is presented together with its formulation.

In the following chapter the complete procedure developed for determining well-distributed Pareto points, relative to local and global Pareto frontiers, for a non-hierarchically decomposable system is presented. A brief description of the software implementation of the algorithm used for obtaining the results presented is also given.

5 THE CODE-M algorithm

The NC+ method, the global optimisation procedure, the NAND and SAND algorithms, automatic differentiation, and the GSE equations are integrated in the CODE-M algorithm, an algorithm for Conceptual Design global Multiobjective Multidisciplinary optimisation. In the following section a description of the complete algorithm is given, followed by some details of its implementation.

5.1 The algorithm

The flowchart of the CODE-M algorithm for determining well-distributed Pareto points on the local and global Pareto frontiers, of a non-hierarchically decomposable system is depicted in Figure 49.

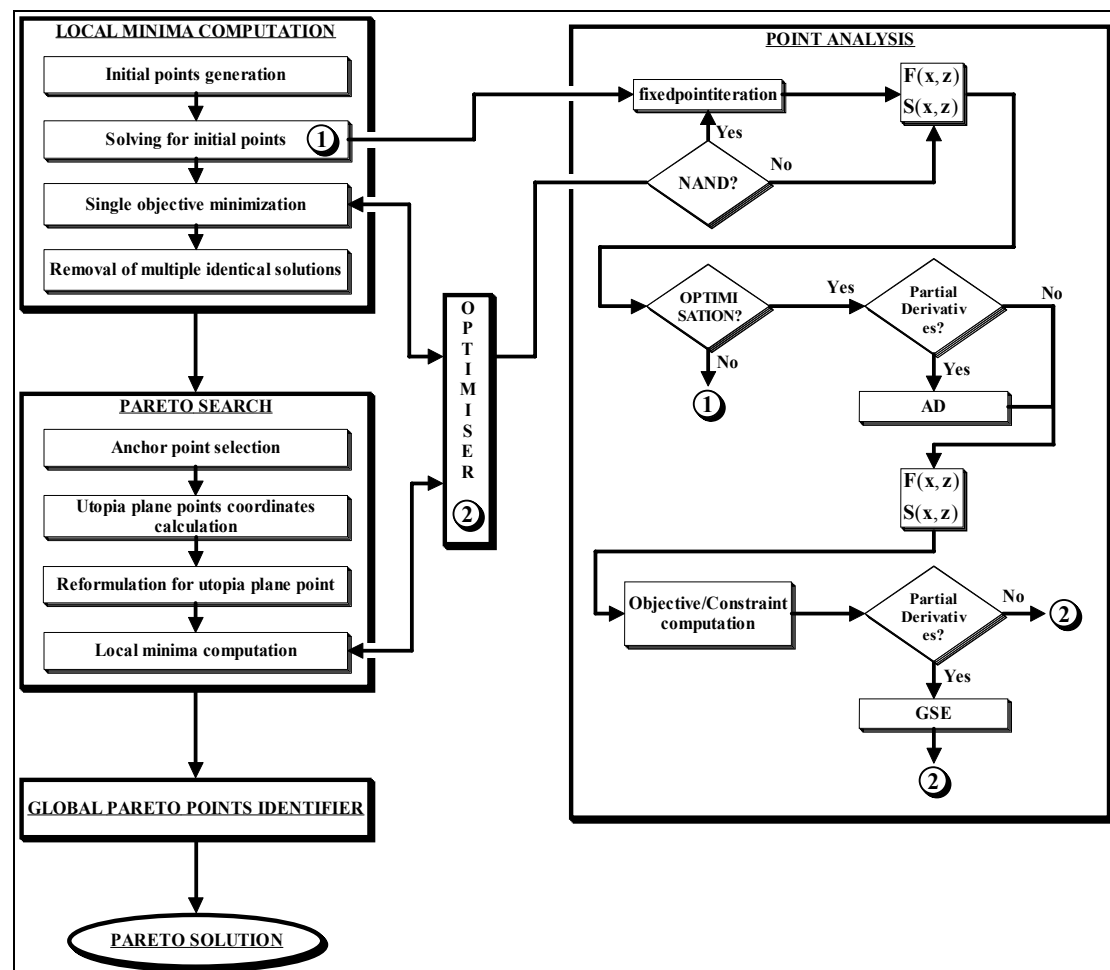


Figure 49: Multiobjective Multidisciplinary Algorithm Flowchart

For simplifying the description of the algorithm we consider four main sub-algorithms: the LOCAL MINIMA COMPUTATION, the PARETO SEARCH, the POINT ANALYSIS and finally the GLOBAL PARETO POINTS IDENTIFIER sub-algorithm.

1. LOCAL MINIMA COMPUTATION

The steps followed by the Local Minima Computation sub-algorithm are similar to the steps 1-3 of the global optimisation procedure of section 3.5.5, with the difference that the procedure is extended to take into account the R coupling variables (section 4.1). The input required by the sub-algorithm is the number of starting points L . The outputs are the L_i^* local minima for each of the M objective functions i .

The steps of this sub-algorithm are the following:

a. Initial points generation

A Latin-Hypercube sampling procedure is used for obtaining L well distributed starting points $\mathbf{x}_{SPi}$, $i = 1, \dots, L$ in the design space.

b. Solving for initial points

For each of the initial points $\mathbf{x}_{SPi}$, $i = 1, \dots, L$ the POINT ANALYSIS sub-algorithm is called in order to determine the value of the coupling variable $\mathbf{z}_{SPi} = \mathbf{S}(\mathbf{x}_{SPi}, \mathbf{z}_{SPi})$.

c. Single objective minimisation

L single-objective optimisations are performed by means of sub-algorithm POINT ANALYSIS for each of the M objectives. Thus for objective i and starting point $(\mathbf{x}_{SPj}, \mathbf{z}_{SPj})$ the optimisation problem is formulated as:

c.1. for the NAND case:

$$\begin{array}{ll}
\min_{\mathbf{x}} F_i(\mathbf{x}) \\
\text{subject to } M_{ic} \\
\text{inequality constraints:} & g_{m_{ic}}(\mathbf{x}, \mathbf{z}) \leq 0, \quad m_{ic} = 1, \dots, M_{ic} \\
M_{ec} \text{ equality constraints:} & h_{m_{ec}}(\mathbf{x}, \mathbf{z}) = 0, \quad m_{ec} = 1, \dots, M_{ec} \\
\text{optimisation starting point:} & (\mathbf{x}_{SPj}, \mathbf{z}_{SPj})
\end{array}$$

c.2. for the SAND case:

$$\begin{array}{ll}
\min_{\mathbf{x}, \mathbf{z}} F_i(\mathbf{x}, \mathbf{z}) \\
\text{subject to } M_{ic} \\
\text{inequality constraints:} & g_{m_{ic}}(\mathbf{x}, \mathbf{z}) \leq 0, \quad m_{ic} = 1, \dots, M_{ic} \\
M_{ec} \text{ equality constraints:} & h_{m_{ec}}(\mathbf{x}, \mathbf{z}) = 0, \quad m_{ec} = 1, \dots, M_{ec} \\
R \text{ equality constraints:} & z_r = S_r(\mathbf{x}, \mathbf{z}), \quad r = 1, \dots, R \\
\text{optimisation starting point:} & (\mathbf{x}_{SPj}, \mathbf{z}_{SPj})
\end{array}$$

yielding *local anchor points* $\mu_{ij}^* \in \mathbb{R}^M$, $i \in \{1, \dots, M\}$, $j \in \{1, \dots, L_i^*\}$ and corresponding design points and coupling variables $(\mathbf{x}_{ij}^*, \mathbf{z}_{ij}^*) \in \mathbb{R}^{N+M}$. Where L_i^* is the number of local optima obtained for objective i .

d. Removal of multiple identical solutions

In order to remove multiple coinciding *anchor points*, for each objective i it is checked that $\|\bar{\mathbf{x}}_{ij}^* - \bar{\mathbf{x}}_{ik}^*\|_\infty > \varepsilon$ for each $j, k \in \{1, \dots, L_i^*\}$, $j \neq k$. Where, each component of the design point vectors is normalised with respect to its lower and upper bounds, i.e. $\bar{x}_{ijl}^* = (x_{ijl}^* - x_{Ul}) / (x_{Ll} - x_{Ul})$ for $l = 1, \dots, N$. If for two points $\|\bar{\mathbf{x}}_{ij}^* - \bar{\mathbf{x}}_{ik}^*\|_\infty \leq \varepsilon$ then the μ_{ij}^* and μ_{ik}^* are compared and the

point with the greater value in case the objective is to be minimised, or the smaller value in case the point is to be maximised, is removed. The value of ε is a parameter and is to be chosen for $0 < \varepsilon \ll 1$.

2. PARETO SEARCH

The steps followed by the PARETO SEARCH sub-algorithm are similar to the steps 4-7 of the global optimisation procedure of section 3.5.5, but extended in order to incorporate the *peripheral region* determination of the *peripheral region anchor points* (section 3.5.4) and the formulation of the NC+ (section 3.5.3) method taking into account the NAND and SAND approaches.

The inputs to this sub-algorithm are the local minima obtained by the Local Minima Computation sub-algorithm, while the outputs are the Pareto points.

The steps of this sub-algorithm are the following:

a. Anchor point selection

For each objective i and *anchor point* j we determine $M-1$ anchor points $\bar{\mu}_{pk_p}^* = \bar{\mu}_{pk_p}^*$ for which:

$$\max_{k_1, \dots, k_p, \dots, k_M} \left(\min \left(\left\{ \left\| \bar{\mu}_{1,k_1}^* - \bar{\mu}_{2,k_2}^* \right\|, \dots, \left\| \bar{\mu}_{i-1,k_{i-1}}^* - \bar{\mu}_{i,j}^* \right\|, \left\| \bar{\mu}_{i,j}^* - \bar{\mu}_{i+1,k_{i+1}}^* \right\|, \dots \right\} \right) \right)$$

$$\dots, \left\| \bar{\mu}_{M-1,k_{M-1}}^* - \bar{\mu}_{M,K_M}^* \right\|, \left\| \bar{\mu}_{M,K_M}^* - \bar{\mu}_{1,K_1}^* \right\| \right)$$

where $\bar{\mu}_{ijl}^* = \mu_{ijl}^* / (\mu_{Ll} - \mu_{Ul})$ for $l=1, \dots, M$. Further details are given at step 4 of the global optimisation procedure in section 3.5.5.

b. Utopia plane points coordinates calculation

For objective i and relative local minimum j , *local anchor points*

$\mu_{1k_1}^*, \dots, \mu_{ij}^*, \dots, \mu_{Mk_M}^*$ are used for defining the K *utopia plane points*:

$$\mathbf{p}_{ijk} = \sum_{\substack{p=1 \\ p \neq i}}^M \alpha_p \bar{\mu}_{pk_p}^* + \alpha_i \mu_{ij}^*, \quad k = \{1, 2, \dots, K\}.$$

If $M > 2$ and $\mathbf{p}_{ijk}$ belongs to an edge, thus $\alpha_q = 0$ for $\forall q \neq u, u+1$ with $q, u \in \{1, \dots, M\}$, then as described in section 3.5.4, we also search for the *peripheral region* Pareto points by generating the *peripheral region utopia plane points*:

$$\mathbf{p}_{ijk}^+ = \mathbf{p}_{ijk} + qn_d \mathbf{s}_i, \quad q = 1, 2, \dots$$

where n_d corresponds to the distance between two adjacent *utopia plane points*.

c. Reformulation for utopia plane point

For *utopia plane point* $\mathbf{p}_{ijk}$ the optimisation problem is reformulated according to the NC+ method:

c.1. for the NAND case:

$$\min_{\mathbf{x}} \bar{F}_i(\mathbf{x}, \mathbf{z})$$

subject to M_{ic}

inequality

$$g_{m_{ic}}(\mathbf{x}, \mathbf{z}) \leq 0,$$

$$m_{ic} = 1, \dots, M_{ic}$$

constraints:

M_{ec} equality

$$h_{m_{ec}}(\mathbf{x}, \mathbf{z}) = 0,$$

$$m_{ec} = 1, \dots, M_{ec}$$

constraints:

subject to the additional $M - 1$ constraints:

$$\mathbf{v}_j \cdot (\mathbf{p}_{ijk} - \bar{\mathbf{F}}(\mathbf{x}, \mathbf{z})) \leq 0 \quad \forall j \in \{1, 2, \dots, M\}, j \neq i$$

and subject to the additional constraints:

$$\mathbf{v}_i \cdot \left(\bar{\mathbf{F}}(\mathbf{x}, \mathbf{z}) - \mathbf{p}_{ijk} - n_c \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \right) \leq 0$$

c.2. for the SAND case:

$$\min_{\mathbf{x}, \mathbf{z}} \bar{F}_i(\mathbf{x}, \mathbf{z})$$

subject to M_{ic}

inequality constraints: $g_{m_{ic}}(\mathbf{x}, \mathbf{z}) \leq 0$, $m_{ic} = 1, \dots, M_{ic}$

M_{ec} equality constraints: $h_{m_{ec}}(\mathbf{x}, \mathbf{z}) = 0$, $m_{ec} = 1, \dots, M_{ec}$

R equality constraints: $z_r = S_r(\mathbf{x}, \mathbf{z})$, $r = 1, \dots, R$

subject to the additional $M - 1$ constraints:

$$\mathbf{v}_j \cdot (\mathbf{p}_{ijk} - \bar{\mathbf{F}}(\mathbf{x}, \mathbf{z})) \leq 0 \quad \forall j \in \{1, 2, \dots, M\}, j \neq i$$

and subject to the additional constraints:

$$\mathbf{v}_i \cdot \left(\bar{\mathbf{F}}(\mathbf{x}, \mathbf{z}) - \mathbf{p}_{ijk} - n_c \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \right) \leq 0$$

where $\mathbf{v}_j$ and $\mathbf{v}_i$ are calculated, as described in section 3.5.3, and where

$\bar{\mathbf{F}}$ is normalised with respect to the *anchor points* considered thus

$$\bar{F}_l = \frac{F_l - \mu_{LBl}^*}{\mu_{UBl}^* - \mu_{LBl}^*} \quad \text{with} \quad \mu_{UBl}^* = \max \left\{ \mu_{1k_1^*l}^*, \dots, \mu_{ijl}^*, \dots, \mu_{Mk_M^*l}^* \right\} \quad \text{and}$$

$$\mu_{LBl}^* = \min \left\{ \mu_{1k_1^*l}^*, \dots, \mu_{ijl}^*, \dots, \mu_{Mk_M^*l}^* \right\}.$$

d. Local minima computation

An optimisation is performed in order to determine the Pareto solution relative to *utopia plane point* $\mathbf{p}_{ijk}$. The optimiser will make use of the POINT ANALYSYS sub-algorithm in order to obtain the desired solution.

3. POINT ANALYSIS

The sub-algorithm POINT ANALYSIS has the function of executing the complex system, thus determining $\mathbf{y} = \mathbf{F}(\mathbf{x}, \mathbf{z})$ and $\mathbf{z}_O = \mathbf{S}(\mathbf{x}, \mathbf{z})$, eventually manipulating its inputs and outputs according to the calling sub-algorithm.

Given an input point $(\mathbf{x}, \mathbf{z})$, the sub-algorithm returns as output $(\mathbf{y}, \mathbf{z}')$ or, if called by the OPTIMISER, the objectives and constraints of the optimisation problem. If also the Jacobians are required by the optimiser then these are returned too.

The steps of this sub-algorithm are the following:

a. If NAND (The NAND approach is being followed)

Given input variables $(\mathbf{x}, \mathbf{z})$ from one of the calling sub-algorithms, a fixed point iteration procedure is performed in order to determine the converged value of the coupling variables $\mathbf{z}'$ such that $\mathbf{z}' = \mathbf{S}(\mathbf{x}, \mathbf{z}')$. $\mathbf{y} = \mathbf{F}(\mathbf{x}, \mathbf{z}')$ and $\mathbf{z}' = \mathbf{S}(\mathbf{x}, \mathbf{z}')$ are then obtained.

b. If not NAND (The SAND approach is being followed)

Given input variables $(\mathbf{x}, \mathbf{z})$ from one of the calling sub-algorithms a $\mathbf{y} = \mathbf{F}(\mathbf{x}, \mathbf{z}')$ and $\mathbf{z}' = \mathbf{S}(\mathbf{x}, \mathbf{z}')$ are then obtained with $\mathbf{z}' = \mathbf{z}$.

c. If an optimisation is being performed

Given the input variables $(\mathbf{x}, \mathbf{z}')$ obtained from steps a or b

i. If partial derivatives (Partial derivatives are to be computed)

If the optimiser requires the Jacobians to be computed

1. AD

Automatic differentiation is applied. The presented approach is valid for AD tools making use of operator overloading, for which the original variables $(\mathbf{x}, \mathbf{z}')$ are transformed with variables of an automatic differentiation class. For further details on operator overloading see section 2.4.

ii. $\mathbf{y} = \mathbf{F}(\mathbf{x}, \mathbf{z}')$ and $\mathbf{z}' = \mathbf{S}(\mathbf{x}, \mathbf{z}')$ computation

In case AD was applied, the results obtained from the computation of the system will yield $\mathbf{y}$ and $\mathbf{z}'$ of class AD.

iii. Objectives/Constraints computation

$\mathbf{opt}_{out} = [F_i, \mathbf{g}, \mathbf{h}, \mathbf{c}]$ is computed according to the formulation of the optimisation problem according to the sub-algorithm requiring the optimisation to be performed. With $\mathbf{c}$ we indicate for brevity the M additional constraints of the NC+ method. If a single objective optimisation was to be performed, then $\mathbf{c}$ is empty.

iv. If AD/Converged system

If automatic differentiation was applied and the system is converged, which is always the case if the NAND approach was followed then,

1. GSE

Global sensitivity equations are applied in order to determine the correct Jacobians as described in section 4.2.

4. IDENTIFICATION OF GLOBAL PARETO POINTS

Out of all the Pareto points obtained by the PARETO SEARCH sub-algorithm the IDENTIFICATION OF GLOBAL PARETO POINTS sub-algorithm determines the global Pareto points by pairwise comparison.

For doing this, the concept of dominance (section 2.3.1) is applied as follows. For each pair of Pareto solutions $(\mathbf{x}_A, \mathbf{z}_A)$ and $(\mathbf{x}_B, \mathbf{z}_B)$, if $F_i(\mathbf{x}_A, \mathbf{z}_A) < F_i(\mathbf{x}_B, \mathbf{z}_B)$ ($i = 1, \dots, M$) then point $(\mathbf{x}_B, \mathbf{z}_B)$ is dominated by point $(\mathbf{x}_A, \mathbf{z}_A)$, thus $(\mathbf{x}_B, \mathbf{z}_B)$ is not a global Pareto point.

5.2 Implementation

The CODE-M algorithm was implemented in MATLAB. A user friendly interface was developed in order to simplify the setup of the different test cases.

Furthermore the MATLAB code was compiled in order to produce a stand alone version. Portions of the code were written making use of object oriented programming, this in order to ease the handling of independent variables, constant variables, objective functions and constraints.

The *lhsdesign* function, part of the Statistics Toolbox was used for determining the starting points. *lhsdesign* is an implementation of a Latin-Hypercube sampling algorithm (The MathWorks, Inc., 2007b). The *fmincon* function, part of the Optimisation Toolbox was used as optimiser (The MathWorks, Inc., 2007a). *fmincon* is an SQP algorithm which solves a sequence of quadratic programming problems in order to minimise the Lagrangian function, a linear combination of the scalar objective function and its constraints. For avoiding the computation of the Hessian of the Lagrangian function, the BFGS method was chosen for generating its approximation. For determining the solution at each step of the SQP algorithm the *line search* method was chosen.

The MAD (MATLAB AUTOMATIC DIFFERENTIATION) library was used for Automatic Differentiation. MAD, which is part of the TOMLAB package (Forth and Edvall, 2007), was developed by Shaun Forth of Cranfield University. MAD makes use of the operator overloading approach in order to determine the analytical derivatives. Its application is fairly simple, requiring only the transformation of the input variables to a MATLAB code into objects of class *fmad*. A limitation of MAD is that its applicability is limited only to MATLAB functions, thus requiring a different AD tool and the modification of the CODE-M implementation for applying AD with non MATLAB contributing analysis.

Even though the CODE-M implementation was developed in MATLAB an interface allowing the integration of external executables as systems to be optimised has been developed.

5.3 Summary

In this chapter the CODE-M algorithm for solving global multiobjective multidisciplinary optimisation problem is presented. Together with a detailed description of the algorithm, details of its implementation are given.

The implemented CODE-M algorithm is used in the following chapter in order to evaluate its various features on a set of test cases. It is also applied for performing a multiobjective multidisciplinary design on a simplified aircraft test case (USMAC). More details on the implementation are given in Appendix.

6 Tests, Results, Analysis and Discussion

For analysing the effectiveness of the CODE-M algorithm proposed in chapter 5 a number of test cases were selected. The aim was that of evaluating the algorithm in its various features. Thus in each of the following sections a different aspect is considered. First the capabilities of the NC+ method in solving the limitations of the NC method are shown in section 6.1, by means of some simple test cases. In section 6.2 the results obtained for a number of global multiobjective optimisation problems are presented. The results demonstrate the unique capability of the algorithm in obtaining both global and local Pareto sets. In section 6.3 the results obtained for a number of test cases for non-hierarchically decomposable systems are given. In section 6.4 the efficiency of AD is analysed. Finally, in section 6.5 the algorithm is applied to a simplified multiobjective multidisciplinary aircraft conceptual design optimisation problem and the results obtained are presented.

6.1 The NC+ method

Some limitations of the existing multiobjective optimisation methods, which make use of the reformulation approach, have been identified through the evaluation of these methods with a number of existing test cases (see chapter 3); this lead initially to the development of the modified PP based method, subsequently to the development of the DHCBI method and finally to the development of the NC+ method. Although the DHCBI method solved the identified limitations, it was noticed that on a number of simple test cases it performed poorly with respect to the NC method, from a computational cost point of view; this lead to the development of the NC+ method.

6.1.1 NC+ computational cost

In order to evaluate the computational cost of the NC+ method with respect to the NC method the same test cases used for comparing the DHCBI and NC method (Table 2) were used. The characteristics of the test cases are reported in sections 3.1 and 3.2.1 and are briefly summarised:

- CRC_cvx / CRC_ccv:
2 objectives optimisation problem
unique *anchor points*
11 *utopia plane points*
- SPHR_ccv-Case1 (see SPHR_ccv-Case1 in section 3.1)
3 objectives optimisation problem
fixed *anchor points*
28 *utopia plane points* obtained with the NBI method (no *peripheral region* considered)

As also stated in section 3.5.3 the *anchor points* are considered fixed in order to focus the attention only on the number of analyses required for obtaining the Pareto points.

As can be seen the two methods required identical number of iterations for obtaining the solutions, the reason can be attributed to the similarity in the formulation of the optimisation problems for the two methods.

	NC Number of analyses	NC+ Number of analyses
CRC_cvx	202	202
CRC_ccv	137	137
SPHR_ccv-Case1	420	420

Table 3: Comparison of the number of analyses performed by the NC and NC+ method

6.1.2 Independence of the NC+ method from the positioning of the *anchor points*

As it was shown in section 3.2.1, the NC method is not capable of obtaining Pareto points when the *anchor points* have particular relative positions. One such case is shown in Figure 24b. To demonstrate that the NC+ method solves this problem it was applied to the same test cases. The results obtained are shown in Figure 50 for the 3 objective test cases with fixed *anchor points* SPHR_ccv-Case1 and SPHR_ccv-Case4 (see section 3.2.1). By comparing these results with the ones obtained with the

NC method (Figure 24), it is possible to see that the NC+ method is not affected by the positioning of the *anchor points*. In fact, as shown in Figure 50b the Pareto points obtained are well distributed over the Pareto front corresponding to the portion of the sphere, while as shown in Figure 24a the original NC method was capable of obtaining as a Pareto solution only *anchor point* μ_2^* .

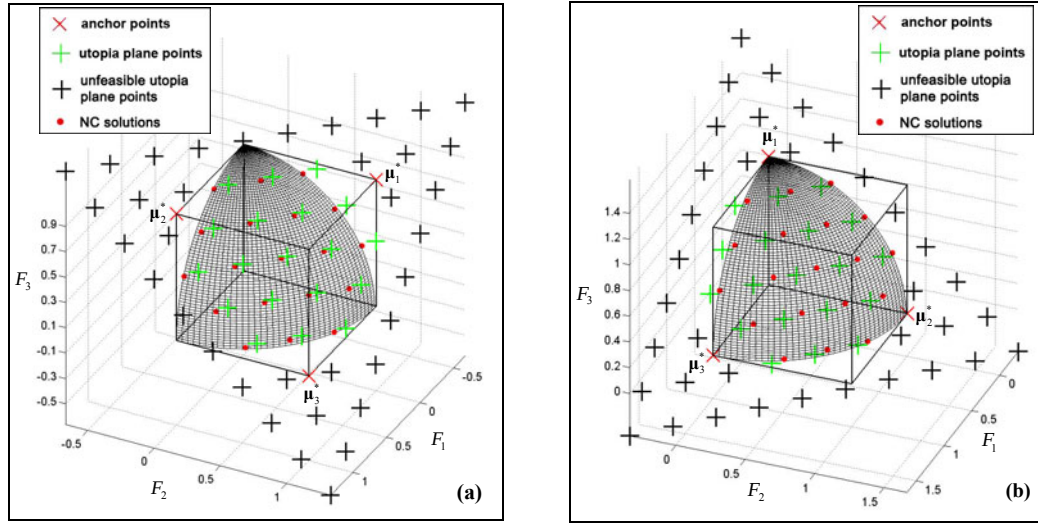


Figure 50: NC+ solution for problem SPHR_ccv: (a) Case1, (b) Case4

6.1.3 Complete representation of the Pareto front

One important aspect that has to be taken in account in order to obtain Pareto points representative of the complete Pareto front, is the ability of the methods to obtain Pareto points belonging to the *peripheral region*. For this purpose initially the NC method solved this problem by enlarging the polygon of vertices μ_i^* , in order to have the orthogonal projection of the Pareto front completely enclosed in the enlarged polygon. As shown in section 3.1 the position of the *anchor points* can potentially greatly reduce the number of *utopia plane points*. This problem was initially solved by the proposed modified PP-based method (section 3.5.1), which allowed the rotation of the search domain for the *utopia plane points* belonging to the edge of the polygon. The solution obtained by means of the rotation of the search domain obtained with the modified PP-based method for test case half_SPHR_cvx is shown in Figure 39. A second more general approach, applicable to any of the reformulation

methods, is proposed in section 3.5.4. The results obtained with this approach in conjunction with the NC+ method, for the same test case, are shown in Figure 51. As shown in the figure, this approach allows obtaining the Pareto points belonging to the *peripheral region* similarly to the rotation approach. Analogously to the results presented in Figure 39 the number of *utopia plane points* requested was 66. As a result 99 Pareto points were obtained. As can be observed by comparing Figure 39 and Figure 51, while with the modified PP-based method redundant Pareto points were obtained (in particular on the edge of the Pareto front), the distribution obtained with this more general approach allows to obtain well-distributed Pareto points in all regions of the Pareto front.

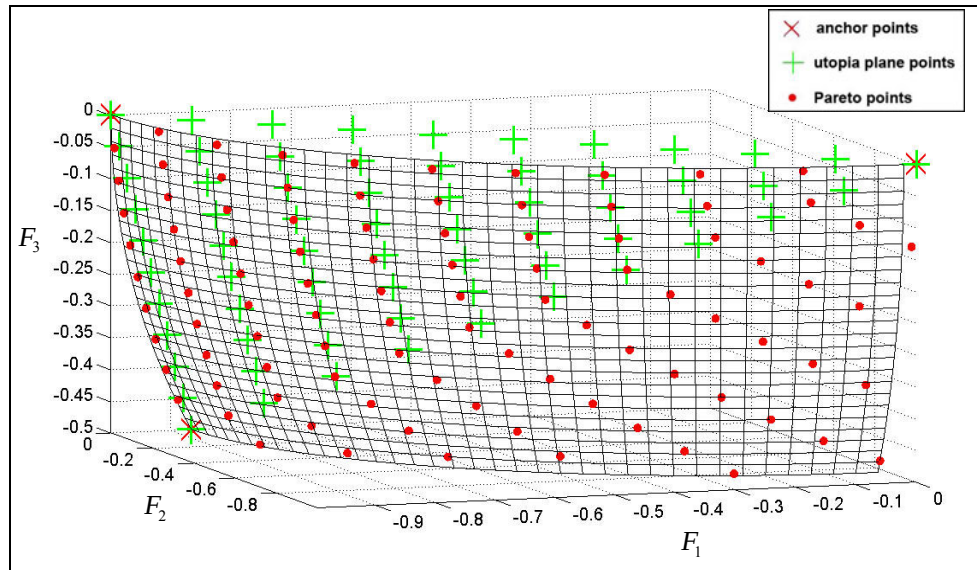


Figure 51: half_SPHR_cvx solution obtained with the NC+ method showing the capability of the algorithm in obtaining the solutions belonging to the *peripheral region*

6.2 NC+ Global optimisation

The procedure described in section 3.5.5 is applied to a number of multiobjective global optimisation problems. The aim is to demonstrate the capability of the procedure in conjunction with the NC+ method to obtain the local and global Pareto frontiers.

The following multiobjective optimisation test cases have been developed for evaluating Genetic Algorithms; all of these are summarised in Van Veldhuizen and

Lamont (1999). The test cases have been developed in order to introduce the different difficulties that can be encountered when performing global Multiobjective optimisation. It must be noted that, to the author's knowledge, for the NBI, PP-based and NC method no attempt at solving problems of this level of complexity has ever been attempted.

For all of the test cases parameter n_c of the NC+ method was computed as $1/4$ of the distance between two adjacent *utopia plane points*. Also, for all optimisation problems the procedure described in section 3.5.5 was applied with number of starting points, $L = 50$, and number of requested *utopia plane points*, $K = 50$.

It must be noted that the weak Pareto solutions are included as part of the global Pareto solutions, since the round-off error can make the distinction between global and local Pareto points difficult. As described in section 5.1, for determining the global solutions the Pareto points obtained are compared pairwise. If one of the two points is dominated, then the dominated one is a local Pareto solution. Thus for a minimisation problem of M objective functions $F_i(\mathbf{x})$ ($i = 1, \dots, M$), when comparing two Pareto solutions $\mathbf{x}_A$ and $\mathbf{x}_B$, if $F_i(\mathbf{x}_A) < F_i(\mathbf{x}_B)$ ($i = 1, \dots, M$) then point $\mathbf{x}_B$ is dominated by point $\mathbf{x}_A$ and $\mathbf{x}_B$ is a local Pareto point. Still, due to the round-off error the real comparison performed is $F_i(\mathbf{x}_A + \varepsilon_A) < F_i(\mathbf{x}_B + \varepsilon_B)$, where ε_A and ε_B are the errors for the two solutions. Thus it is possible that if the difference between $F_i(\mathbf{x}_A)$ and $F_i(\mathbf{x}_B)$ is very small a global solution could be considered dominated even when this is not the case. To avoid this problem, the dominated solutions are identified by evaluating $F_i(\mathbf{x}_A) + \varepsilon < F_i(\mathbf{x}_B)$, for a predetermined value ε , in order to remove the possibility of erroneously considering any global Pareto solution as local, with the drawback of not being able to remove some of the weak Pareto solutions.

6.2.1 Problem Deb1

Problem Deb1 was initially used in section 3.3 in order to demonstrate the global optimisation problem, where its formulation can also be found. Here the proposed global optimisation procedure is applied to test case Deb1 in order to demonstrate the capability of the procedure to obtain both the local and global Pareto frontier. It

should be noted that although discontinuities are present, the NC+ algorithm was still capable of obtaining the correct solution.

The results obtained for this test case are shown in Figure 52, Figure 53 and Figure 54. 10835 analyses were required for obtaining the solution. Figure 52 shows the two overlapping objective functions. The values x_1 and x_2 for each vertical line correspond to the values of the independent variables of each solution obtained. The values of the independent variables for each of the solutions obtained are also shown in Figure 53, where the two overlapping contour lines for the two objective functions are also shown. Figure 54 shows the solution with respect to the criterion (objective space).

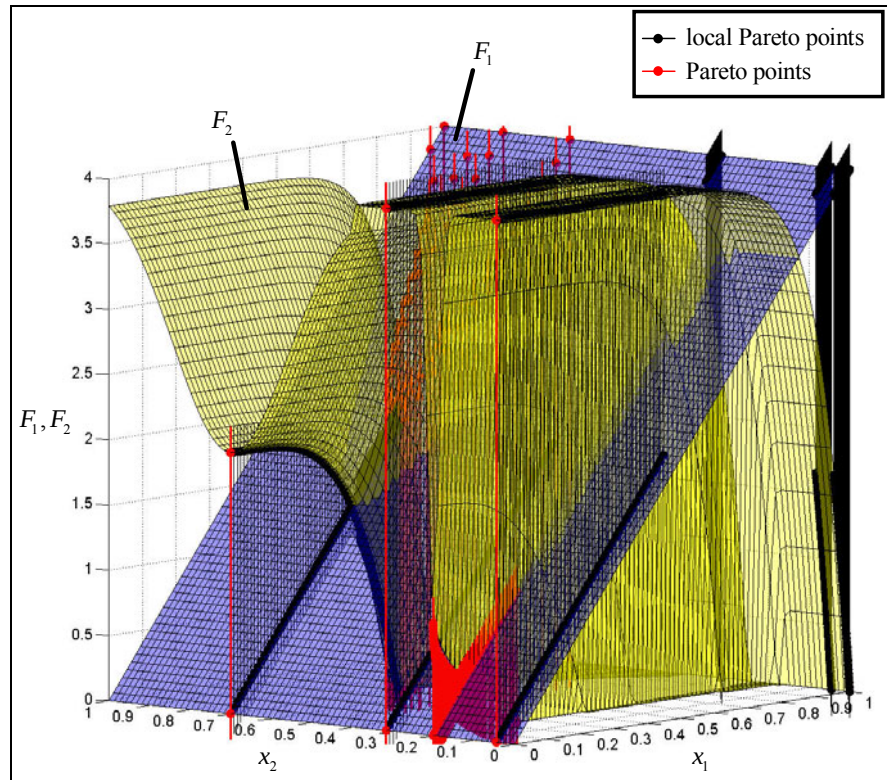


Figure 52: Deb1 solution obtained with the NC+ method (mixed criterion design space)

By examining the three figures it is possible to see that both the global and local Pareto frontiers are identified by the red and black dots, respectively. As shown in Figure 30, because of the extension of the region of global minima of the two objective functions in the design space, many weak Pareto points are included in the global Pareto set.

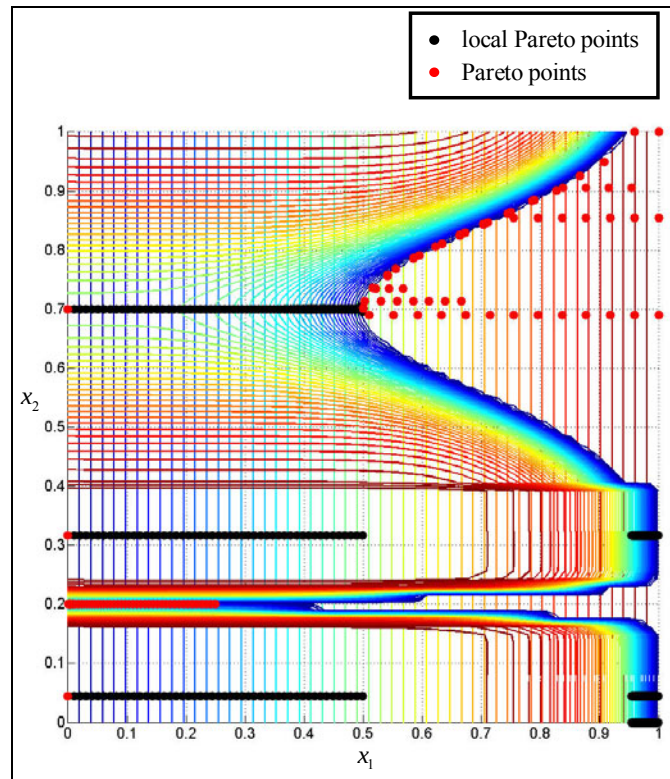


Figure 53: Deb1 solution obtained with the NC+ method (design space)

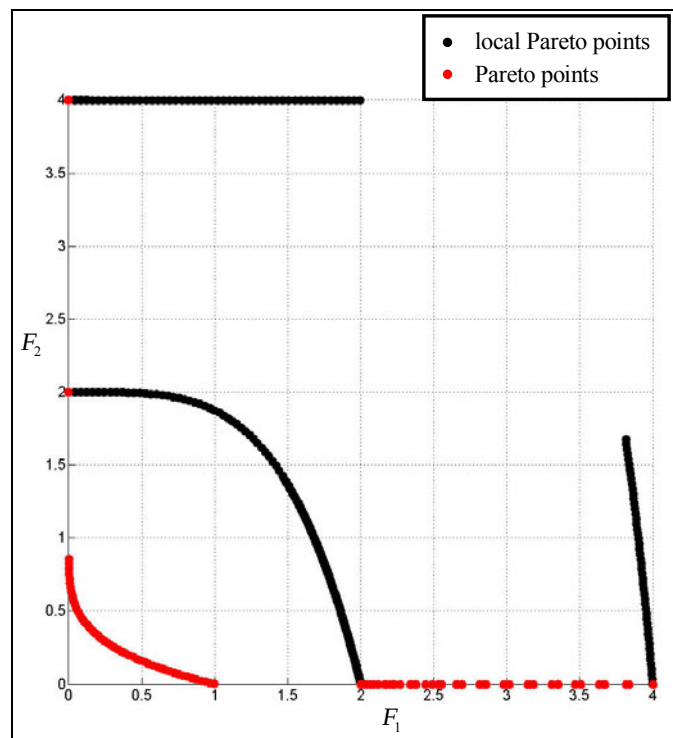


Figure 54: Deb1 solution obtained with the NC+ method (criterion space)

6.2.2 Problem Deb2

The formulation of test case Deb2 is the following:

Problem Deb2:

$$\min[F_1(x_1), F_2(x_1, x_2)]$$

where: $F_1(x_1) = x_1$ and $F_2(x_1, x_2) = g(x_2) \cdot h(F_1, g)$

where: $g(x_2) = 1 + 10x_2$

and where: $h(F_1, g) = 1 - \left(\frac{F_1}{g}\right)^\alpha - \frac{F_1}{g} \sin(2\pi q F_1)$

for $0 \leq x_i \leq 1, i = 1, 2$

where $\alpha = 2$ and $q = 5$.

The peculiarity of test case Deb2 is the discontinuity of the Pareto front, moreover, multiple local minima exist for each of the single objective functions, since h is a periodic function. In this test case parameter q defines the number of discontinuous regions in a unit interval.

The results obtained are shown in Figure 55, Figure 56 and Figure 57. 4584 analyses were required for obtaining the complete solution. In Figure 55 a portion of the mixed design criterion space is shown, with the two overlapping surfaces representing the two objective functions. The figure also shows the discontinuity of the Pareto solution in the design space. The values of the independent variables are also shown in Figure 56, together with the overlapping contour lines of the two objective functions. Figure 57 shows the values of the objective functions for the Pareto points. The discontinuity of the global Pareto frontier in the criterion space is also shown.

As it can be observed, the global optimisation procedure in conjunction with the NC+ method was capable of generating Pareto points in all regions of the global Pareto front.

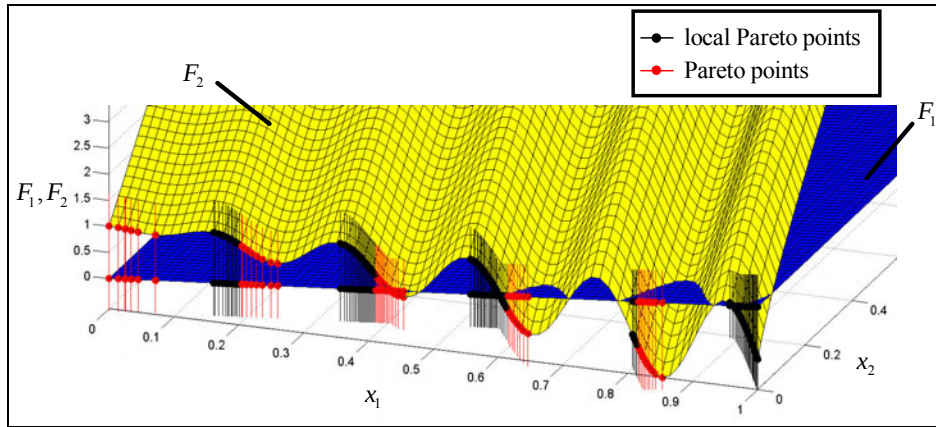


Figure 55: Deb2 solution obtained with the NC+ method (mixed criterion design space)

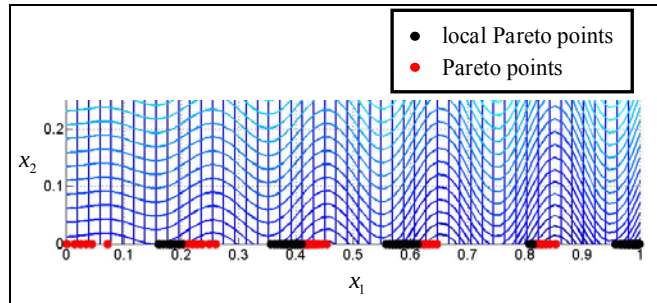


Figure 56: Deb2 solution obtained with the NC+ method (design space)

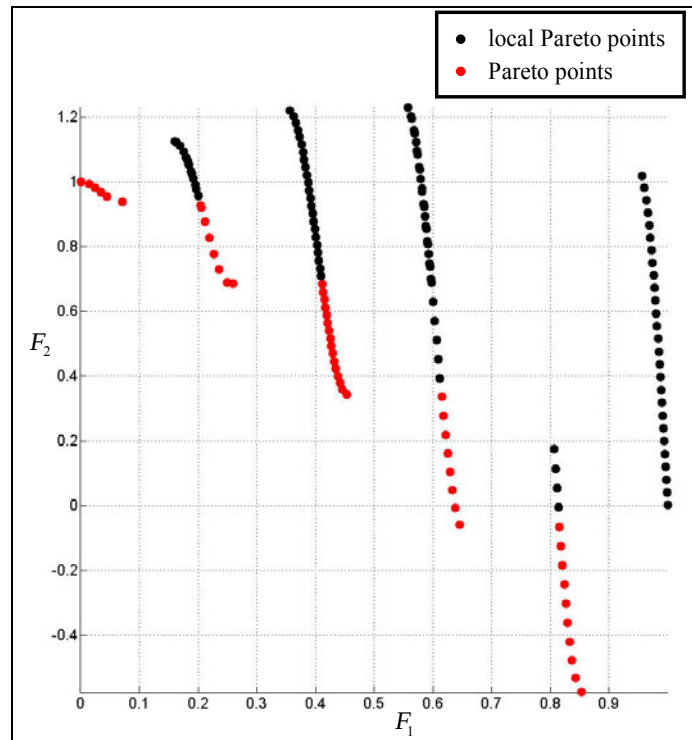


Figure 57: Deb2 solution obtained with the NC+ method (criterion space)

6.2.3 Problem Kursawe:

The formulation of test case Kursawe is the following:

Problem Kursawe:

$$\min[F_1(\mathbf{x}), F_2(\mathbf{x})]$$

where:
$$F_1(\mathbf{x}) = \sum_{i=1}^N \left(-10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}} \right)$$

and where:
$$F_2(\mathbf{x}) = \sum_{i=1}^N \left(|x_i|^{0.8} + 5 \sin(x_i)^3 \right).$$

for:
$$-5 \leq x_i \leq 5, \quad i = 1, \dots, N$$

The peculiarity of test case Kursawe is its symmetry, as shown in Figure 58 and more clearly by the contour lines in Figure 59. Furthermore, the global Pareto front is also discontinuous.

33345 analyses were required for obtaining the solution. Figure 58 shows the results, including a visualisation of the two surfaces representing the two objective functions. Figure 59 shows the results obtained in the design space together with the overlapping contour lines of the two objective functions. Figure 60 shows the solution obtained in the criterion space.

As it can be seen from the figures, the global Pareto front is composed of four distinct regions, one of which corresponds to the single point for which $\mathbf{x} = (0, 0)$, the minimum of objective function F_1 .

The figures show that the procedure in conjunction with the NC+ method is capable of obtaining to complete global Pareto front, regardless of the symmetry problem.

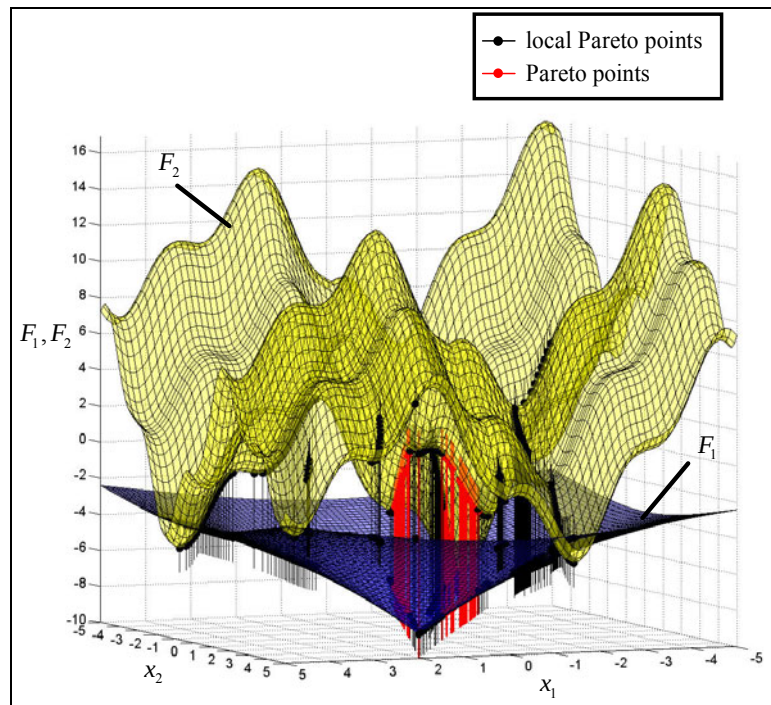


Figure 58: Kursawe solution obtained with the NC+ method (mixed criterion design space)

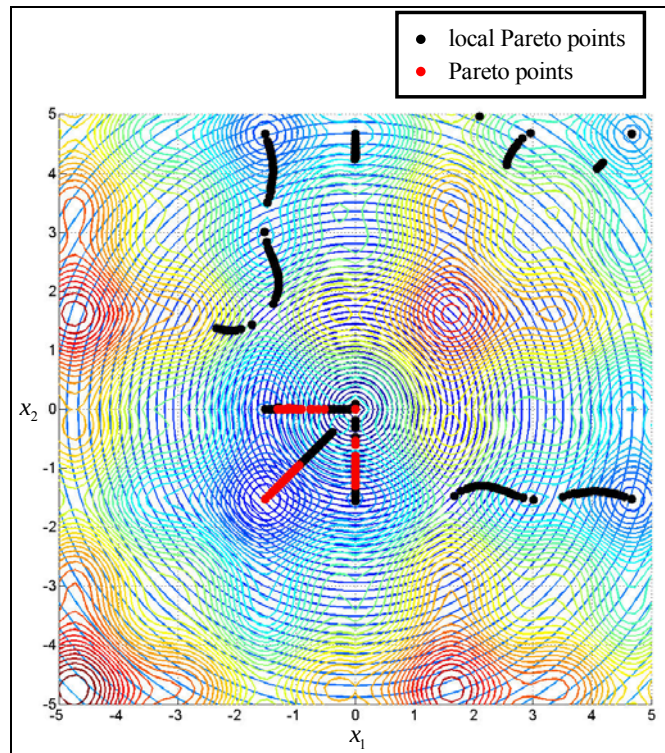


Figure 59: Kursawe solution obtained with the NC+ method (design space)

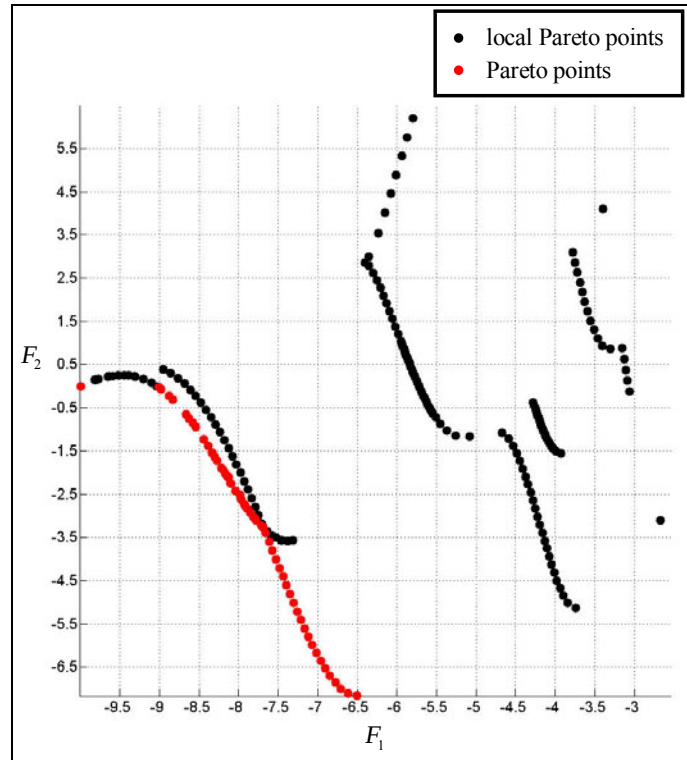


Figure 60: Kursawe solution obtained with the NC+ method (criterion space)

6.2.4 Problem Fonseca:

The formulation of test case Fonseca is the following:

Problem Fonseca:

$$\min[F_1(\mathbf{x}), F_2(\mathbf{x})]$$

where:

$$F_1(\mathbf{x}) = 1 - e^{-\sum_{i=1}^N \left(x_i - \frac{1}{\sqrt{N}}\right)^2}$$

and where:

$$F_2(\mathbf{x}) = 1 - e^{-\sum_{i=1}^N \left(x_i + \frac{1}{\sqrt{N}}\right)^2}$$

for:

$$-2 \leq x_i \leq 2, \quad i = 1, \dots, N$$

The solution for the relatively simple Fonseca test case is shown in Figure 61, Figure 62 and Figure 63. For obtaining the solution 3140 analysis were required. The overlapping surfaces of the two objective functions are shown in Figure 61, together with the solution. Figure 62 shows the results obtained in the design space together

with the overlapping contour lines of the two objective functions. Figure 63 shows the solution obtained in the criterion space.

As it can be seen from the figures, optimisation problem Fonseca does not present any local Pareto front, but a single continuous global Pareto front.

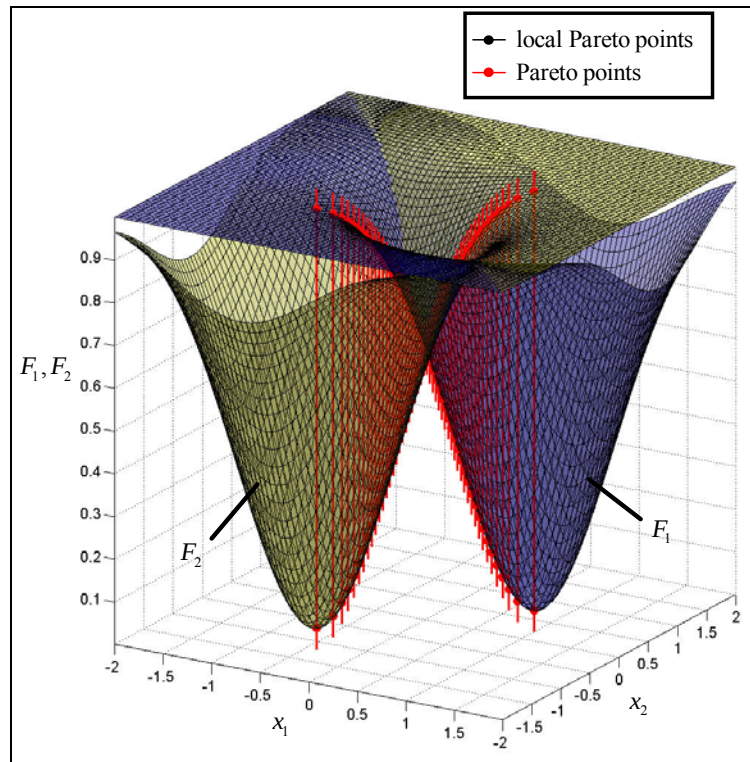


Figure 61: Fonseca solution obtained with the NC+ method (mixed criterion design space)

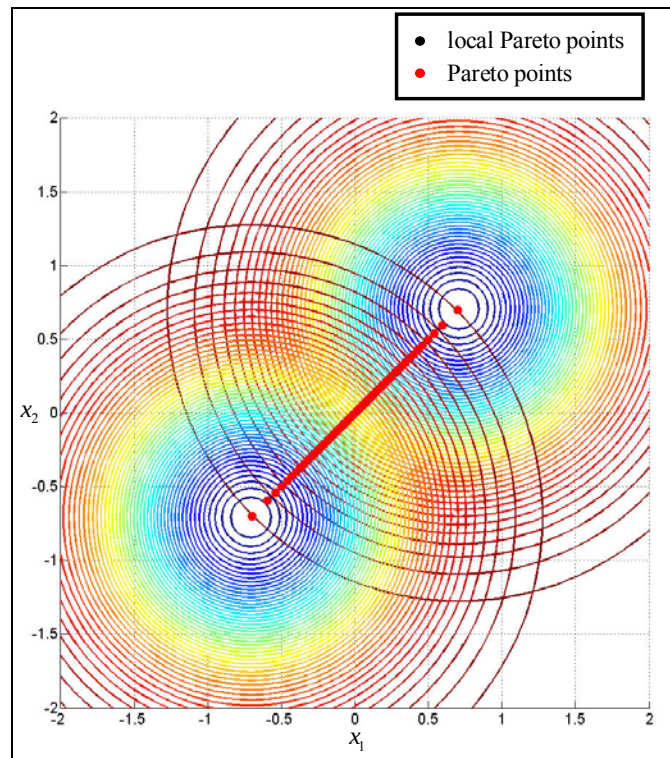


Figure 62: Fonseca solution obtained with the NC+ method (design space)

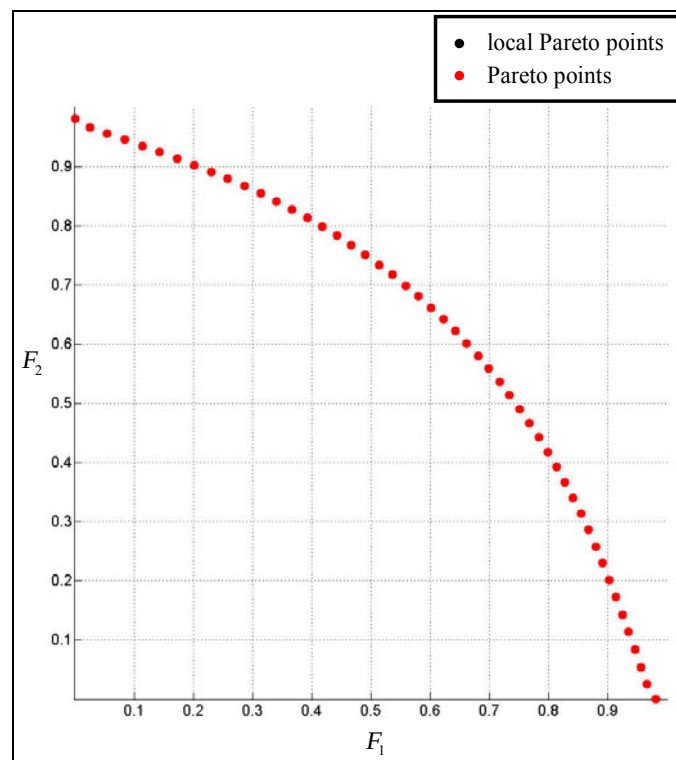


Figure 63: Fonseca solution obtained with the NC+ method (criterion space)

6.2.5 Problem Viennet:

The formulation of test case Viennet is the following:

Problem Viennet:

$$\min[F_1(\mathbf{x}), F_2(\mathbf{x}), F_3(\mathbf{x})]$$

where: $F_1(\mathbf{x}) = \frac{1}{2}(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$

where: $F_2(\mathbf{x}) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15$

and where: $F_3(\mathbf{x}) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1e^{(-x_1^2 - x_2^2)}$.

for: $-3 \leq x_i \leq 3, i = 1, \dots, 2$

The Pareto front of test case Viennet is widely distributed, with objective functions of F_1 , F_2 and F_3 scaled differently, as can be observed in Figure 64.

For obtaining the results shown in Figure 64, Figure 65 and Figure 66, 61116 analyses were performed. Figure 64 shows the three overlapping surfaces representing the 3 objective functions and the results obtained. The values of the independent variables for the solutions together with the contour lines of the objective functions are shown in Figure 65. The solutions obtained in the criterion space are shown in Figure 66.

As can be seen from the figures, although the objective functions are differently scaled and the global Pareto frontier is widely spread, the global optimisation procedure and the NC+ method were once again capable of determining the solution.

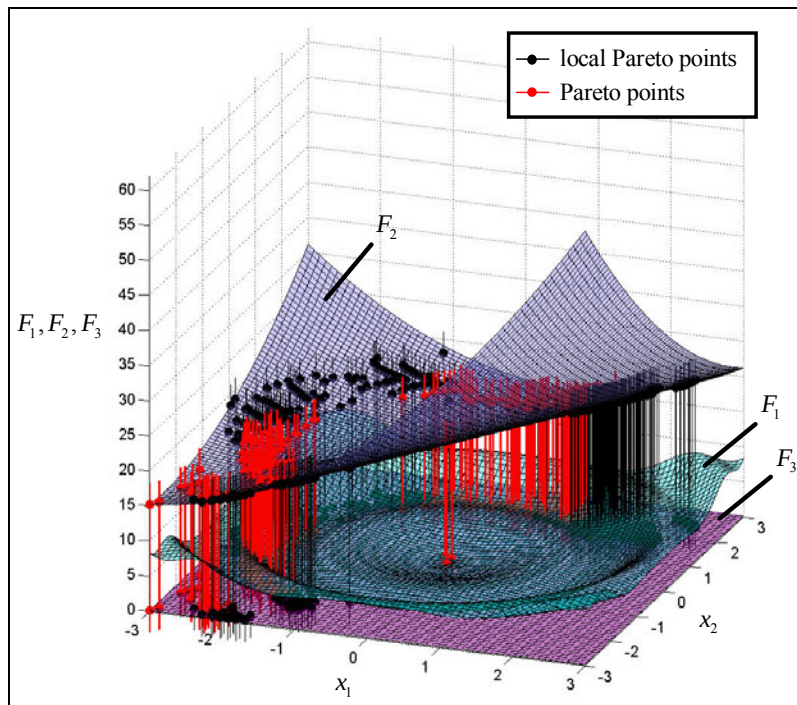


Figure 64: Viennet solution obtained with the NC+ method (mixed criterion design space)

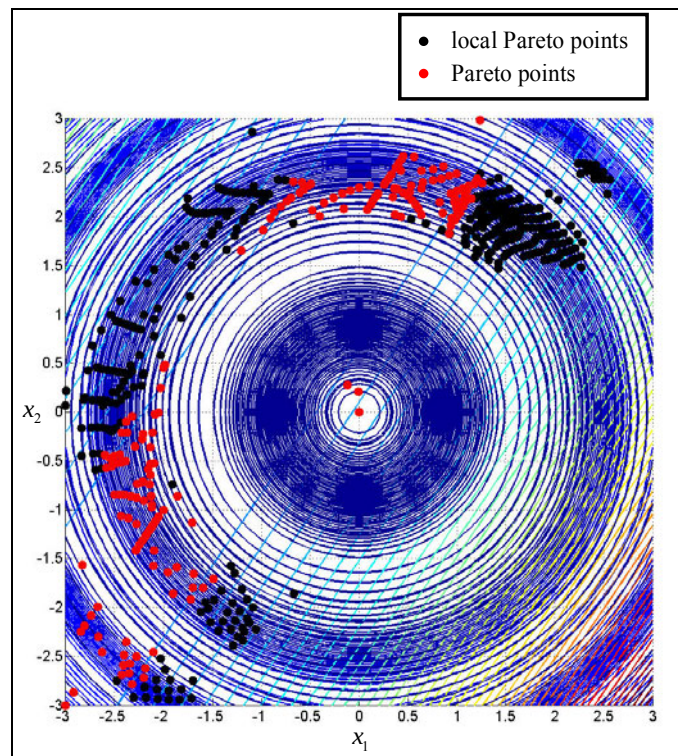


Figure 65: Viennet solution obtained with the NC+ method (design space)

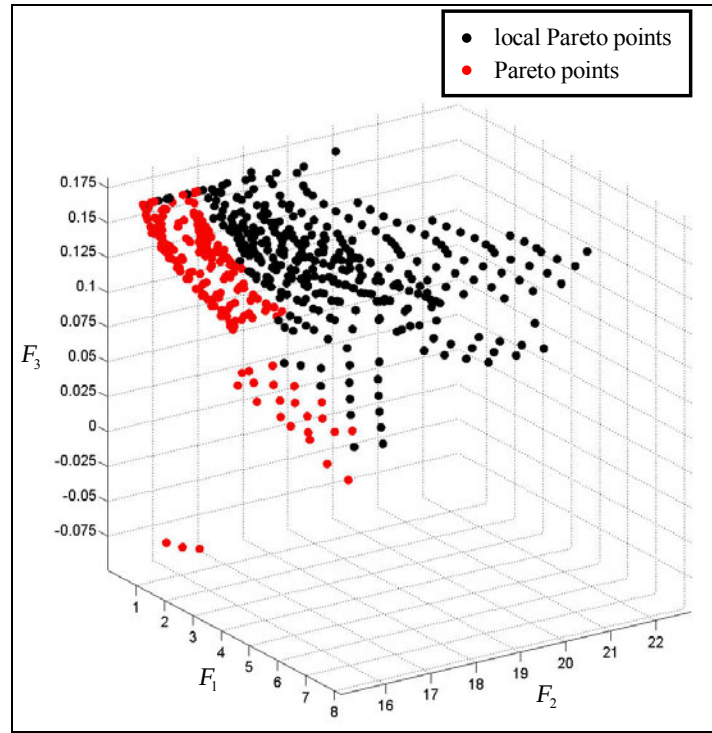


Figure 66: Viennet solution obtained with the NC+ method (criterion space)

The number of analyses required for solving the five different test cases presented in this section is summarised in Table 4. As can be seen, although the number of starting points, L and the number of requested *utopia plane points*, K were both fixed to a value of 50, the number of analyses performed varies greatly for the five test cases. The reason for this is related to how complicated the test cases are, in particular, the number of objectives considered and the number of local minima present. The two most demanding test cases are Kursawe, for which numerous local minima exist, and Viennet for which in addition to the fact that the objectives are 3 each of these have widely spread local minima.

Nevertheless, for all five cases the proposed approach was capable of generating well-distributed Pareto points on both global and local Pareto frontiers.

	Global Pareto procedure with NC+ method
	Number of analyses
Deb1	10835
Deb2	4584
Kursawe	33345
Fonseca	3140
Viennet	61116

Table 4: Number of analyses required for the solution of the Multiobjective optimisation test cases considered.

6.3 Multiobjective Multidisciplinary Optimisation

An advantage, when performing multiobjective optimisation with the methods which use the reformulation approach, is that for obtaining each subsequent Pareto point, the previously obtained Pareto point can be used as a starting point for the following optimisation. Thus, it is of interest to evaluate which among the NAND and SAND approaches can gain from this advantage.

For evaluating the NAND and SAND approaches discussed in section 4.1 the following two test cases have been considered.

SPHR_ND_ccv_MD

$$\min_{\mathbf{x}} \mathbf{F}(\mathbf{x})$$

where: $\sum_{i=1}^N x_i^2 \geq 1$

and where: $z_i = \frac{\pi}{2} \sin\left(\frac{z_i}{10} - \frac{x_i}{100}\right), i = 1, \dots, N$

for: $0 \leq x_i \leq 1, i = 1, \dots, N$

where $\mathbf{F}(\mathbf{x}) = \mathbf{x} = [x_1, \dots, x_N]^T$.

SPHR_3D_ccv_mltc_MD

$$\min_{\mathbf{x}} \mathbf{F}(\mathbf{x})$$

where: $\sum_{i=1}^N x_i^2 \geq 1$

where: $z_i = \frac{\pi}{2} \sin\left(\frac{z_i}{10} - \frac{x_i}{100}\right), i = 1, \dots, N$

and where: $x_i < 10^{-4}, i = 4, \dots, N$

for: $0 \leq x_i \leq 1, i = 1, \dots, 3$

where $\mathbf{F}(\mathbf{x}) = [x_1, \dots, x_3]^T$.

Both test cases are based on the SPHR_ccv problem (section 3.1). For test case SPHR_ND_ccv_MD the SPHR_ccv test case problem has been generalised to M dimensions, with the number of objectives, M , coinciding with the number of independent variables of the problem. The Pareto front for this test case will coincide with the portion of an N dimensional hyper-sphere for which $0 \leq x_i \leq 1$ for $i = 1, \dots, N$. For test case SPHR_3D_ccv_mltc_MD the number of objectives is 3, but all of the N independent variables are used for determining the radius constraint. Furthermore, in both test cases multidisciplinary is taken into account by introducing the N additional constraints, which are as many as the independent variables.

The following three cases were considered:

SPHR_2D_ccv_MD-Case1

The test case is based on test case SPHR_ND_ccv_MD for $N = 2$. The following *anchor points* are considered.

$$\boldsymbol{\mu}_1^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ and } \boldsymbol{\mu}_2^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

SPHR_3D_ccv_MD-Case1

The test case is based on test case SPHR_ND_ccv_MD for $N = 3$. The following *anchor points* are considered.

$$\mu_1^* = \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \mu_2^* = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix} \text{ and } \mu_3^* = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix}$$

SPHR_3D_ccv_mltc_MD-Case1

The test case is based on test case SPHR_3D_ccv_mltc_MD for $N = 8$. The following *anchor points* are considered.

$$\mu_1^* = \begin{bmatrix} 0 \\ 0.976 \\ 1 \end{bmatrix}, \mu_2^* = \begin{bmatrix} 0.982 \\ 0 \\ 0.936 \end{bmatrix} \text{ and } \mu_3^* = \begin{bmatrix} 1 \\ 0.921 \\ 0 \end{bmatrix}$$

For all three cases the number of requested *utopia plane points* is 55. The *anchor points* are fixed since, in particular for the SPHR_3D cases, they are not unique. Thus by fixing the *anchor points* we ensure that also the *utopia plane points* are fixed, therefore the optimisation problems to be solved by means of the NAND and SAND approaches, for each *utopia plane point* are the identical.

The results obtained for the three optimisation problems with fixed *anchor points* obtained with the NAND and SAND approaches are summarised in Table 5. For each of the three cases the number of *utopia plane points*, obtained with the NAND and SAND approaches, were identical and are reported in the table.

	Obtained Pareto points	NAND	SAND
SPHR_2D_ccv_MD-Case1	55	715	1265
SPHR_3D_ccv_MD-Case1	79	1386	1746
SPHR_3D_ccv_mltc_MD- Case1	79	1487	9941

Table 5: Comparison between the number of analyses required by the NAND and SAND approaches for three different test cases.

As can be seen in Table 5, the NAND approach performed better than the SAND approach for all three test cases. The difference between the number of analysis between the NAND and SAND approaches is particularly evident for SPHR_3D_ccv_mltc_MD-Case1 for which the number of analysis needed for the SAND approach is more than 6 times the number needed by the NAND approach. Thus the NAND approach appears to be more efficient. Obviously it is difficult to reach a general conclusion since the efficiency of the methods is strictly related to the test case to which these are applied.

6.4 Automatic Differentiation

For analysing the gain in efficiency that can be obtained by applying AD to an optimisation procedure, five simple test cases were considered. For all of them the functions are continuous and differentiable. The test cases considered are CRC_cvx (section 3.2.1), CRC_ccv (section 3.2.1), SPHR_ccv-Case1 (section 3.1) and the following two test cases.

SPHR_ND_ccv

$$\min_{\mathbf{x}} F(\mathbf{x})$$

where: $\sum_{i=1}^N x_i^2 \geq 1$

for: $0 \leq x_i \leq 1, i = 1, \dots, N$

where $F(\mathbf{x}) = \mathbf{x} = [x_1, \dots, x_N]^T$.

SPHR_3D_cvx_mltc

$$\min_{\mathbf{x}} F(\mathbf{x})$$

where: $\sum_{i=1}^N x_i^2 \geq 1$

and where: $x_i < 10^{-4}, i = 4, \dots, N$

for: $0 \leq x_i \leq 1, i = 1, \dots, N$

where $F(\mathbf{x}) = [x_1, x_2, x_3]^T$.

In particular for the two test cases the following cases were considered:

SPHR_8D_ccv-Case1

The test case is based on test case SPHR_ND_ccv for $N = 8$. The following *anchor points* are considered.

$$\begin{aligned} \mu_1^* &= \begin{bmatrix} 0.0000e+000 \\ 6.0059e-001 \\ 1.5727e-001 \\ 6.1536e-001 \\ 1.6768e-001 \\ 6.9435e-001 \\ 8.4516e-001 \\ 5.9886e-002 \end{bmatrix}, \quad \mu_2^* = \begin{bmatrix} 5.6243e-001 \\ 0.0000e+000 \\ 1.5727e-001 \\ 6.1536e-001 \\ 1.6768e-001 \\ 6.9435e-001 \\ 8.4516e-001 \\ 5.9886e-002 \end{bmatrix}, \quad \mu_3^* = \begin{bmatrix} 5.6243e-001 \\ 6.0059e-001 \\ 0.0000e+000 \\ 6.1536e-001 \\ 1.6768e-001 \\ 6.9435e-001 \\ 8.4516e-001 \\ 5.9886e-002 \end{bmatrix}, \quad \mu_4^* = \begin{bmatrix} 5.6243e-001 \\ 6.0059e-001 \\ 1.5727e-001 \\ 0.0000e+000 \\ 1.6768e-001 \\ 6.9435e-001 \\ 8.4516e-001 \\ 5.9886e-002 \end{bmatrix}, \\ \mu_5^* &= \begin{bmatrix} 5.6243e-001 \\ 6.0059e-001 \\ 1.5727e-001 \\ 6.1536e-001 \\ 0.0000e+000 \\ 6.9435e-001 \\ 8.4516e-001 \\ 5.9886e-002 \end{bmatrix}, \quad \mu_6^* = \begin{bmatrix} 5.6243e-001 \\ 6.0059e-001 \\ 1.5727e-001 \\ 6.1536e-001 \\ 1.6768e-001 \\ 0.0000e+000 \\ 8.4516e-001 \\ 5.9886e-002 \end{bmatrix}, \quad \mu_7^* = \begin{bmatrix} 5.6243e-001 \\ 6.0059e-001 \\ 1.5727e-001 \\ 6.1536e-001 \\ 1.6768e-001 \\ 6.9435e-001 \\ 0.0000e+000 \\ 5.9886e-002 \end{bmatrix} \text{ and } \mu_8^* = \begin{bmatrix} 5.6243e-001 \\ 6.0059e-001 \\ 1.5727e-001 \\ 6.1536e-001 \\ 1.6768e-001 \\ 6.9435e-001 \\ 8.4516e-001 \\ 0.0000e+000 \end{bmatrix} \end{aligned}$$

SPHR_3D_ccv_mlhc-Case1

The test case is based on test case SPHR_3D_ccv_mlhc for $N = 8$. The following *anchor points* are considered.

$$\mu_1^* = \begin{bmatrix} 0 \\ 0.976 \\ 1 \end{bmatrix}, \quad \mu_2^* = \begin{bmatrix} 0.982 \\ 0 \\ 0.936 \end{bmatrix} \text{ and } \mu_3^* = \begin{bmatrix} 1 \\ 0.921 \\ 0 \end{bmatrix}$$

As in section 6.3 the *anchor points* are fixed in order to perform the identical optimisations both with and without applying AD. The number of analyses required for obtaining the Pareto solutions without applying AD (i.e. using finite differences) and applying AD are summarised in Table 6.

	NC+ Number of analysis with no AD	NC+ Number of analysis with AD
CRC_cvx	202	176
CRC_ccv	137	121
SPHR_ccv-Case1	420	329
SPHR_8D_ccv	4601	995
SPHR_3D_cvx_mlrc	1390	301

Table 6: Comparison of the number of analyses required for solving five test cases with finite differences and with AD.

As expected, the results demonstrate the effectiveness of AD in minimising the number of analysis required for solving the optimisation problems.

6.5 USMAC

The USMAC test case was provided by a major airframe manufacturer, in the context of the European project VIVACE. USMAC stands for Ultra Simplified Model of Aircraft. The USMAC model is considered for testing the complete multiobjective multidisciplinary algorithm CODE-M proposed in this research. The test case contains 96 models and 126 variables.

The 96 models were arranged together in order to generate one unique system with 17 input variables and 6 output variables. The input and output variables as well as the optimisation problem are described below.

Input variables:

Input Variables	Units
Npax (number passengers)	
NpaxFront (number of passengers per row)	
Naisle (number of aisles)	
FNslst (engine thrust)	decaN
BPR (engine bypass ratio)	
ne (number of engines)	
Awing (wing area)	m ²
span (wing span)	m
phi (wing sweep angle)	°
tuc (wing thickness to chord ratio)	
RA (range)	NM
MTOW (maximum take-off weight)	kg
Fuel (fuel quantity)	kg
altcrz (cruise altitude)	ft
Machcrz (cruise mach)	
alto (takeoff altitude)	ft
altapp (approach altitude)	ft

Output variables:

Output Variables	Units
MTOW (maximum take-off weight) kg	kg
RA (range) NM	NM
RAtime (flight time)	h
tofl (takeoff field length)	m
vapp (approach speed)	kts
vzclb (climb rate)	ft/min
kfncth (cruise thrust coefficient)	
Kff (wing fuselage fuel ratio)	

As can be seen from the input and output variables, the maximum take off weight (MTOW) and the range (RA) appear both in the inputs and the outputs, making the system non-hierarchically decomposable.

The multi-objective optimisation problem is set as follows:

- The independent variables considered were:

Independent Variables
FNslst = [12700,12800] (decaN)
Awing = [154,155] m ²
span = [32,36] m
phi = [29,31] °
tuc = [0.8,0.9]
Fuel = [17000,17500] kg

- The constant variables considered were:

Constant Variables
Npax = 150
NpaxFront = 6
Naisle = 1
BPR = 6
ne = 2
alterz = 35000 ft
Machcrz = 0.82
altto = 0 ft
altapp = 0 ft

- Objectives considered:

Objectives
range (RA) to be maximised
(MTOW) to be minimised
(vzclb) to be maximised

- Constraints considered:

Constraints
$\text{tofl} \leq 2000 \text{ m}$
$\text{vapp} \leq 120 \text{ kts}$
$\text{vzclb} \geq 500 \text{ ft/min}$
$\text{kfncth} \leq 1$
$\text{Kff} \geq 0.75$

- Coupling variables:

Coupling variables
range (RA) to be maximised
(MTOW) to be minimised

For the USMAC test case the number of starting points L considered was 25, while the number of *utopia plane points* K was 40. The CODE-M algorithm described in section 5.1 was applied with the NAND approach and AD in order to obtain the results shown from Figure 67 to Figure 74. 496915 analyses were required to determine the solution, 62 Pareto points were obtained. In all figures the Pareto points are numbered in order to be distinguishable across the figures.

Figure 67 and Figure 68 are two representations of the objective functions. Figure 67 shows the three dimensional criterion space. A green surface, linking the obtained global Pareto points, was added in order to give a better visualisation of the global Pareto front. The bar chart of Figure 68 shows the values of MTOW, RA and vzclb for the Pareto points. The three anchor points correspond to Pareto points 2, 5

and 29, where for point 5 MTOW is a minimum, for point 2 the range RA is a maximum, while the climb rate vzclb is maximised for point 29.

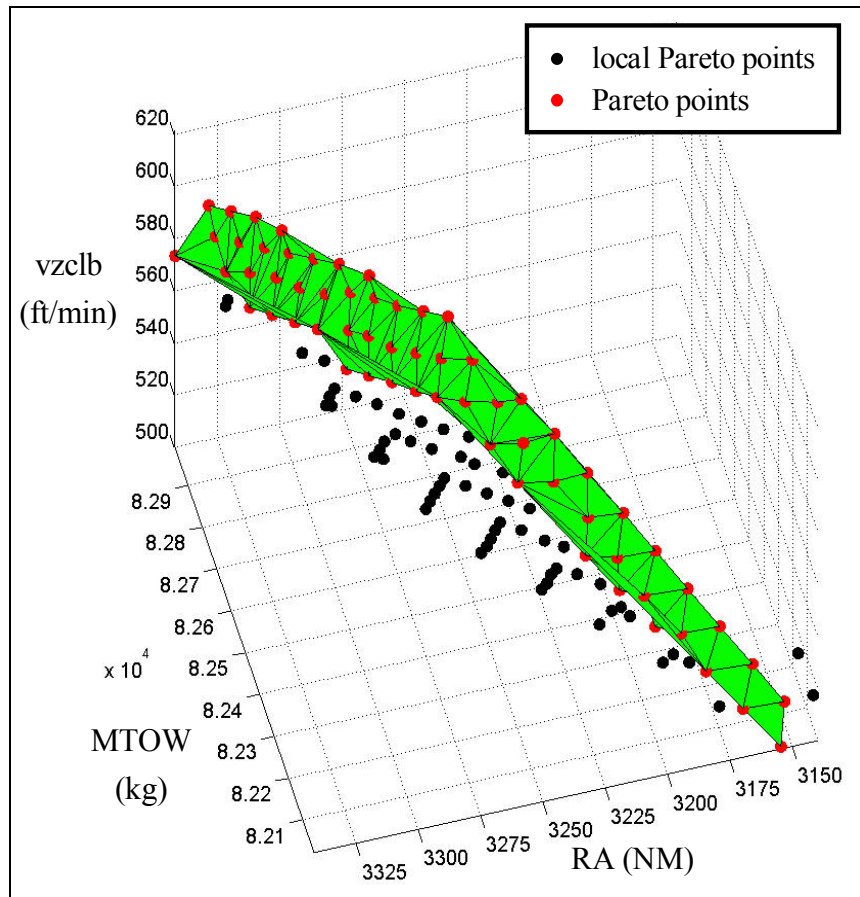


Figure 67: Criterion Space

Figure 69 shows the 2 dimensional cross representations of the global Pareto points obtained for MTOW with respect to RA, MTOW with respect to vzclb and for RA with respect to vzclb. It is interesting to notice how both RA and vzclb increase at almost a constant ratio with the takeoff weight MTOW, approximately up to a value of MTOW and RA of 82350 kg and 3250 NM, respectively. For greater values of MTOW the rate of increase of RA drops and the range of vzclb for similar values of MTOW broadens. By taking two points from this region e.g. points 21 and 44, for which the values can be seen in Figure 70, it is interesting to notice that for an almost identical value of MTOW of 82622 the difference in range between point 21 and 44 is of approximately 9 NM while the difference in vzclb is of almost 33 ft/min. It is information as this which is of interest to the designer.

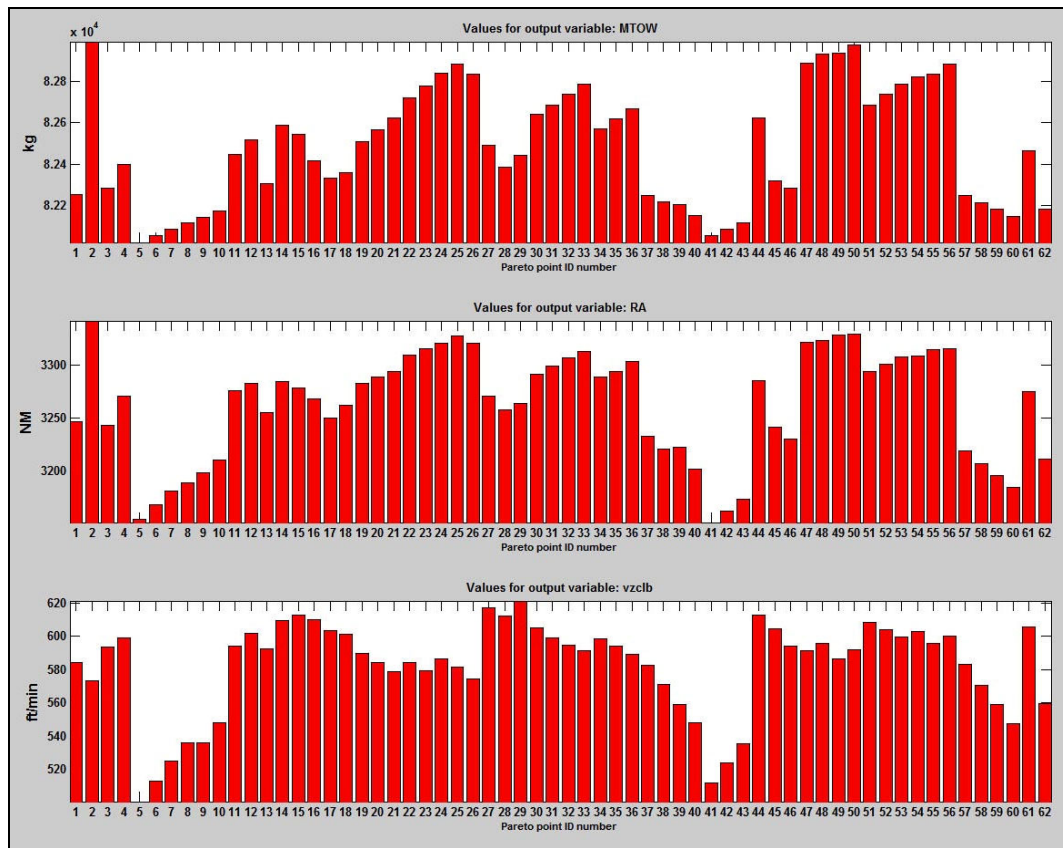


Figure 68: USMAC Objectives

Figure 71, Figure 72 and Figure 73 show the values for all of the independent variables for the Pareto points obtained with respect to MTOW, RA and vzclb, respectively. Once again it is interesting to notice the patterns of variation of the design variables in correspondence to the point for which MTOW and RA have values 82350 kg and 3250 NM, respectively. From the figures it can be observed that another particularly interesting point can be identified, the point for which MTOW, RA and vzclb have approximately values 82450 kg, 3265 NM and 620 ft/min, respectively, in correspondence to point 29. Point 29 is the *anchor point* for objective vzclb.

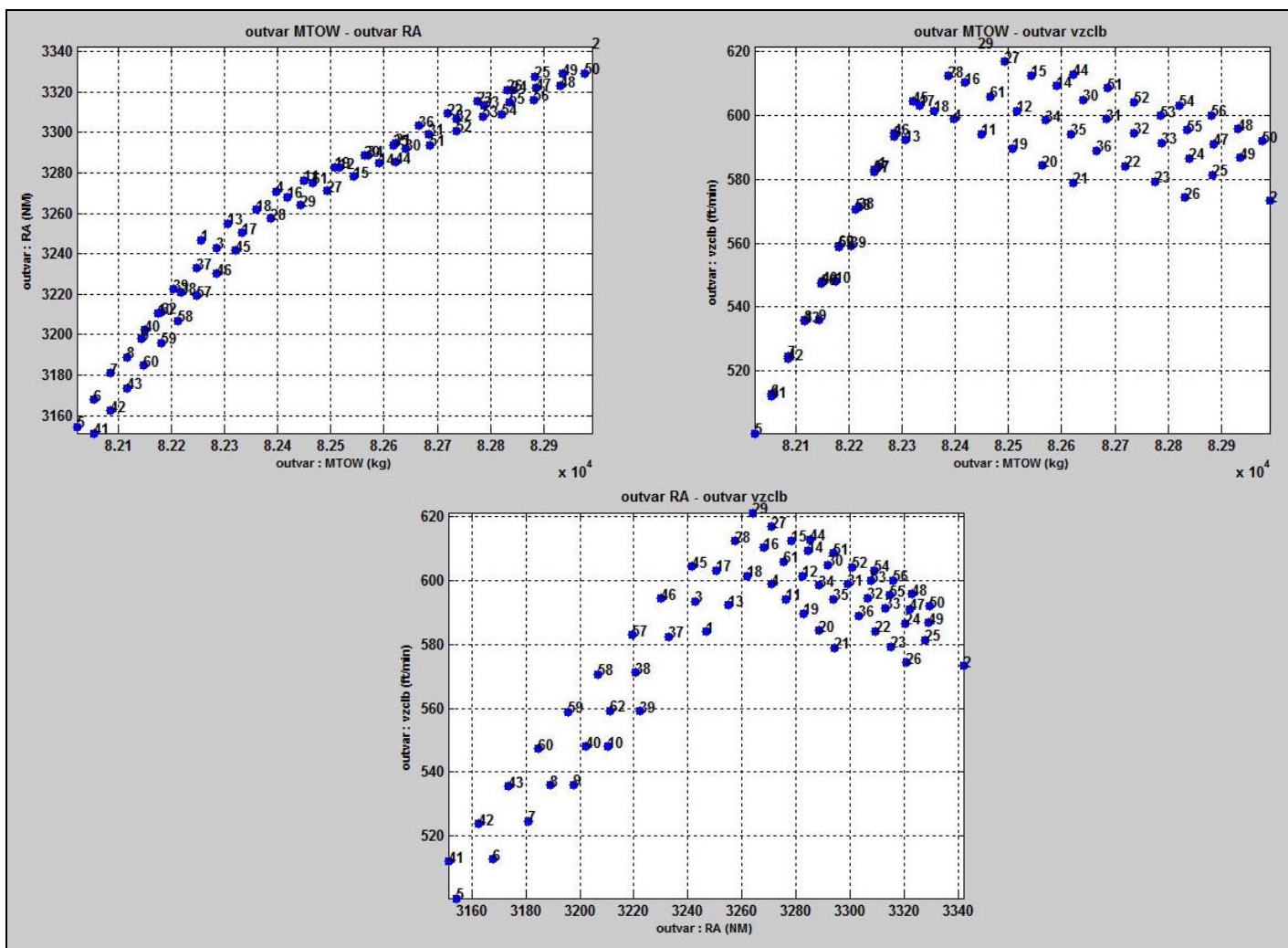


Figure 69: USMAC Objectives 2D

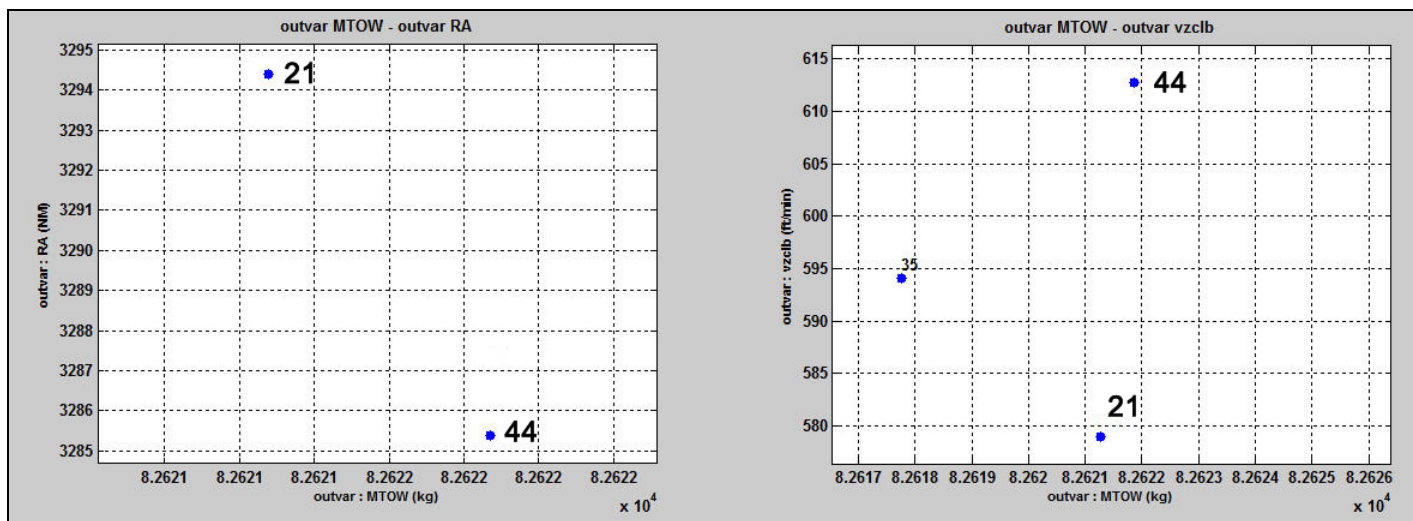


Figure 70: Detail of the value of the objectives for Pareto points 21 and 44.

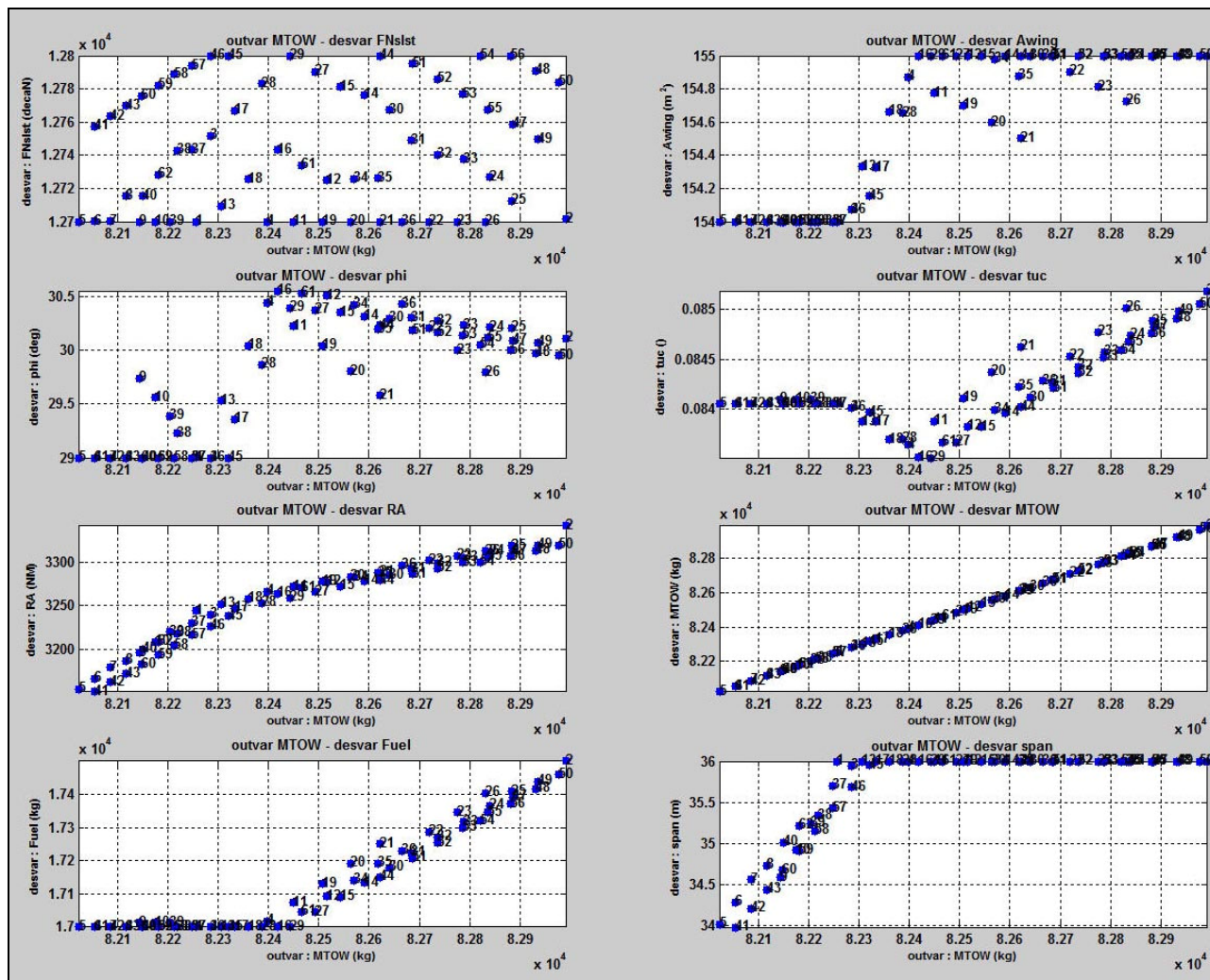


Figure 71: USMAC MTOW/desvar

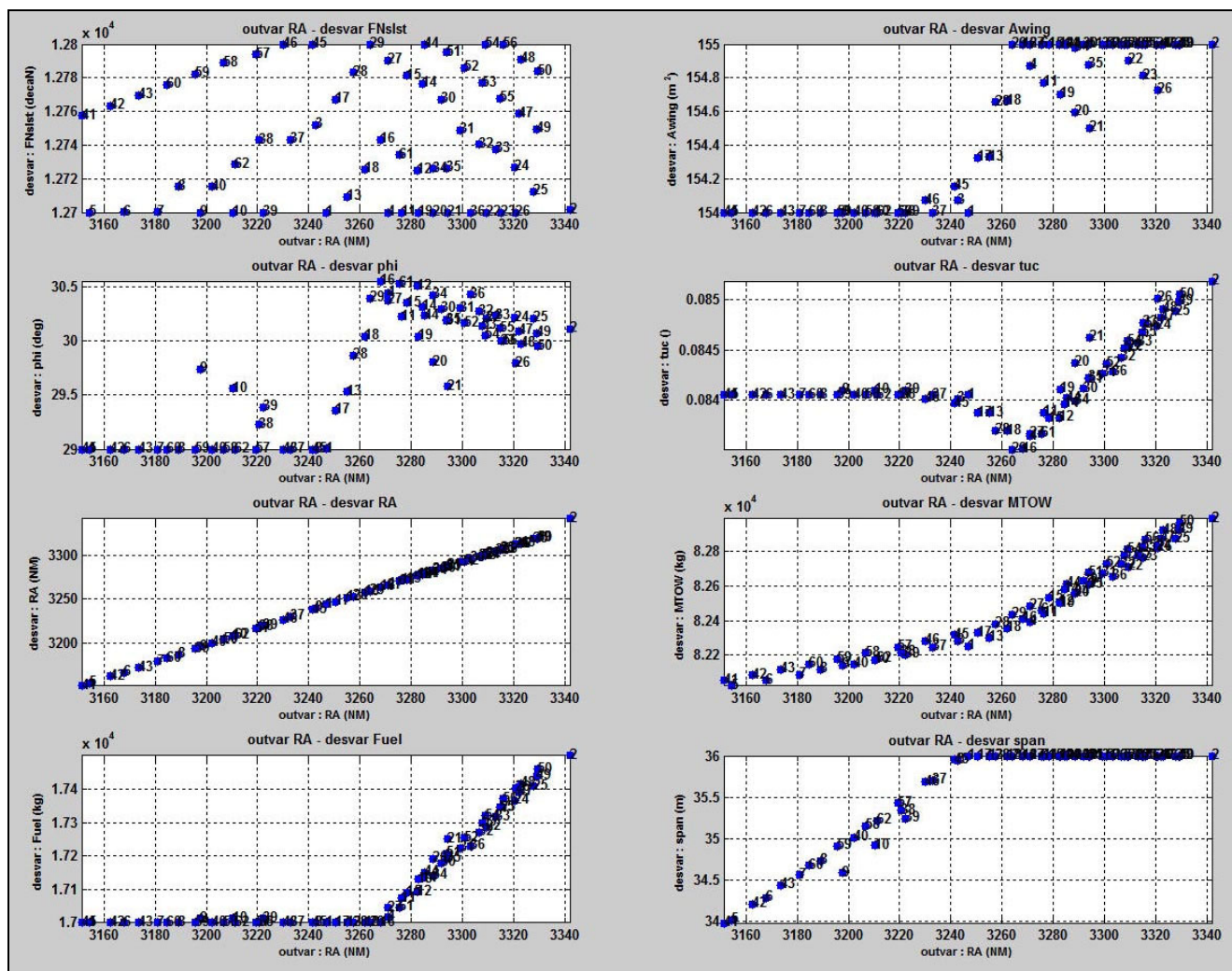


Figure 72: USMAC RA/desvar

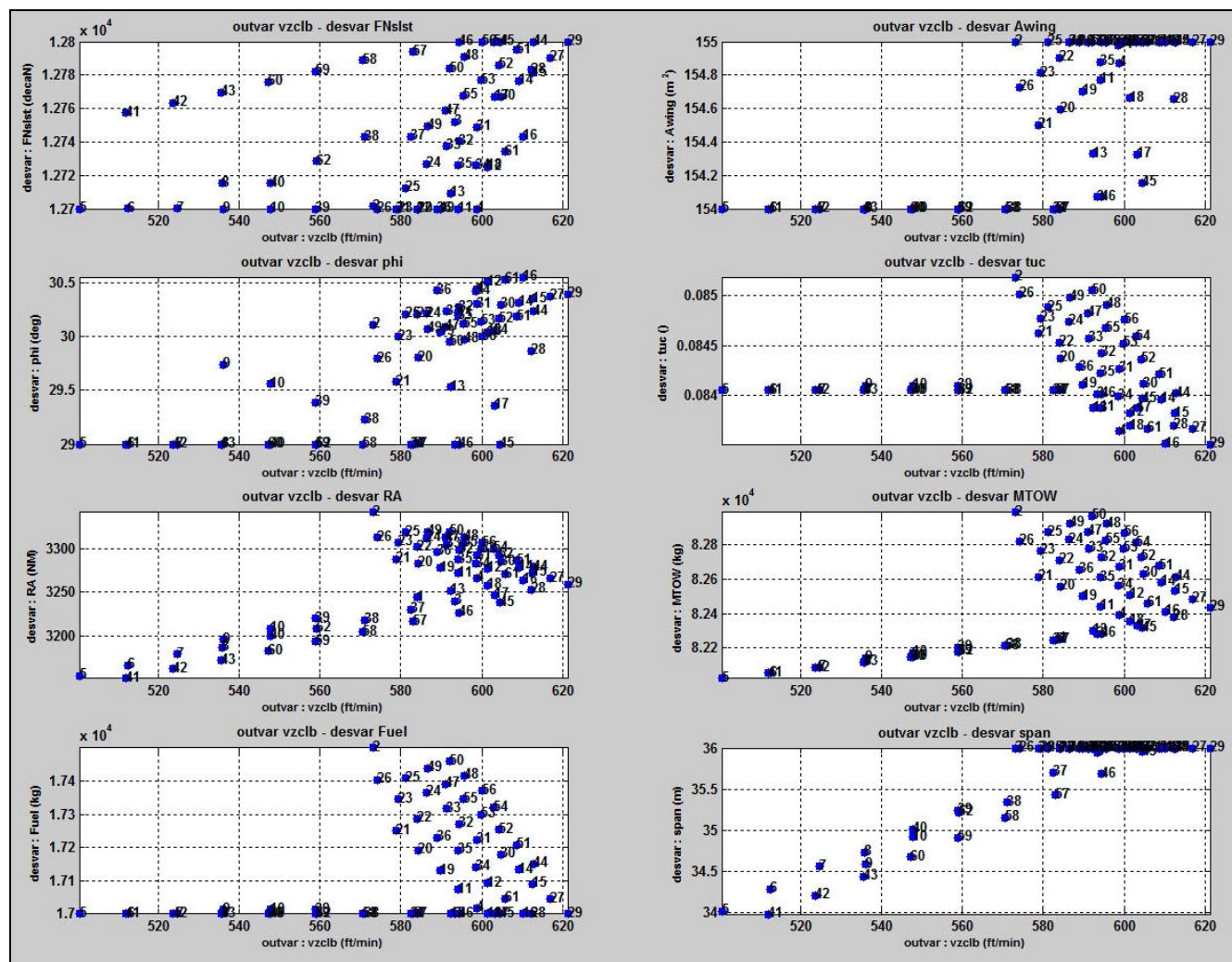


Figure 73: USMAC vzclb/desvar

Figure 74 shows the values of the constraints for all Pareto points. From the figure it can be seen that constraint $\text{tofl} \leq 2000$ m is never active. Constraint $\text{vapp} \leq 120$ kts is active for points 1-4, 9-36, 38, 39, 44-56 and 61. Constraint $\text{vzclb} \geq 500$ ft/min is active only for point 5. Constraint $\text{kfncth} \leq 1$ is active for point 5. Finally constraint $\text{Kff} \geq 0.75$ is active for all Pareto points. It is to be noted that the values of Kff for the Pareto points are all approximately the same and are equal to 0.75. The reason for which point 11 has apparently a higher value is simply due to the normalisation of the values of vzclb with respect to the maximum and minimum values of vzclb obtained for all Pareto points. This implies that constraint $\text{Kff} \geq 0.75$ is active for all the Pareto points. Thus, for the *anchor point* corresponding to the minimisation of MTOW (point 5) the constraints on vzclb , kfncth and Kff are active; for the *anchor point* relative to the maximisation of RA (point 2) the constraints on vapp and Kff are active; while for the *anchor point* relative to vzclb (point 29), the active constraints are on vapp and Kff . Finally it can be observed that the two main constraints which limit the solution are those on vapp and Kff .

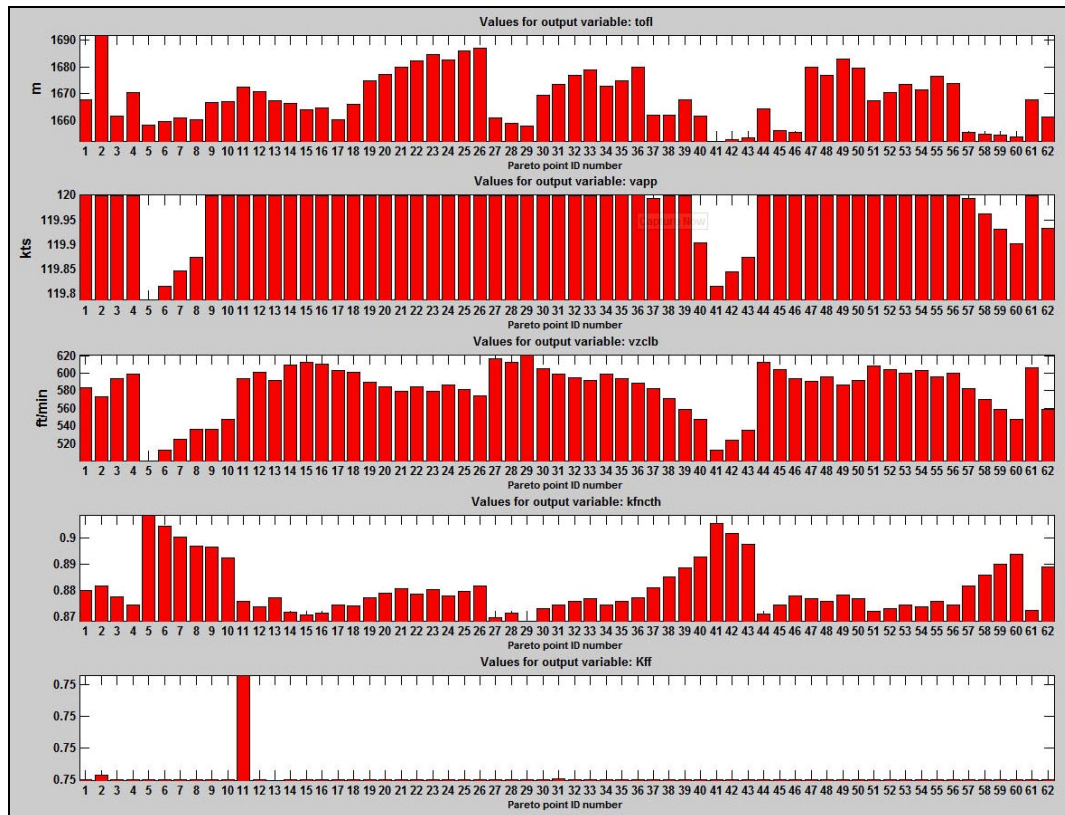


Figure 74: USMAC Constraints values

From the figures it can be seen that when MTOW is at its minimum (Figure 71) the aircraft is the smallest possible (Awing and span are at their minimum), the Fuel is the least amount allowed and the thrust FNs1st is also at its minimum; as a result RA is at its minimum as well as vzclb. For greater values of RA and vzclb, for values of MTOW smaller than 82300 kg, the design variables that vary the most are span and FNs1st, while Awing and Fuel are constant and are at their minimum. The increase in span and FNs1st produces an increase in vzclb and RA while increasing the MTOW. When MTOW reaches 82300 kg the span reaches its maximum. For greater values of MTOW, up to 82450 kg, Awing starts increasing, while the Fuel remains constant (Figure 71). When MTOW reaches 82450 kg Fuel is still at its minimum, span, Awing and FNs1st are at their maximum and the anchor point for maximum vzclb is reached (Figure 71, Figure 73). For values of MTOW greater than 82450 kg Fuel becomes the predominant variable, determining all of the remaining Pareto points. It can be noticed, in fact, that for values of MTOW greater than 82450 kg Fuel starts increasing. Furthermore, the total variation in Fuel is of 500 kg, while the variation of MTOW from 82450 kg to its maximum is of approximately 550kg. While such increase in weight produces a reduction of vzclb (Figure 73), it leads to an increase in RA. Finally, RA reaches its maximum when also Fuel is at its maximum (Figure 72).

As shown from the results an important amount of information can be obtained when applying the proposed multiobjective multidisciplinary algorithm CODE-M, it is then up to the Decision Maker to formulate a choice.

6.6 Summary

In this chapter the results for a number of test cases are given. Initially the proposed NC+ method was applied to a number of test cases demonstrating its capability of overcoming the problems encountered by the NC method. Subsequently, the NC+ method together with the global optimisation procedure were applied to a number of global multiobjective optimisation problems, in order to demonstrate the unique capability of the procedure to generate solutions representative both of the global and local Pareto fronts. It must be noted that, to the author's knowledge, for the NBI, PP-based and NC method no attempt at solving problems of this level of complexity has ever been attempted. A number of simple test cases for evaluating the

computational cost of the NAND and SAND approaches in conjunction with the NC+ method were performed, showing, that at least for the test cases considered, the NAND approach performed better. A number of simple test cases were then used for evaluating the performance of AD. Finally the proposed CODE-M algorithm was applied to the USMAC test case, a simplified aircraft model. The results obtained demonstrating the capability of the algorithm of solving a multiobjective multidisciplinary optimisation problem for aircraft conceptual design.

Although the approach was successfully applied to all the test cases considered, one main limitation of the CODE-M algorithm has to be acknowledged. The optimal value of the number of starting points required for the execution of the algorithm can not be known without a prior knowledge of the problem to be solved. A too small number of starting points could lead the algorithm to fail in identifying all of the local minima; this, potentially leading to the deterioration of its ability to obtain a Pareto set representative of all local and eventually also the global Pareto fronts. On the contrary, a too large number could lead to a large number of nugatory optimisations, thus increasing the computational cost. Nevertheless not having any adequate information relative to the optimisation problem to which the algorithm has to be applied, it is preferable to exceed the number of starting points in order to avoid the latter case.

7 Summary, Conclusions and Future Work

After an initial investigation of the aircraft design process at the conceptual phase, it has been possible to identify the computational difficulties faced by the designer. The conceptual design phase is peculiar in that the contributing analyses, which are needed in order to model the behaviour of the aircraft, are in fact hundreds of generally simple models which are associated with thousands of variables. For performing trade-off analysis the contributing analyses need to be assembled according to the particular design study to be performed; this eventually leads to a non-hierarchically decomposable arrangement due to complex coupling exist among the contributing analysis. Through discussion with industry it has become apparent that optimisation tools are seldom used, even though they could greatly enhance the work of the designers. The job of the designer, in fact, corresponds to implicitly solving a multiobjective multidisciplinary optimisation problem.

In order to effectively tackle the conceptual design computational problem, in its optimisation aspects, it was necessary to identify among the available technologies those considered most suitable. Among the methods and techniques that were considered of interest, multiobjective optimisation algorithms which make use of the reformulation approach, multidisciplinary design optimisation algorithms and automatic differentiation were identified as appropriate tools.

An investigation of different reformulation techniques for multiobjective optimisation and the identification of their limitations, lead to the development of the modified PP-based method, then to the DHCBI and finally to the development of the NC+ method. This sequence reflects the efforts of the author to stay abreast with the dynamic developments in this field. A novel procedure for global multiobjective optimisation was proposed as part of these efforts. It utilises multiobjective optimisation methods which make use of the reformulation approach, allowing their application to real life optimisation problems. The procedure has two unique capabilities; the first is that of allowing the application of the NC+ method to global optimisation problems. The second feature, making this approach unique amongst all

current multiobjective optimisation methods, is the ability to generate well-distributed Pareto points on both local and global Pareto fronts simultaneously.

Since aircraft conceptual design is inherently multidisciplinary, where potentially the system of contributing analyses can be non-hierarchically decomposable, multidisciplinary optimisation algorithms were considered. Given the amount of contributing analysis to be taken into account, even in the case of a hierarchically decomposable system arranging the disciplines becomes a challenge. Thus the application of the more complex multidisciplinary optimisation algorithms such as CO, CSSO and BLISS were considered impractical at this stage. Therefore, for the current research the two more common and simpler NAND and SAND algorithms were considered.

For improving the efficiency of the complete algorithm Automatic Differentiation was applied as an effective means for improving the computational cost of performing multiobjective multidisciplinary optimisation with gradient based optimisation algorithms. Since Automatic Differentiation had to be applied to non-hierarchically decomposable systems, a formulation of the Global Sensitivity Equations tailored for the problem to be solved was presented.

The NC+ method, the global optimisation procedure, the NAND and SAND algorithms, Automatic Differentiation, and the Global Sensitivity Equations were integrated in the proposed CODE-M algorithm for global multiobjective multidisciplinary optimisation.

The algorithm was implemented in a MATLAB code which was used for:

- testing the capability of the NC+ method,
- testing the global optimisation procedure in order to generate global and local Pareto fronts,
- comparing the efficiency of NAND and SAND approaches when applied in conjunction with the NC+ method,
- evaluating the efficiency of automatic differentiation.

The results obtained demonstrate that the NC+ method solves the limitations of the existing multiobjective optimisation methods based on the reformulation approach. Furthermore, the unique ability of the proposed global multiobjective

multidisciplinary algorithm in obtaining well-distributed Pareto points, on both the global and local Pareto fronts, is demonstrated.

The main limitation of the proposed approach is related to the choice of the parameter specifying to the number of starting points required for the execution of the algorithm. In fact, its optimal value can not be known without a prior knowledge of the problem to be solved. A too small number of starting points could lead the algorithm to fail in identifying all of the local minima, potentially leading to the deterioration of its ability to obtain a Pareto set representative of all local and eventually also the global Pareto fronts. On the contrary, a too large number could lead to a large number of nugatory optimisations, thus increasing the computational cost.

As part of the test and evaluation process the proposed algorithm was applied to the USMAC test case, a simplified aircraft model provided as test case by a major airframe company in the context of the European project VIVACE. The test result showed that it was possible to obtain the Pareto front as required. It was also shown that useful information for the designer can be obtained through analysis of the solution data. Furthermore, the industrial partner, provider of the USMAC test case, gave a very positive feedback on the prototype tool developed by the author.

Currently the effectiveness of the developed algorithm is restricted to those problems for which the constituent functions are smooth and the variables are continuous. However, it would be of interest as part of a future research effort, to extend the algorithm's applicability to problems where discontinuities and discrete variables are present. For this purpose the integration of evolutionary or hybrid optimisation algorithms could be an interesting option to be investigated. Also it would be of interest to develop a procedure capable of arranging automatically the contributing analyses, which would allow the application and integration of the more complex but more efficient MDO algorithms, such as CO, CSSO and Bliss, in the proposed approach.

8 References

- AIAA Technical Committee on Multidisciplinary Design Optimization (MDO) (1991). *Current State of the Art in Multidisciplinary Design Optimization*. MDO Technical Committee, White Paper, AIAA, Reston, VA, January 1991
- Athan, T.W., and Papalambros, P.Y. (1996). A Note on Weighted Criteria Methods for Compromise Solutions in Multi-Objective Optimization. *Engineering Optimisation*, 27, pp. 155-176, 1996.
- Balachandran, L.K., Fantini, P, and Guenov, M.D. (2007). Computational Process Management for Aircraft Conceptual Design. Accepted for presentation at the *7th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, 18 - 20 Sep 2007, Belfast, Northern Ireland.
- Balling, R.J. and Sobieszczanski-Sobieski, J. (1994). Optimization of Coupled Systems: A Critical Overview of Approaches. In *Proceedings of The AIAA/NASA/ USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City, FL, September 7-9, 1994.
- Balling, R.J. (2000), Pareto Sets in Decision-Based Design, *Engineering Valuation and Cost Analysis*, Vol. 3, pp. 189-198.
- Barthelemy, J.F. and Hall, L.E. (1992). Automatic Differentiation as a Tool in Engineering Design, AIAA-1992/4743. *Proceedings of the 4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, September 21-23, 1992, Cleveland, OH.
- Bischof, C. and Griewank, A. (1994). Computational differentiation and multidisciplinary design. *Inverse Problems and Optimal Design in Industry*, H. Engl and J. McLaughlin, eds., Stuttgart, 1994, Teubner Verlag, pp. 187-211.
- Cramer, J. E., Dennis, J. E. Jr., Frank, P.D., Lewis, R.M. and Shubin, G.R. (2000). Problem Formulation for Multidisciplinary Optimization. *SIAM Journal of Optimization*, no. 4 (November 1994): 754-776.
- Das, I. and Dennis, J.E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimisation problems. *Structural Optimisation*, vol. 14 , Number 1 / August, 1997, pp. 63-69.
- Das, I. and Dennis, J.E. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal of Optimization*, vol. 8, pp. 631-657, 1998.

Das, I. (1999). An Improved Technique for Choosing Parameters for Pareto Surface Generation Using Normal-Boundary Intersection. Proceedings of the *Third World Congress of Structural and Multidisciplinary Optimization WCSMO-3*, Buffalo, NY, March, 1999.

Deb, K. (1998). *Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems* [online]. Technical Report CI-49/98, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, 1998. Available from: <http://citeseer.ist.psu.edu/deb99multiobjective.html> [Accessed 22 July 2007].

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp 182-197.

Fantini, P, Balachandran, L.K. and Guenov, M.D. (2007). Computational Intelligence in Multi Disciplinary Optimization at Conceptual Design Stage. Proceedings of the First International Conference on Multidisciplinary Design Optimization and Applications April 17-20, 2007 Besancon, France.

Fletcher, R. (1981). *Practical Methods of Optimization: Vol. 2: Constrained Optimization*. John Wiley & Sons, Inc., Somerset, N.J. 08873 , 1981, 224 pp

Forth, S. A. (2006). An Efficient Overloaded Implementation of Forward Mode Automatic Differentiation in MATLAB. *ACM Transactions on Mathematical Software*, Vol. 32, No.2, 2006, P195-222.

Forth S.A. and Edvall, M.M., (2007). User Guide for Mad - a Matlab Automatic Differentiation Toolbox TOMLAB /MAD Version 1.4 [online]. Available from: http://tomlab.biz/docs/TOMLAB_MAD.pdf [Accessed 22 July 2007].

Frank, P.D., Booker, A.J., Caudel, T.P. and Healy M.J. (1992). A Comparison of Optimization and Search Methods for Multidisciplinary Design. AIAA-1992-4827, Proceedings of the *4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, September 21-23, 1992, Cleveland, OH.

Kodiyalam, S. (1998). *Evaluation of methods for multidisciplinary design optimization, Phase I* [online]. Technical Report NASA/CR-1998-208716, National Aeronautics and Space Administration, September 1998. Available from: <http://citeseer.ist.psu.edu/kodiyalam98evaluation.html> [Accessed 22 July 2007].

Kodiyalam, S. and Yuan, C. (2000). *Evaluation of Methods for Multidisciplinary Design Optimization (MDO), Part II* [online]. NASA/CR-2000-210313, November 2000, pp. 51. Available from: <http://citeseer.ist.psu.edu/kodiyalam98evaluation.html> [Accessed 22 July 2007].

Kodiyalam, S., Sobieszczanski-Sobieski, J. (2001). Multidisciplinary Design Optimization - Some Formal Methods, Framework Requirements, and Application to Vehicle Design. *International Journal of Vehicle Design* (Special Issue), pp. 3-22.

Koski, J. (1985). Defectiveness of Weighting Method in Multicriterion Optimization of Structures. *Communications in Applied Numerical Methods*, 1, pp. 333-337, 1985.

Kroo, I., Altus S., Braun R., Gage P., Sobieski I. (1994), Multidisciplinary Optimisation Methods for Aircraft Preliminary Design. AIAA-94-4325-CP, Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, pp.697-707, Panama City, Florida, September 7-9.

Kursawe, F. (1991). A Variant of Evolution Strategies for Vector Optimization. In H.- P. Schwefel and R. Manner, *Parallel Problem Solving from Nature*, pages 193--197. Springer, Berlin and Heidelberg, 1991.

Marler R.T., Arora J.S. (2004). *Survey of multi-objective optimization methods for engineering*. Journal of *Structural and Multidisciplinary Optimization*, Volume 26, Number 6 / April, 2004, pages 369-395, Springer, Berlin and Heidelberg, 2004.

Mattison, C.A., Mullur, A.A., and Messac, A. (2002). Minimal Representation of Multiobjective Design Space Using Smart Pareto Filter. Proceeding of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation, AIAA-2002-5458, Atlanta, GA, September, 2002.

Messac, A. (1996). Physical Programming: Effective Optimization for Computational Design. *AIAA Journal*, Vol. 34, No. 1, Jan. 1996, pp. 149-158, 1996.

Messac, A. (2000). From Dubious Construction of Objective Functions to the Application of Physical Programming. *AIAA Journal*, Vol. 38, No. 1, pp. 155-163, 2000.

Messac A., Mattson C.A. (2002). Generating Well-Distributed Sets of Pareto Points for Engineering Design using Physical Programming. *Optimisation and Engineering*, Kluwer Publishers, Vol. 3, Issue 4, December 2002, pp. 431-450.

Messac, A., Ismail-Yahaya, A., and Mattson, C.A. (2003). The Normalized Normal Constraint Method for Generating the Pareto Frontier. *Structural and Multidisciplinary Optimization*,

Journal of the International Society of Structural and Multidisciplinary Optimization (ISSMO), Springer, Vol. 25, No. 2, 2003, pp. 86-98.

Messac, A., and Mattson, C. (2004). Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier. AIAA Journal, 2004, Vol. 42, No. 10, pp. 2101-2111, 2004.

Miettinen, K.M. (1999). *Nonlinear Multiobjective Optimisation*. Kluwer Academic, Boston, 1999, 298p.

Palli N., Azarm S., McCluskey P. (1998). An interactive Multistage ϵ -Inequality Constraint Method For Multiple Objective Decision Making., *ASME Journal of Mechanical Design*, Vol. 120, December 1998, pp. 678-686.

Perez, R., Liu, H. and Behdinan, K. (2004). Evaluation of Multidisciplinary Optimization Approaches for Aircraft Conceptual Design. AIAA-2004-4537, *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, Aug. 30-1, 2004

Shukla, P. K., Deb, K. and Tiwari, S. (2005). Comparing classical generating methods with an evolutionary multi-objective optimization method. Proceedings of the *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO-2005)*. Guanajuato, Mexico. Lecture Notes on Computer Science 3410, pages 311- 325.

Sobieszczanski-Sobieski, J. (1988). Optimization by decomposition: a step from hierarchic to non-hierarchic systems. Presented at the *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, Hampton, VA, NASA CP-3031, Part 1. Also NASA TM-101494 1988.

Sobieszczanski-Sobieski J. (1990). Sensitivity of Complex, Internally Coupled Systems. *AIAA Journal*, Vol. 28 No. 1, 153-160, 1990.

Sobieszczanski-Sobieski, J. and Haftka, R. T. (1996). Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments. *34th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, AIAA Paper No. 96-0711, January 15-18, 1996, pp. 32

Sobieszczanski-Sobieski, J., Agte, J. S. and Sandusky, R. R. Jr. (1998). Bi-Level Integrated System Synthesis (BLISS), NASA/TM-1998-208715, August 1998, pp. 42, also published as AIAA 98-4916.

The MathWorks, Inc., (2007), Optimization Toolbox 3 User's Guide.

The MathWorks, Inc., (2007), Statistics Toolbox 6 User's Guide.

Utyuzhnikov, S.V., Fantini, P., Guenov, M.D. (2005a). Numerical Method for Generating the Entire Pareto Frontier in Multiobjective Optimisation. Proceedings of *EUROGEN 2005*, 12-14 September 2005, Munich, Germany.

Utyuzhnikov, S.V., Guenov, M.D., Fantini, P. (2005b). Application of the Modified Physical Programming Method to Generating the Entire Pareto Frontier in Multiobjective Optimisation. Proceedings of *EUROGEN 2005*, 12-14 September 2005, Munich, Germany.

Van Veldhuizen, D.A., Lamont, G.B. (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Thesis (PhD), Air Force Institute of Technology Wright-Patterson AFB OH School of Engineering.

9 Publications of the author

9.1 Conference Papers

Balachandran, L.K., Fantini, P, and Guenov, M.D. (2007). Computational Process Management for Aircraft Conceptual Design. Accepted for presentation at the *7th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, 18 - 20 Sep 2007, Belfast, Northern Ireland

Fantini, P, Balachandran, L.K. and Guenov, M.D. (2007). Computational Intelligence in Multi Disciplinary Optimization at Conceptual Design Stage. Presented at the *First International Conference on Multidisciplinary Design Optimization and Applications*, April 17-20, 2007 Besancon, France.

Maginot J., Guenov, M.D. and Fantini, P (2007). A Method for Adaptability Study of Pareto Solutions in Multi-Objective Optimization. Accepted for presentation at the *7th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, 18 - 20 Sep 2007, Belfast, Northern Ireland

Utyuzhnikov, S.V., Fantini, P., Guenov, M.D. (2005). Numerical Method for Generating the Entire Pareto Frontier in Multiobjective Optimisation. Proceedings of *EUROGEN 2005*, 12-14 September 2005, Munich, Germany.

Utyuzhnikov, S.V., Guenov, M.D., Fantini, P. (2005). Application of the Modified Physical Programming Method to Generating the Entire Pareto Frontier in Multiobjective Optimisation. Proceedings of *EUROGEN 2005*, 12-14 September 2005, Munich, Germany.

9.2 Journal publications

Fantini, P, Balachandran, L.K. and Guenov, M.D. (2007). Computational Intelligence in Multi Disciplinary Optimization at Conceptual Design Stage. Submitted for publication.

Utyuzhnikov, S.V., P. Fantini, P. and Guenov, M.D. (2007). A Method for Generating a Well-Distributed Pareto Set in Nonlinear Multiobjective Optimization. Submitted for publication.

10 Bibliography

10.1 Approximations

Alexandrov N.M., Dennis J.E. Jr., Lewis R.M., Torczon V., (1998). A Trust Region Framework For Managing The Use Of Approximation Models In Optimisation. ICASE Report No. 97-50, 1997.

Altus T. (2002). A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS). NASA/CR-2002-211652, May 2002, pp. 90.

Baker C.A., Grossman B., Haftka R.T., Mason W.H., Watson L.T., (1998). HSCT Configuration Design Space Exploration Using Aerodynamic Respons Surface Approximation. AIAA98-4803.

Golovidov O., Kodiyalam S., Marineau P., (1998). Flexible Implementation of Approximation Concepts in an MDO Framework. AIAA 98-4959.

Hagan M.T., Demuth H.B., Beale M. (1996). Neural network design. PWS Publishing Comp., Boston, MA.

Kodiyalam S., Lin J.S., Wujek B.A., Design of Experiments Based Response Surface Models for Design Optimisation. Engineous Software Technical Paper.

Rodriguez J.F., Pérez V.M., Renaud J.E., (2000). Sequential Approximate Optimisation Using Variable Fidelity Response Surface Approximations. AIAA-2000-1391.

Simpson T., Mauery T., Korte J., Farrokh Mistree (1998). Comparison of Response Surface and Kriging Models in the Multidisciplinary Design of an Aerospike Nozzle. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimisation, St. Louis, Missouri, AIAA 98-4755, September 2-4, 1998.

Simpson T.W., Booker A.J., Ghosh D., Giunta A.A., Koch P.N., Yang R.-J., Approximation Methods in Multidisciplinary Analysis and Optimisation: A Panel Discussion. <http://www.me.psu.edu/simpson/approximation/approx-panel.html>

Yang B.S., Yeun Y.-S., RuyW.-S., (2002). Managing Approximation Models in Multiobjective Optimisation. Struct Multidisc Optim 24, 141–156 Springer-Verlag 2002, Digital Object Identifier (DOI) 10.1007/s00158-002-0224-0.

10.2 Sensitivity Analysis

Frey H.C., Patil S.R., Identification and Review of Sensitivity Analysis Methods. Civil Engineering Department North Carolina State University Raleigh, NC. (Not Published)

Kleijnen J.P.C., (1995). Sensitivity Analysis and Related Analyses: a Survey of Statistical Techniques. International Symposium SAMO95 Theory and applications of Sensitivity Analysis of Model Output in computer simulation 25-27 September 1995 Belgirate, Italy.

10.3 Multidisciplinary Design and Optimisation (MDO)

Agarwal H., John E. Renaud J.E., Preston E.L., (2003). Trust Region Managed Reliability Based Design Optimisation Using Evidence Theory. AIAA-2003-1779.

AIAA Technical Committee on Multidisciplinary Design Optimisation (MDO) White Paper on Current State of the Art. January 15, 1991.

Alexandrov N.M., Kodiyalam S., (1998). Initial Results of an MDO Method Evaluation Study. AIAA-98-4884.

Alexandrov N.M., Michael Lewis R.M., (2003). Addressing the Challenges of Simulation-Based Design Optimisation. SIAM Conference on Computational Science and Engineering 8-13 February 2003.

Besnard E., Cordier-Lallouet N., Kural O., Chen H.P., (1999). DESIGN/OPTIMISATION WITH ADVANCED SIMULATED ANNEALING. AIAA 99-0186.

Biedron R.T., Mehrotra P., Nelson M.L., Preston F.S., Rehder J.J., Rogers J.L., Rudy D.H., Sobieski J., Storaasli O.O., (1999). Compute as Fast as the Engineers Can Think!. NASA/TM-1999-209715.

Fox R.L. and Schmit L.A. (1966). Advances in Integrated Approach to Structural Synthesis. Journal of Spacecraft Rockets, Vol 3, pp. 858-866.

Giesing J.P., Barthelemy J.-F., (1998). A Summary of Industry MDO Applications and Needs. The Boeing Company, Long Beach, CA and NASA/Langley Research Center, VA.

Giunta A.A., Sobieszczanski-Sobieski J., (1998). Progress Toward Using Sensitivity Derivatives in a High-Fidelity Aeroelastic Analysis of a Supersonic Transport. NASA-AIAA-98-4763.

Jussien N., Lhomme O., Dynamic Backtracking with Constraint Propagation Application to numeric CSPs. Ecole des Mines de Nantes - Departement Informatique.

Kincaid R.K., Griffith M., Sykes R., (2002). Bell-Curve Genetic Algorithm for Mixed Continuous and Discrete Optimisation Problems. NASA-AIAA-2002-1675.

Kodiyalam S., (1998). Evaluation of Methods for Multidisciplinary Design Optimisation (MDO). Phase I. NASA-CR-1998-208716.

Kodiyalam S., Sobieszczanski-Sobieski J., (1999). Bi-Level Integrated System Synthesis with Response Surfaces. AIAA 99-1306-wip.

Kroo I., Altus S., Braun R., Gage P., Sobieski I., (1994). Multidisciplinary Optimisation Methods for Aircraft Preliminary Design. AIAA-94-4325.

Laban M. (2004). Multi-Disciplinary Analysis and Optimisation of Supersonic Transport Aircraft Wing Planforms. AIAA-2004-4542 10th AIAA/ISSMO Multidisciplinary Analysis and Optimisation Conference, Albany, New York, Aug. 30-1, 2004.

Laban M., Arendsen P., Rouwhorst W.F.J.A., Vankan W.J., (2002). A Computation Design Engine for Multi-Disciplinary Optimisation With Application to a Blended Wing Body Configuration. MOB Prohct, National Aerospace Laboratory, NLR, The Netherlands.

Michelena N., Parky H.A., (1999). Papalambros P., Kulkarni D., Hierarchical Overlapping Coordination for Large-Scale Optimisation by Decomposition. AIAA-J-99.

Mukhopadhyay V., Sobieszczanski-Sobieski J., (2002). Analysis, Design and Optimisation of Non-Cylindrical Fuselage for Blended-Wing Body (BWB) Vehicle. NASA-AIAA-2002-5664.

Onwubiko, Chinyere, (2000). Introduction to engineering design optimisation. Upper Saddle River, NJ : Prentice Hall, 2000.

Palli N., Azarm S., McCluskey P. (1998). An interactive Multistage ϵ -Inequality Constraint Method For Multiple Objective Decision Making. ASME Vol. 120, December 1998, pp. 678-686.

Papalambros P.Y., Wilde D.J., (2000). Principles of optimal design : modeling and computation. Cambridge University Press, 2000.

Parsons S., Hunter A., (1998). A Review of Uncertainty Handling Formalism. Department of Electronic Engineering, Queen Mary and Westfield College.

Patrick N. Koch, J.P. Evans, and David Powell, (2001). Interdigitation for Effective Design Space Exploration Using iSIGHT. *Journal of Structural and Multidisciplinary Optimisation*, Vol. 23, No. 2, pp. 111-126.

Rao J., Badhrinath K., Pakala R., Mistree F., (1995). A Study of Optimal Design Under Conflict Using Models of Multi-Player Games (Not published).

Röhl, P.J., Mavris, D.N., Schrage, D.P., HSCT Wing Design through Multilevel Decomposition, AIAA-95-3944, Presented at the 1st AIAA Aircraft Engineering, Technology, and Operations Congress, Los Angeles, CA, September 19-21, 1995.

Sobieszczanski-Sobieski J., Emiley M.S., Agte J.S., Sandusky R.R. Jr, (2000). Advancement of Bi-Level Integrated System Synthesis (BLISS). NASA-2000-tm210305.

Sobieszczanski-Sobieski J., Haftka R.T., (1996). Multidisciplinary Aerospace Design Optimisation - Survey of Recent developments. NASA-AIAA-96-0711.

Sobieszczanski-Sobieski J., Kodiyalam S., (1999). Bliss-S a New Method for Two-Level Structural Optimisation. AIAA-99-1345.

Sobieszczanski-Sobieski J., Laba K., Kincaid R., (1998). Bell-Curve Based Evolutionary Optimisation Algorithm. NASA-AIAA-98-4791.

Striz A.G., Sobieszczanski-Sobieski J., (1996). Displacement Based Multilevel Structural Optimisation. NASA-AIAA-96-4098.

Wallace R.J., Freuder E.C., Stable Solutions for Dynamic Constraint Satisfaction Problems. Computer Science Department University of New Hampshire (Not Published).

Walsh J.L., Townsend J.C., Salas A.O., Samareh J.A., Mukhopadhyay V., Barthelemy J.-F., (2000). Multidisciplinary High-Fidelity Analysis and Optimisation of Aerospace Vehicles, Part 1: Formulation. AIAA-2000-0418.

10.4 Uncertainty

Koch P. N., Wujek B., Oleg G., (2000). A Multi-Stage, Parallel Implementation of Probabilistic Design Optimisation in an MDO Framework. AIAA 2000-4805.

Koch P. N., Gu, L., (2001). Addressing Uncertainty using the iSIGHT Probabilistic Design Environment. First Annual Probabilistic Methods Conference, Newport Beach, CA, June 18-19, 2001.

Koch P. N., (2002). Probabilistic Design - Optimizing for Six Sigma Quality. AIAA-2002-1471.

Koch P. N., (2002). Simulation-Based Design in the Face of Uncertainty. 51st Joint Army-Navy-NASA Air Force propulsion meeting.

10.5 Design of Experiments (DOE)

Giunta A., Wojtkiewicz S.F.Jr., Eldred M.S. (2003). Overview of modern design of experiments methods for computational simulations. AIAA 2003-649, 41st Aerospace Sciences Meeting and Exhibit 6-9 January 2003, Reno, Nevada.

Simpson T., Peplinski J., Koch P., Allen J. (1997). On the use of Statistics in Design and the Implications for Deterministic Computer Experiments. DETC97/DTM-3881, Proceedings of DETC'97 1997 ASME Design Engineering Technical Conferences September 14-17, 1997, Sacramento, California.

Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J. and Morris, M.D., (1992). Screening, Predicting, and Computer Experiments. Technometrics, Vol. 34(1), pp. 15-25.

10.6 Other

Kuncicky, David C., (2004). MATLAB programming. Upper Saddle River, NJ : Pearson Prentice Hall, 2004.

Raymer D.P. (1999). Aircraft Design: A Conceptual Approach. AIAA Education Series, New York, NY, 1999.

Torenbeek E. (1976). SYNTHESIS OF SUBSONIC AIRPLANE DESIGN. Kluwer Academic Publishers.

11 Appendix

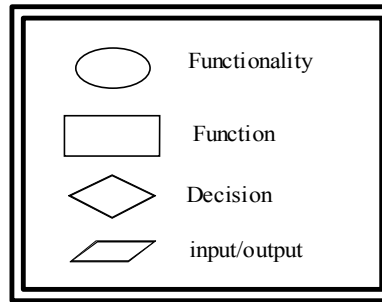
The description of the implementation of the CODE-M algorithm is presented in the following sections. The implementation will be referred to as Calculation Engine (CE). The software here presented allows the setup and solving of any multi-objective optimisation problem with possible presence of strongly connected components.

11.1 Terms and legend

This document contains the Architectural, Interface and Design (Implementation) description of the CE.

Term	Description
CE	Calculation Engine
Model	Computational model of the A/C. Given all the required input variables the model will produce a set of outputs.
Input Variables	Input variables to the model
Output Variables	Output variables to the model
Design Variables	Independent variables for the optimisation process
Constant Variable	Constant variables for the optimisation process
Objective Variables	Variables to be minimised or maximised by the optimisation process
Constraint Variables	Variables set us constraints for the optimisation process
SCC	Strongly Connected Component. When one or more input variables to the model coincide with one or more variables of the output a SCC is formed.
Subfunction	A subfunction is a function contained in the body of a function. A subfunction, is visible only to the other functions or subfunctions in the same file. Subfunctions are not visible outside the file where they are defined.

Legend:



11.2 Architecture Description

The CE can be seen as having three main functionalities as shown in Figure 75. A more comprehensive description of the individual functionalities is shown in Figure 76.

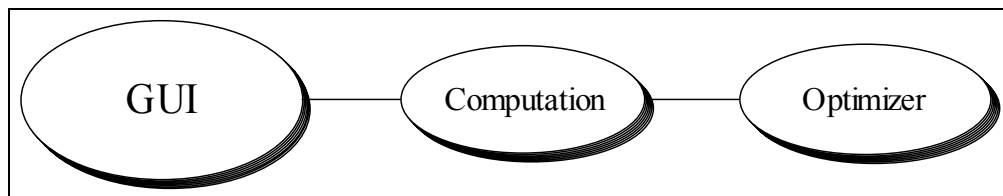


Figure 75: Top- level Functional System

Through the GUI all the data necessary for performing the multi-objective optimisation is collected. All such information is then passed to the Computation component which sets up a number of optimisations. The solutions of the optimisations will be used to determine the single objective local minima at first and finally the Pareto points. Each of the optimisations set up by the Computation component is performed by the Optimiser component. The information relative to the variables and the Model function are stored in objects.

The CE Prototype was developed in Matlab and thus can be executed on any platform supported by Matlab. Currently the compiled standalone version is only available for windows. The model on which the CE can perform the optimisation procedure can be a windows .exe executable or a Matlab .m.

Figure 76 shows the three main components of the CE (GUI, COMPUTATION and OPTIMISER). The figure also shows the functionalities relative to each component.

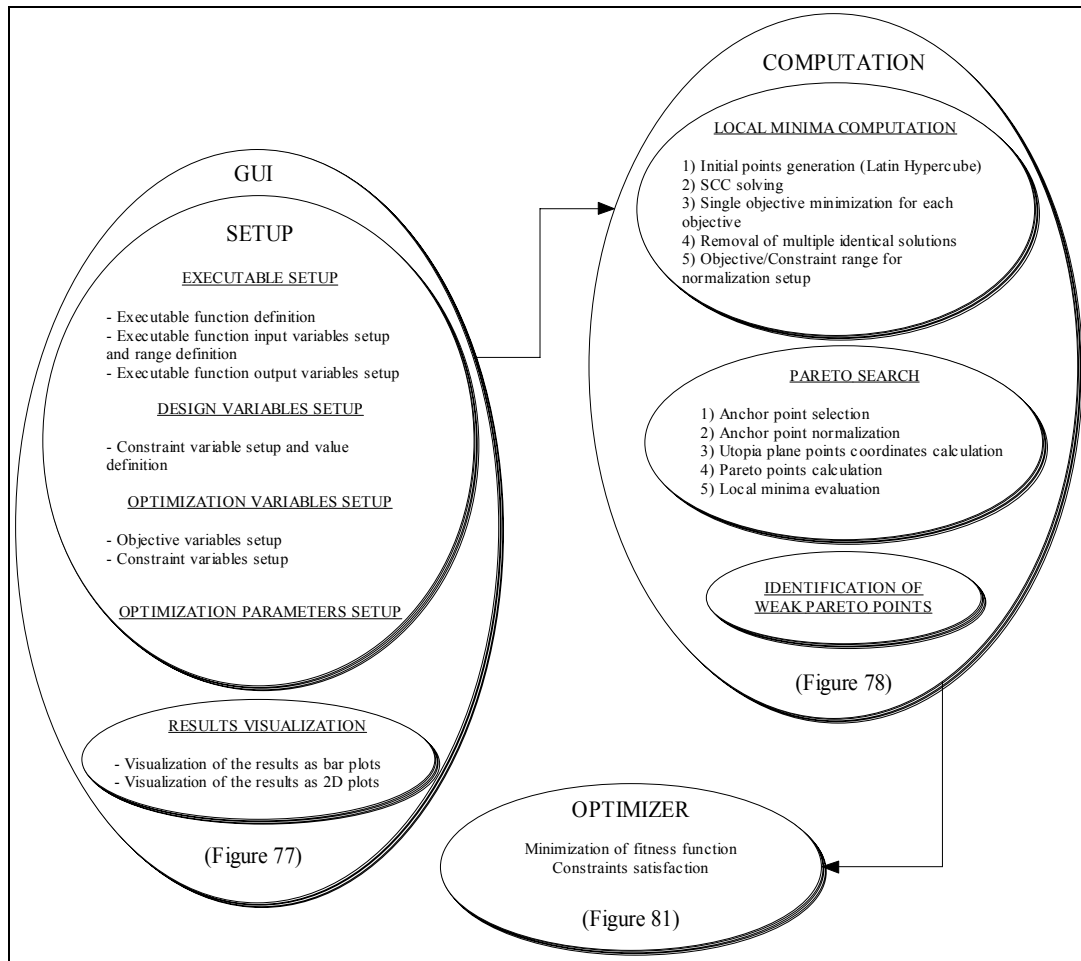


Figure 76: Detailed Functional System

Figure 77 to Figure 81 show the breakdown structure of each of the main three components of Figure 76. Non-fundamental functions have been omitted. It is to be noted that the information relative to the variables and the Model function are stored in objects.

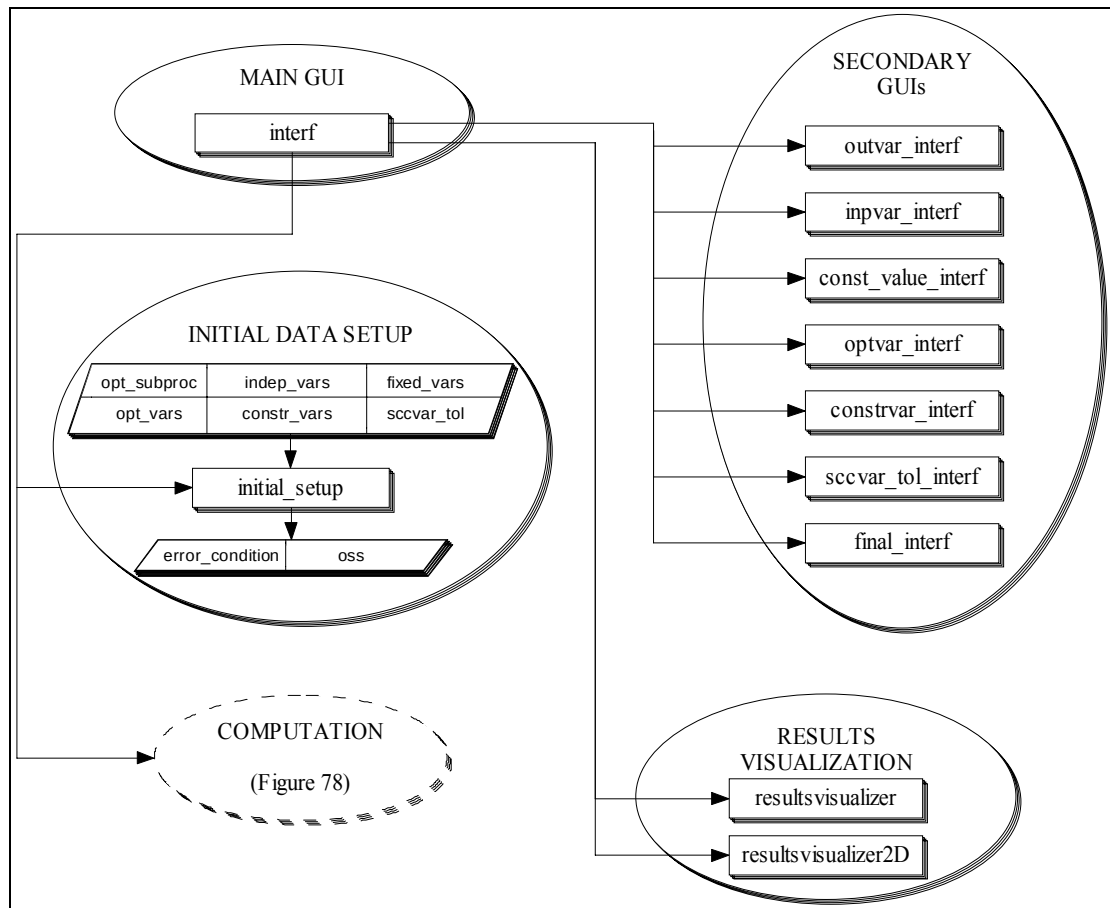


Figure 77: GUI breakdown

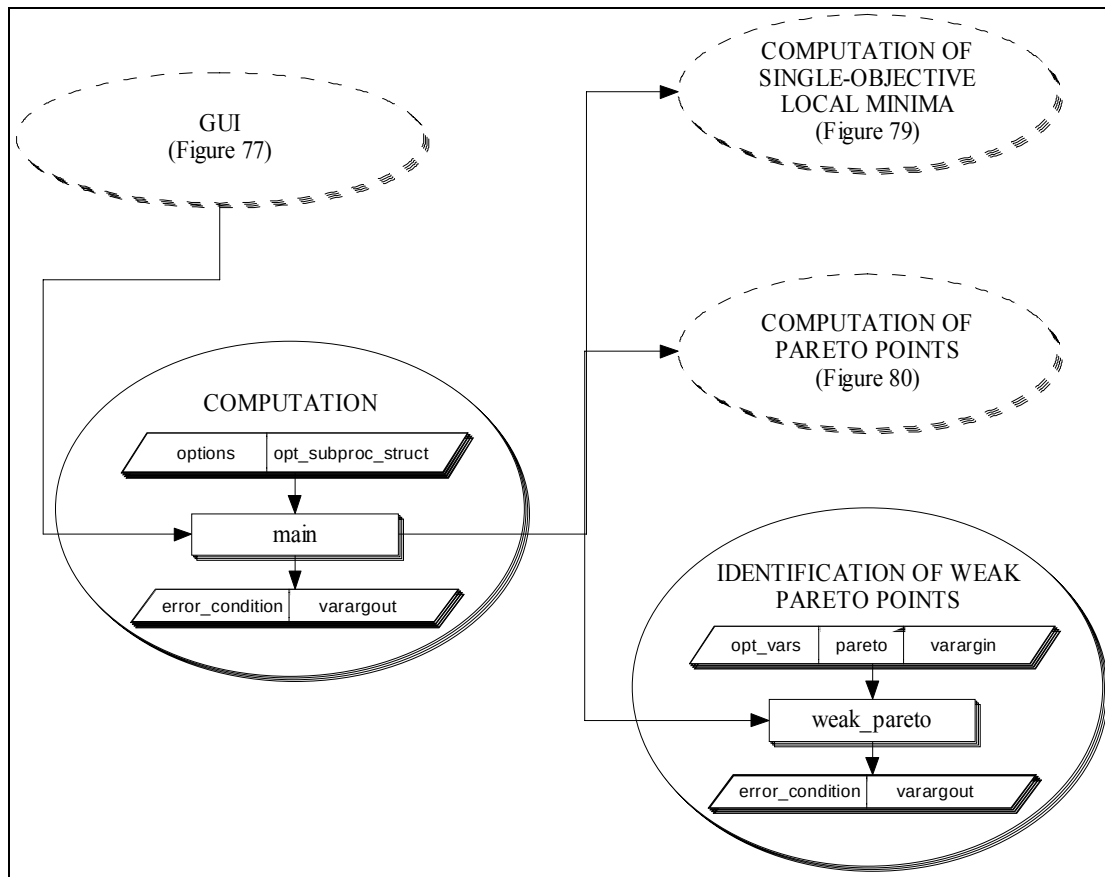


Figure 78: Computation Module breakdown

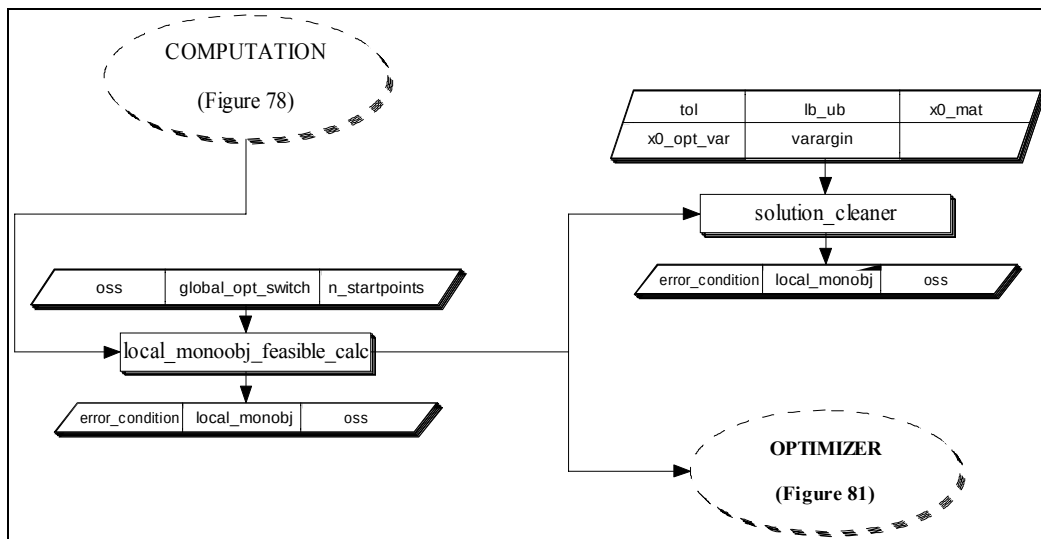


Figure 79: Computation of single-objective local minima Module breakdown

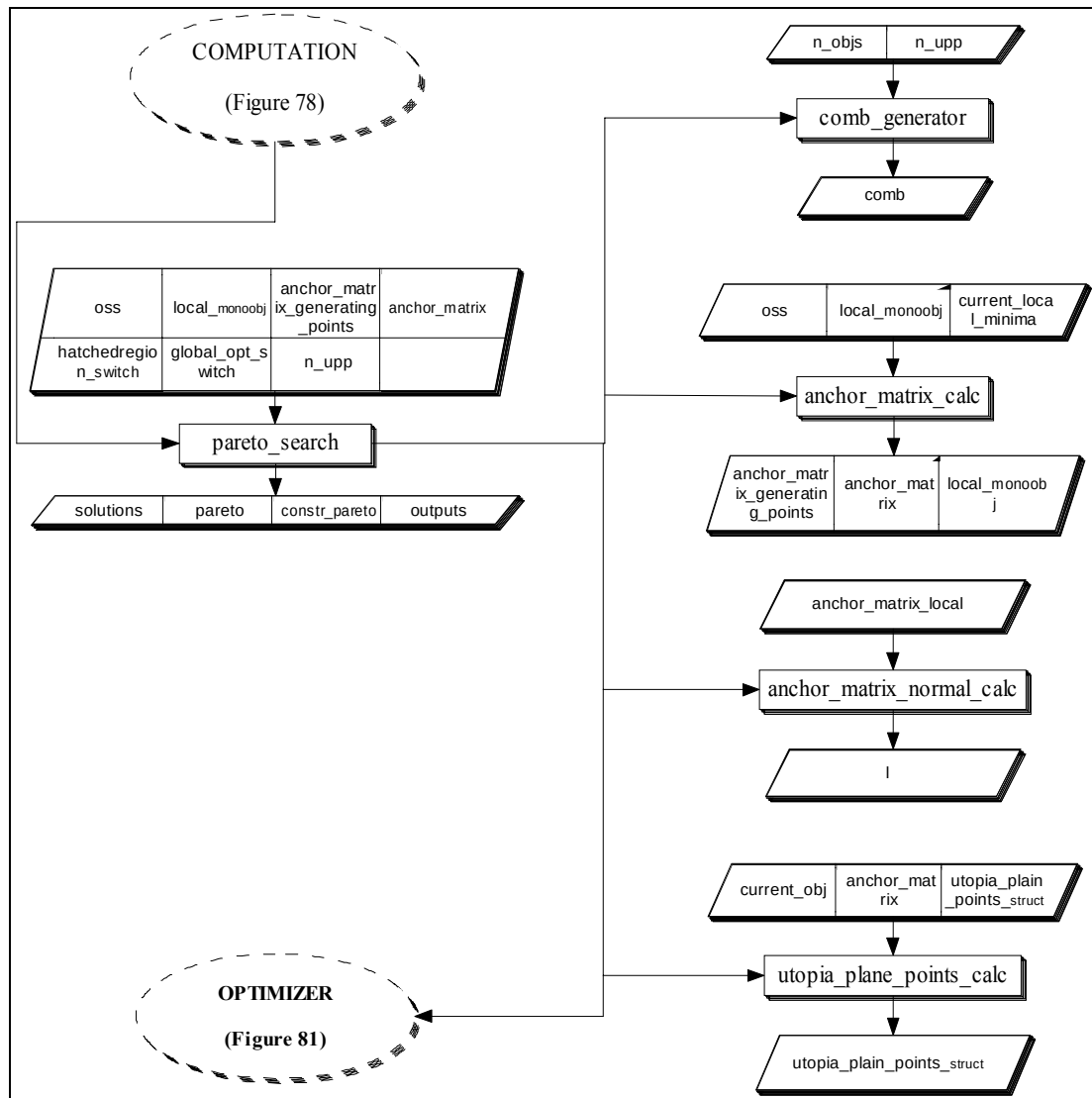


Figure 80: Computation of Pareto points Module breakdown

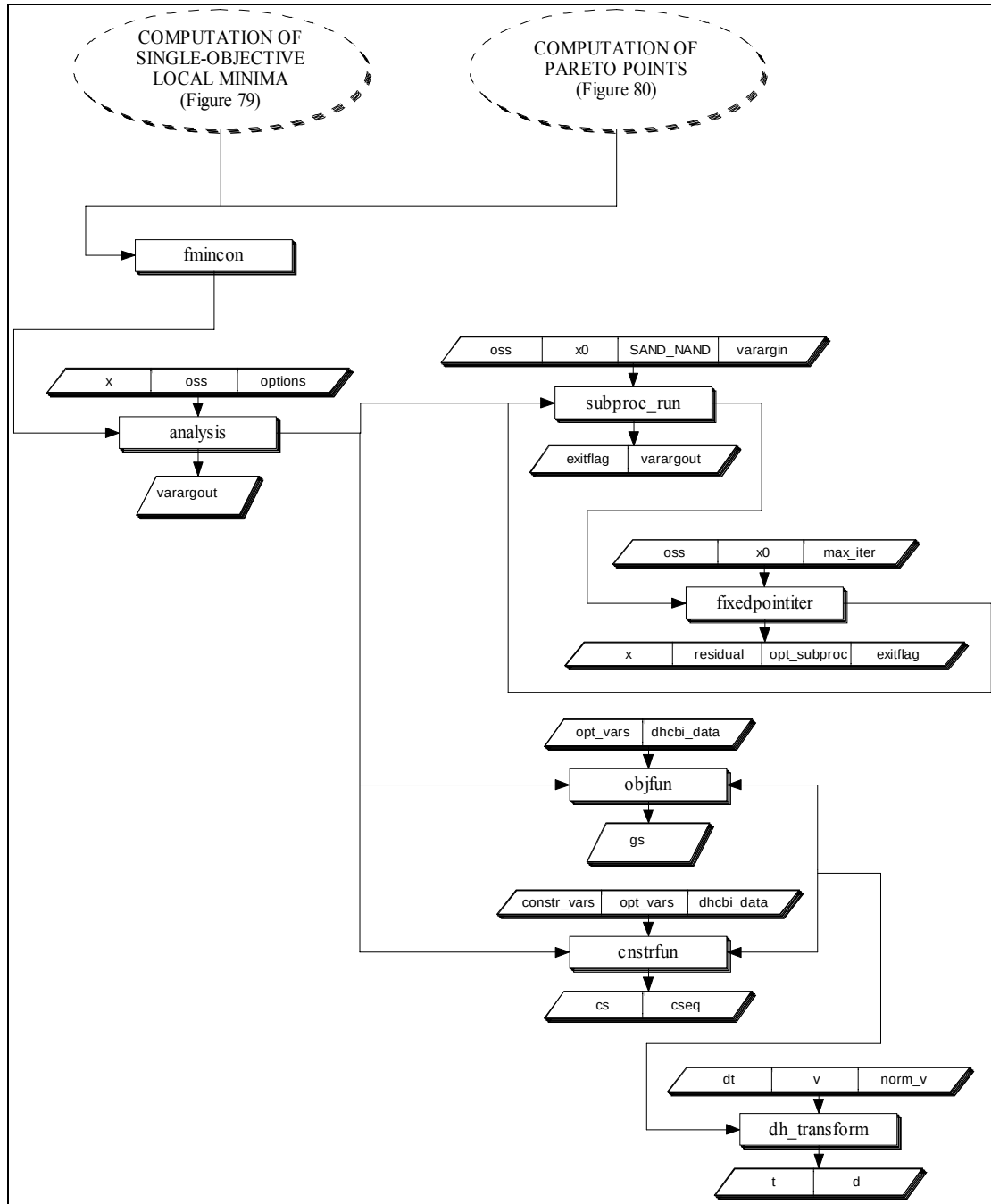


Figure 81: Optimisation Module breakdown

11.3 Components Details

As for the description of the architecture, three main components are presented: the GUI (paragraph 11.3.1), the Computation component (paragraph 11.3.2) and the Optimiser Component (paragraph 11.3.3).

In the following sections, for each component, a brief description of the functions is given.

11.3.1 Component GUI

ID	Short Title	Description	Function
1	Main Interface	<p>Main interface for the setup of the optimisation problem</p> <p><u>called functions:</u> outvar_interf inpvar_interf const_value_interf optvar_interf constrvar_interf sccvar_tol_interf resultsvisualizer resultsvisualizer2D</p>	interf
ID	Short Title	Description	Function
1.1	Output variables setup interface	<p>Interface for setting up name, unit and range of variation of each input variable of the model</p> <p><u>called functions:</u> NILL</p>	inpvar_interf
ID	Short Title	Description	Function
1.2	Input variables setup interface	<p>Interface for setting up name and unit of each output variable of the model</p> <p><u>called functions:</u> NILL</p>	outvar_interf
ID	Short Title	Description	Function
1.3	Constant variables setup interface	<p>Interface for setting up the value of each constant variable</p> <p><u>called functions:</u> NILL</p>	const_value_interf
ID	Short Title	Description	Function
1.4	Optimisation variables setup interface	<p>Interface for setting up the value of each optimisation variable</p> <p><u>called functions:</u> NILL</p>	optvar_interf
ID	Short Title	Description	Function
1.5	Constraint variables setup interface	<p>Interface for setting up the value of each constraint variable</p> <p><u>called functions:</u> NILL</p>	constrvar_interf
ID	Short Title	Description	Function
1.6	SCC variables tolerance setup interface	<p>Interface for setting up the tolerance value of each SCC variable</p> <p><u>called functions:</u> NILL</p>	sccvar_tol_interf
ID	Short Title	Description	Function
1.7	Results visualization as bar plot	<p>Interface for setting up all the parameters for solving required by the Computation component</p> <p><u>called functions:</u> NILL</p>	final_interf

ID	Short Title	Description	Function
1.8	Initial Setup Module	Analyses the optimisation problem data gathered through the interface and generates structure oss containing all the information needed by the computation components <u>called functions:</u> NILL	initial_setup

ID	Short Title	Description	Function
1.9	Results visualization as bar plot	Interface for setting up and displaying the visualization of the results as bar plots <u>called functions:</u> NILL	resultsvisualizer

ID	Short Title	Description	Function
1.10	Results visualization as 2D plots	Interface for setting up and displaying the visualization of the results as 2D plots <u>called functions:</u> NILL	Resultsvisualizer 2D

11.3.2 Component Computation

ID	Short Title	Description	Function
2	Main Computation Function	Backbone of the computation module which links all the other components <u>called functions:</u> initial_setup local_monobj_feasible_calc pareto_search weak_pareto	main

ID	Short Title	Description	Function
2.1	Computation of single-objective local minima	Determines the local minima for each of the objectives defined <u>called functions:</u> fmincon solution_check (subfunction)	local_monobj_feasible_calc

ID	Short Title	Description	Function
2.1.1	Removal of multiple points	Determines for each objective if a point is present multiple times, if any such point is found it is removed. In case the objective indices (empty x0_opt_var) are not given the points are compared independently of the objectives. <u>called functions:</u> NILL	solution_cleaner

ID	Short Title	Description	Function
2.2	Compute Pareto points	Main function for the computation of the Pareto points <u>called functions:</u> comb_generator anchor_matrix_calc anchor_matrix_normal_calc utopia_plane_points_calc dominated_anchorpoints_ceck (subfunction) solve (subfunction) fmincon	pareto_search

ID	Short Title	Description	Function
2.2.1	Combination generator	Computation of the trade-off matrix for the computation of the utopia plane points (comb_generator is a recursive function) <u>called functions:</u> comb_generator	comb_generator

ID	Short Title	Description	Function
2.2.2	Anchor points selection	Determination of the anchor points <u>called functions:</u> NILL	anchor_matrix_calc

ID	Short Title	Description	Function
2.2.3	Compute normal vector	Determination of the vector normal to the utopia plane <u>called functions:</u> NILL	anchor_matrix_normal_calc

ID	Short Title	Description	Function
2.2.4	Compute utopia plane points	Determines the utopia plane points <u>called functions:</u> NILL	utopia_plane_points_calc

ID	Short Title	Description	Function
2.3	Weak Pareto points	Determines weak Pareto points <u>called functions:</u> NILL	weak_pareto

11.3.3 Component Optimiser

ID	Short Title	Description	Function
3	Analyse and Compute Fitness / Constraints	Main function for the determination of the fitness or the constraints according to request <u>called functions:</u> subproc_run objfun cnstrfun	analysis

ID	Short Title	Description	Function
3.1	Model analysis	Execution of the model for a given design point <u>called functions:</u> fixedpointiter	subproc_run

ID	Short Title	Description	Function
3.1.1	SCC solver	Solution of the strongly connected component if present <u>called functions:</u> subproc_run	fixedpointiter

ID	Short Title	Description	Function
3.2	Compute objective function	Computation of the value of the objective function <u>called functions:</u> dh_transform	objfun

ID	Short Title	Description	Function
3.3	Compute constraints values	Computation of the values of the constraints <u>called functions:</u> dh_transform	cnstrfun

ID	Short Title	Description	Function
3.2.1	Compute vector decomposition	Determines the Pareto points <u>called functions:</u> NIL	dh_transform

Although the software is being extensively tested, it is still possible that no solution might be found. The cases in which this might happen are the following:

- User supplied constraints are such that no feasible solution exists.
- The number of starting points and the maximum number of iterations defined by the user are not adequate to the complexity of the optimisation problem.

An additional possible problem is that the optimiser might not be capable of obtaining all the local Pareto solutions. This, again, might happen if the number of starting points supplied by the user is not adequate to the complexity of the optimisation problem. It must be noted that these limitations are common to any optimisation tool.

Although the software is ideally suited for application to smooth objective functions, it has been successfully applied to problems with discontinuities.