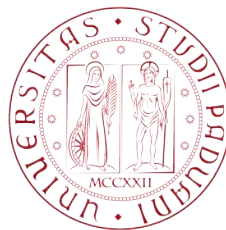


UNIVERSITÀ DI PADOVA

*Corso di Laurea in Ingegneria Aerospaziale
Dipartimento di Ingegneria Industriale DII*

Tesi di Laurea Magistrale

Comparison of genetic and tabu search algorithms in aerodynamic design of S-ducts



Supervisor

Prof. Ernesto Benini

Riccardo Tridello

Correlatori:

Prof. M. Savill

Dr. T. Kipouros

14 July 2017

Acknowledgments

I would like to express my gratitude to my supervisors Prof. Ernesto Benini and Prof. Mark Savill, for giving me the opportunity to work on this interesting project.

I am grateful to Dr. Nicola Chiereghin and to Dr. Kipouros, for their great help and their support throughout this thesis. Their experience and advice have been fundamental for this work.

I would like to thank my father and my mother, Maurizio and Giuliana, for their constant presence and support given during my university career.

At least but not the least I would like to thank my girlfriend Francesca for her love and support during these years.

Contents

Cover	I
List of Tables	VIII
List of Figures	IX
1 Introduction	1
1.1 Context and Background	1
1.2 Distributed Propulsion (DP) and Boundary Layer Ingesting (BLI)	3
1.3 Aims of the Projects	4
1.4 Tools Used	5
2 Literature Review	7
2.1 S-Duct Flow Physics	7
2.1.1 Swirl	7
2.1.2 Total Pressure Losses	11
2.1.3 Flow Separation	12
2.1.4 Flow Distortion Consequences	13
2.2 Prior Studies on S-Ducts	15
2.2.1 Experimental Researches	15
2.2.2 Computational Research	18
3 Multi-Objective Optimization	21
3.0.1 Concept of Dominance	23
3.0.2 Resolution methods	24
4 Optimization Algorithms	25
4.1 Tabu Search	26
4.1.1 Multi-Objective Tabu Search (MOTS)	26

4.1.2	Algorithm Overview	27
4.1.3	Hooke and Jeeves Move	29
4.1.4	Intensification Memory	30
4.2	GDEA - II	31
4.2.1	Multi-Objective Evolutionary Algorithms	31
4.2.2	The Genetic Diversity Evolutionary Method (GeDEM)	32
4.2.3	The Genetic Diversity Evolutionary Algorithm (GDEA)	34
4.2.4	Differences from GDEA to GDEA - II	35
5	Methodology	39
5.1	Geometry	40
5.2	Parametrization	42
5.2.1	Constraints	43
5.3	Mesh Description	44
5.3.1	Mesh Generation	45
5.4	Ansys Fluent Settings	47
5.5	Objective Functions	48
5.5.1	Swirl	49
5.5.2	Pressure Losses	50
5.6	Optimizations loop	50
5.6.1	Nimrod	51
6	Analysis and Discussion of the Results	53
6.1	Baseline Analysis	54
6.2	Optimization Results	57
6.2.1	GDEA-II Optimization Results	57
6.2.2	Tabu Search Optimization Results	62
7	Algorithms Comparison	65
7.1	Pareto Front Comparison	65
7.2	Optimum Geometries Comparison	68
7.2.1	Lowest Total Pressure Leakages Solutions	69
7.2.2	Lowest Swirl Angle Solutions	71
7.2.3	Best Trade Off Solutions	72
8	Conclusions	75
8.1	Recommendations for Future Researches	76

Bibliography	79
A Nimrod Schedule File	83

List of Tables

- 5.1 *S-Duct baseline geometry parameters* 41
- 5.2 *S-Duct baseline geometry non-dimensional parameters* 42
- 5.3 *Boundary conditions parameters for the simulations* 48
- 5.4 *Flow parameters* 48
- 6.1 *Baseline geometry objective functions* 54
- 6.2 *Baseline compared to Generation 1 Pareto front* 59
- 6.3 *Baseline compared to the final Pareto front of GDEA-II* 61
- 6.4 *Baseline compared to MOTS Pareto front solutions* 63
- 7.1 *Improvements of MOTS from its starting point, and improvements of GDEA-II from the baseline, withing the first 53 solutions for both of the algorithms* 67
- 7.2 *trade-off solutions of GDEA-II* 73
- 8.1 *Improvements of MOTS and GDEA-II from the baseline* 76

List of Figures

1.1	<i>Falcon 7X, still in production</i>	2
1.2	<i>Boeing 727, not in production</i>	2
1.3	<i>General Dynamics F-16, example of a military usage of S-Ducts</i>	2
1.4	<i>Concept of distributed propulsion</i>	3
1.5	<i>Benefits of BLI-podded case and 100% BLI. The momentum excess created by the podded engine is equal to the momentum deficit of the airframe [2]</i>	4
2.1	<i>Velocity components and swirl angle [7]</i>	8
2.2	<i>Schematization of the bulk swirl [7]</i>	8
2.3	<i>Paired swirl description based on pressure gradient and momentum theory [7]</i>	9
2.4	<i>Tightly-wound vortex attached to the ground [7]</i>	10
2.5	<i>Tightly-wound vortex attached to airplain surface [7]</i>	10
2.6	<i>Representation of total, static and dynamic pressure</i>	11
2.7	<i>3D representation of flow separation in an S-Duct [10]</i>	12
2.8	<i>Surface oil flow patterns on one symmetric half of the duct surface [10]</i>	13
2.9	<i>Close up of the surface oil flow patterns developed in the separation region [10]</i>	13
2.10	<i>Diverted flow during compressor stall [11]</i>	14
2.11	<i>Facility scheme of Wellborn's experiment [10]</i>	15
2.12	<i>Half shell of the circular diffusing S-Duct [10]</i>	16
2.13	<i>Axial distribution of surface static pressures for three circumferential positions [10]</i>	17
2.14	<i>Total pressures in the five measurement planes [10]</i>	18
2.15	<i>Delot's S-Duct model</i>	19
2.16	<i>PR distributions (on the left) and Swirl distributions (on the right) comparison between the baseline and an optimized design. [18]</i>	20
3.1	<i>Design space of a MOOP divided in feasible and infeasible space</i>	22
3.2	<i>Visual representation of the Pareto Front</i>	23
4.1	<i>Simplification scheme of the optimizations carried out in this thesis</i>	25

4.2	<i>Flow chart for the MOTS_{ii} algorithm.</i>	27
4.3	<i>Point selection for the Hooke and Jeeves move and Tabu search memories [22]</i>	29
4.4	<i>Single point crossover</i>	32
4.5	<i>GeDEM fitness assignment [26]</i>	33
4.6	<i>Modification of the Pareto optimality definition in the application of GeDEM [26]</i>	34
4.7	<i>The reflect step of the simplex algorithm applied to a problem in \mathbb{R}^2 [27]</i>	36
5.1	<i>S-Duct geometry [10]</i>	40
5.2	<i>S-Duct baseline scheme</i>	41
5.3	<i>Control Points.</i>	43
5.4	<i>Difference between structured (on the left) and unstructured meshes (on the right)</i>	45
5.5	<i>Half geometry mesh</i>	46
5.6	<i>H-O grid at AIP, the outlet of the intake</i>	46
5.7	<i>Area-Averaged swirl parameter</i>	49
6.1	<i>Baseline PR Comparison: Left: Delot Experiment, Center: Delot Simulation, Right: Our simulation</i>	54
6.2	<i>Swirl representation at baseline AIP</i>	55
6.3	<i>Velocity vectors at baseline AIP</i>	55
6.4	<i>z-velocity in baseline symmetry plane</i>	56
6.5	<i>Pressure fluctuations in baseline symmetry plane</i>	56
6.6	<i>Objective functions values of the Generation 0, the orange square is the baseline evaluation</i>	58
6.7	<i>Generation 1 Pareto front</i>	59
6.8	<i>Generation 1 Pareto front</i>	60
6.9	<i>Generation 1 Pareto front</i>	61
6.10	<i>Final Pareto front after Generation 12</i>	62
6.11	<i>All the individuals evaluated with MOTS: "first run" are the individuals from 0 to 15, "second run" are the individuals from 16 to 33, and "third run" are the last 17 individuals, from 16 to 51.</i>	63
6.12	<i>All the individuals evaluated with MOTS compared to the baseline geometry</i>	64
6.13	<i>Close-up on the Pareto front of found from Tabu Search</i>	64
7.1	<i>Pareto fronts comparison between GDEA-II with 13 generations, and MOTS after 52 individuals</i>	66
7.2	<i>Comparison between the starting points of GDEA and MOTS</i>	66

7.3	<i>Highlighting the fact that the individual 0_51 from the starting population is a non-dominated solution even after 12 more generations</i>	67
7.4	<i>Comparison between the first 53 solution of both of the algorithms</i>	68
7.5	<i>Left: Solution 12_0 from GDEA-II, Right: 7th solution from MOTS; both compared against the baseline.</i>	69
7.6	Separate region comparison: Top: 12_0, Bottom: 7	70
7.7	Swirl angle at AIP	70
7.8	<i>Left: Solution 0_51 from GDEA-II, Right: 11th solution from MOTS; both compared against the baseline.</i>	71
7.9	Separate region comparison: Top: 0_51, Bottom: 11	72
7.10	Swirl angle at AIP, solution 0_51	72
7.11	<i>Left: Solution 8_6 from GDEA-II, Right: 6th solution from MOTS; both compared against the baseline.</i>	73
7.12	Separate region comparison: Top: 8_6, Bottom: 4 and 6	74
7.13	Swirl angle at AIP, solution 8_6	74

Chapter 1

Introduction

The topic of this thesis is to compare the results of the multi-objective optimization of the shape of S-Ducts intakes using two different algorithms, a genetic and a tabu search one. An S-duct (or serpentine inlet) is a type of jet engine intake used in several types of aircraft. It was invented as a solution for positioning the engine in a central position, and consist in a channel with an offset and a growth of the cross section, its function is to bring the air from the inlet to the outlet increasing the static pressure in order to help the compressor. The design of airbreathing propulsion systems almost always involves having to deal with the mismatch between the flow conditions in the freestream and those required by the engine at the entrance of the fan or compressor. Achieving the required transformation is the central role of the inlet and the efficiency of this process is the main objective that drives its design. A good inlet design will maximize the increasing of static pressure, while at the same time minimizing the total pressure losses and the fluid distortion. These two objectives are in conflict most of the time and thus the solution is not unique, but it has to be a tradeoff.

1.1 Contex and Background

S-Ducts intakes can be found in a wide range of aircraft propulsion systems, both for military and civil applications. Even if nowadays in the civil aviation S-Ducts are not widely used, there are many researches trying to integrate this type of intakes more often. The main purpose is the reduction of costs thus the research scope for this kind of planes is focused on the improving of efficiency and this can be achieved with distributed propulsion (DP), and boundary layer ingestion (BLI), that will be seen more in detail in the next section. Currently there are many commercial aircraft that are using S-Ducts intakes, both still in production and not. Two examples could be the Falcon 7X for the

former case (Figure 1.1) or Boing 727 for the latter (Figure 1.2). Contrariwise for military aircrafts, the main skills are performances like lightness, reactivity and concealment from radars. S-Ducts suits perfectly these requirements, enabling a reduced mass achieved with the engine integrated within the airframe, and keeping the compressor stage hidden from enemies radars.



Figure 1.1: *Falcon 7X, still in production*



Figure 1.2: *Boeing 727, not in production*



Figure 1.3: *General Dynamics F-16, example of a military usage of S-Ducts*

1.2 Distributed Propulsion (DP) and Boundary Layer Ingesting (BLI)

A conventional subsonic transport aircraft is equipped with two to four turbofan engines. The trend from the fifties until now has favored the twin underwing engine configuration. An alternative is to use a large number of engines. This concept can broadly be described as distributed propulsion.



Figure 1.4: *Concept of distributed propulsion*

Some advantages related with this design could be lowered structural load that lead to a lower aircraft and engine structure weight. Distributed propulsion may also include lower engine installation drag, the potential to increase propulsive efficiency, decreased noise, better material properties of small components, increased aircraft configurational freedom and mass production cost advantages. A smaller engine may also find a wider application for transport aircraft of various sizes as well as business jets and UAVs, which would increase production efficiency and spread development cost. This design has also some disadvantages like increase of pressure and heat losses because of the decreased Reynold number, increase of leakage, increase of maintenance cost and increase scale effects if the engine are too small. Since there are more advantages then disadvantages, distributed propulsion is going to be the future of aeronautical propulsion.

Furthermore recent researches conducted by Boeing [1] are focusing on the achievable advantages of Boundary Layer Ingestion (BLI) in planes that combine Blended Winged Body (BWB) and DP. A key goal of next generation propulsion systems is to provide continued reductions in fuel burn relative to the best gas turbine engines in development today. One path to achieve this is to enable boundary layer ingesting (BLI) propulsion

systems, which can provide significant improvements in propulsive efficiency by producing thrust from the reduced velocity boundary layer air. The next generation of ultra high bypass (UHB) turbofan propulsion systems will feature bypass ratios between 15 and 18, and fan pressure ratios in the 1.25 - 1.35 range. Such engine cycles will result in significantly reduced thrust specific fuel consumption (TSFC). The implementation of this class of propulsion system in a boundary layer ingesting environment would provide substantial vehicle-level fuel burn benefits. A key challenge associated with boundary layer ingesting propulsion systems is the ability of the turbomachinery to operate efficiently in highly distorted flow. In particular, a high-performance, distortion-tolerant fan will be required. The benefit of boundary layer ingestion comes from re-energizing the aircraft wake. This enables less kinetic energy to be wasted (in the sense that the kinetic energy produced is over and above the amount needed for propulsion) [2]. The physical concept can be illustrated simply using the two idealized configurations shown in Figure 1.5

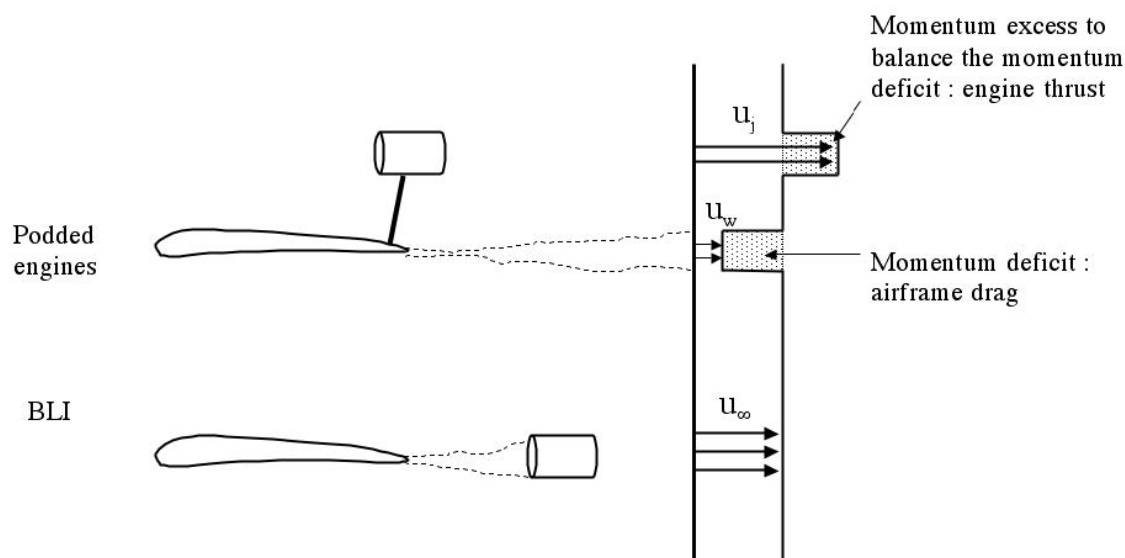


Figure 1.5: *Benefits of BLI-podded case and 100% BLI. The momentum excess created by the podded engine is equal to the momentum deficit of the airframe [2]*

1.3 Aims of the Projects

This project has been carried out at Cranfield University in collaboration with the University of Padova, thanks to Prof. Savill and Prof. Benini. Several other works on S-Ducts have preceded this one, both from PhD and MSc students, and they have been very useful because this work starts from their results. The most important ones come

from Nicola Chiereghin who has previously worked on this field, and provided the main python code used, and Aurora Rigobello that studied S-Ducts with the genetic algorithm NSGA-II and determined the parametrization used in this thesis [3]. As already told, distributed propulsion needs S-Ducts intakes, but their shapes cause pressure and velocity distortions distributions, negatively affecting compressor operations. Thus, in the last years experimental and computational projects have been carried out to predict the flow throw S-Ducts to make the flow as uniform as possible. The goal of this work is the comparison of genetic and tabu search algorithms for the optimization of S-Ducts to obtain the best design, able to minimize total pressure losses and flow deformations. During this thesis we are going to call this two physic phenomenons as objective functions, because they are the ultimate goal of both of the optimizations. In the next section we will study in depth what are these phenomenons and why they occur.

Other works have been done about this topic, but these two optimization algorithms have never been compared before, and this has been possible thanks to sharing of information between the two Universities. This project, with all the other researches, is intended to be a contribution for future studies, giving information about the physics of the problem and about which optimization is better suited for this research.

1.4 Tools Used

Empirical experimentation are expensive and slow, so all of the tests for this research have been done with the help of computer simulations. A lot of different software were used, and all of them were connected to each other using python codes. The briefly description of the optimization loop consist on a geometry creation, entirely done with python, which is the base for a mesh that is created with ANSYS ICEM. Then is the turn of ANSYS FLUENT who simulate the flow throw the duct, and gives back the files that has to be investigated with python to evaluate the objective functions we are looking for at the outlet. ANSYS FLUENT is a computation fluid dynamics (CFD) software tool, that is used by engineers and researchers to simulate and predict flow behavior. This software includes well-validated physical modeling capabilities to deliver analysis of three-dimensional stream, in our case, within the S-Ducts. Because of the huge amount of computation power required, to produce results in a reasonable short period of time, all the software were used on ASTRAL and GRID, two clusters that allows parallel processing and high computing power [4] [5].

The most interesting part of this research is the multi objective optimization (MOOP) based on genetic or tabu search algorithms. In the former case an evolutionary algorithm is

used, the GEDeA-II, created by the Padova University. In the latter case a meta-heuristic algorithm called Tabu Search created by Cranfield University. The aim of this MOOP is to minimize or maximize two or more objective functions, changing the design parameter to find the best trade-off solutions.

Chapter 2

Literature Review

The main goal of this thesis, as already said, is to minimize the objective functions, but to understand what they are, and why they occur, the physics of the problem has to be understood. Thus this chapter will explain the physics of a subsonic flow in a S-Duct intake, and then there will be an overview of the experimentation researches that have been done about this topic.

2.1 S-Duct Flow Physics

Diffusing S-Ducts have centerline curvature and cross-sectional area increase. Curvature of the centerline or changes in the duct cross-sectional shape lead to streamline curvature that give rise to cross-stream pressure gradients. These gradients can produce secondary flows, and in addition to adverse streamwise pressure gradient, due to the increased cross-sectional area, can lead to flow separation. These secondary flows distort the flow stream with three type of distortion: swirl, total pressure distortion and total temperature distortion. In the next sections the first two phenomenons will be described, the third one instead is not interesting for this research because it is important only for military application.

2.1.1 Swirl

S-Ducts intake are always accompanied by the swirl flow and it may cause serious engine/intake compatibility problems, such as engine surge and fan vibration [6]. The swirl is defined to describe the rotation of the flow inside the duct. In this thesis it is measured only at the outlet section, and it is calculated as the deviation of the velocity vector from the axial direction. The velocity vector is divided into the axial U_z and the

tangential U_θ component, which is also known as swirl, and the swirl angle is defined as:

$$\alpha = \arctan \left(\frac{U_\theta}{U_z} \right)$$

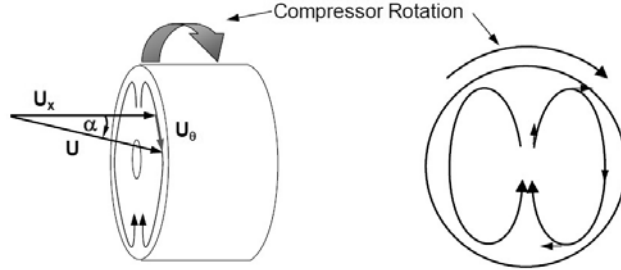


Figure 2.1: *Velocity components and swirl angle* [7]

In Figure 2.1 are shown the components of the velocity vector, and also it can be noticed that the swirl angle is positive when it has the same direction of the compressor rotation. There are four different type of swirl: *bulk*, *paired*, *cross-flow* and *tightly-wound swirl*.

- **Bulk Swirl:**

The bulk swirl distortion occur when the whole flowfield rotate in a single direction about the compressor axis. When it spin with the same rotation of the compressor it is named *co-rotating bulk swirl*, otherwise is called *counter-rotating bulk swirl*. This type of swirl happen when the flow present a non-axis symmetrical total pressure gradient of the S-bend flow [7]. A visual representation of this distortion is shown in Figure 2.2. This kind of swirl can be generated both externally and internally. For

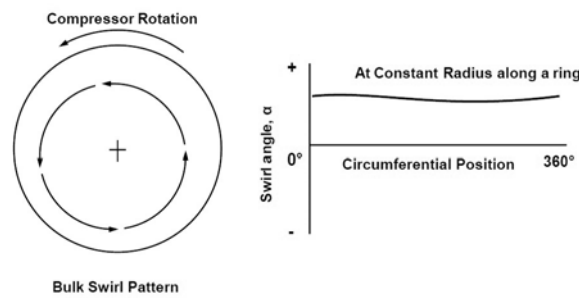


Figure 2.2: *Schematization of the bulk swirl* [7]

example, the former case happens when a large vortex produced before the inlet is ingested by the duct. The latter case instead develop because of the curvature of the S-shaped diffusers ducts.

- Paired Swirl:** This is the most common and important type of swirl and it generate two vortex inside the duct. If they have the same magnitude the swirl is called *twin swirl*, otherwise *offset paired swirl*. This phenomenon can be explained in two different ways, one based on vorticity and the other based on pressure gradient and momentum. The former one can be described in few word, because the whirling of the flow is due by the fact that the flow vorticity vector is deflected by the bent of the duct shape.

With the help of the Figure 2.3 paired swirl can be described more clearly. At the beginning ideal conditions are good enough to understand the behavior of the pressure moving from the inlet to the outlet of the intake. Because of the bend, as the curvature radius increase, the static pressure rises in order to balance centrifugal forces in the top section proximity of the bend. This means also a reduction of the flow velocity. Now, if the ideal condition is taken away, the reader has to remember that boundary layer has to taken in account. Thus the velocity of the stream is not uniform, but it vary from zero at the walls to its maximum at the center of the duct. This means that the flowfield is subjected to a non-uniform momentum distribution. The high velocity core going throw the bend, due to his higher momentum, try to keep his position and it is forced against the upper wall, while the low-momentum fluid has to slips along the walls to the lower part of the bend. Therefore the combination of this events lead to the birth of two paired of counter-rotating vortexes.

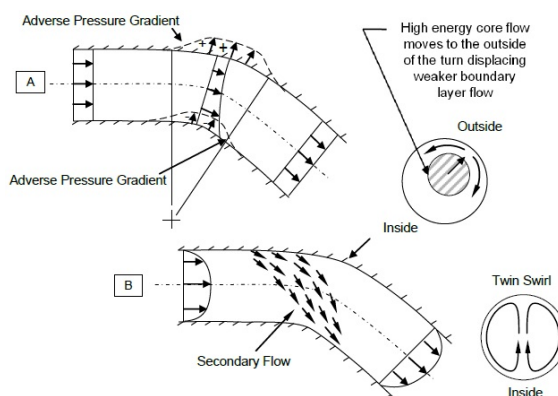


Figure 2.3: *Paired swirl description based on pressure gradient and momentum theory* [7]

- Cross Flow:**

This kind of swirl is very similar to te paired one, except from the fact that the velocity is uniform in the cross-flow direction. Usualy it is related to lift fans, however

it might occur in turboshaft or turboprop with bifurcated intake ducts.

- **Tightly-Wound Vortex:**

Tightly-wound vortices are generated by a lot of different mechanisms such as leading edge extensions, tip vortices and ground operations. In Figure 2.4 can be seen these vortices during static or near-static engine operations at the ground. For this reason they are also known as inlet-ground vortices, even if they can be attached to other surfaces like the ones of the airplane as shown in Figure 2.5



Figure 2.4: *Tightly-wound vortex attached to the ground* [7]



Figure 2.5: *Tightly-wound vortex attached to airplane surface* [7]

2.1.2 Total Pressure Losses

The total pressure is the force per unit area that is felt when a flowing fluid is brought to rest isentropically and is usually measured with a pitot tube type instrument, shown in Figure 2.6. Total pressure is the sum of static and dynamic pressure:

$$P_{tot} = P_{static} + \frac{1}{2}\rho v^2 \quad (2.1)$$

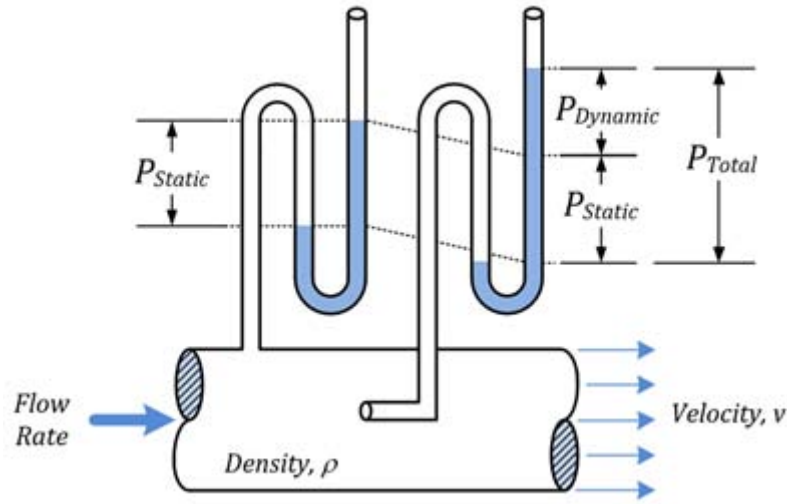


Figure 2.6: Representation of total, static and dynamic pressure

Total pressure losses come from the result of flow separation inside the intake, and S-shaped diffusers always stumble across this problem. The flow separation consist in a vortex that can be seen in the symmetry plane of the duct, and blocks a large part of the cross-section causing an increase of flow velocity. As a matter of fact the total pressure losses are linked to the velocity as shown in (2.1), so bigger is the vortex higher they will be. Reducing total pressure losses is the second goal of this optimization, and to represent this leakage the Pressure Recovery coefficient (PR) was chosen, and it is defined as:

$$PR = \frac{P_{tot,out}}{P_{tot,in}}$$

PR is the ration between inlet and outlet total pressure values. It doesn't describe where these losses happens because it is an area-average evaluation, but it gives an idea of how good is the design of the intake. Obviously we want PR to be as high as possible, but to

have a minimizing problem for both of the objective functions C_p has been defined as:

$$C_p = 1 - PR$$

Thus C_p and the swirl angle α are the objective functions for both of the optimization algorithms used for this research.

2.1.3 Flow Separation

A stationary object exposed to a moving fluid, or alternatively a solid objects traveling through a fluid, acquire a boundary layer around it where viscous forces occur in the layer of fluid close to the solid surface. Flow separation occurs when the boundary layer travels far enough against an adverse pressure gradient that the speed of the boundary layer relative to the object falls almost to zero.[8] [9] The fluid flow becomes detached from the surface of the object, and it takes the forms of a vortex. Flow separation results in increased drag and leakages in general, and for this reason is in our interest to confine this phenomenon as much as possible.

In our specific case, the flow separation occur after the first bend as shown in Figure 2.8 and 2.9. These images have been taken from the Wellborn experiment that will be discussed soon. The flow separation happen because the flow experience an adverse pressure gradient because of the increasing cross-section area of the intake. The 3D trajectories of the streamlines of the flow are displayed in Figure 2.7 as they take not very intuitive paths. The separation region is a 3D complex singularity, and from literature, the flow presents two symmetrical negative bifurcations converging into a spiral node. [10]

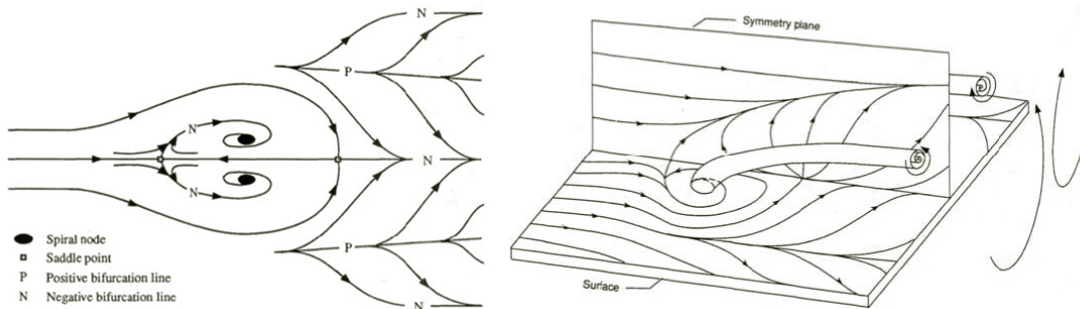


Figure 2.7: 3D representation of flow separation in an S-Duct [10]

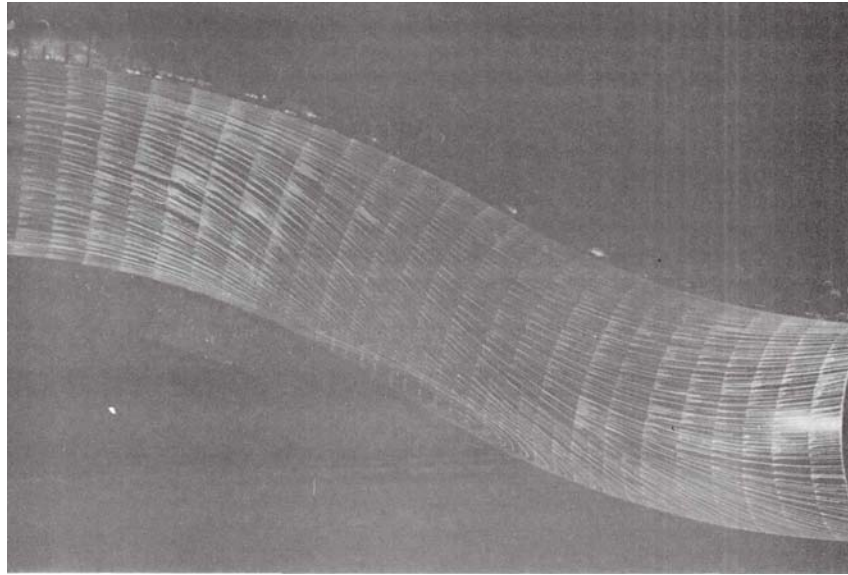


Figure 2.8: *Surface oil flow patterns on one symmetric half of the duct surface* [10]



Figure 2.9: *Close up of the surface oil flow patterns developed in the separation region* [10]

2.1.4 Flow Distortion Consequences

Our goal to reduce the flow distortion is mainly driven by the fact that the compressor needs to be fed with a uniform flow. The presence of a flow distortion affects the efficiency of the compressor downstream the duct, especially when these perturbations bring

to compressor stall. This phenomenon usually happen when the flow rate through a compressor blade row is decreased while the rotor speed is maintained constant, causing angle of attack on the blades to increase until flow separation will occur. However this can easily happen because of the non uniformity of the velocity of the flow, that reaches the outlet of the intake with a different direction other then that for which the compressor blades had been designed. This condition is similar in some respects to the stall of an isolated airfoil; differences are due to the adjacent blading in a cascade. Stalled operation of a compressor is accompanied by a drastic decrease in efficiency and excessive oscillating blade loads capable of causing structural failure. Usually the flow through stalled blades is severely restricted, thus the flow is diverted around this blockage. Effectively this increase blade angle of attack on one side of the restriction while decreasing it on the other side as shown in Figure 2.10. Blade $n^{\circ}1$ will soon become unstalled while at blade $n^{\circ}5$ flow separation will commence. This condition is called "rotating stall" since it propagates circumferentially in a direction opposite to rotor rotation. When these condition become severe enough to couse net flow-rate fluctuations with time, the situtation is called "surge". [11]

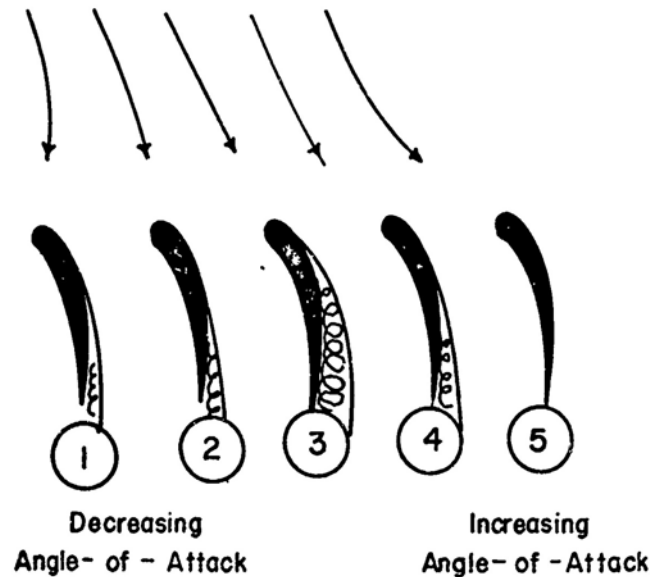


Figure 2.10: *Diverted flow during compressor stall* [11]

2.2 Prior Studies on S-Ducts

Since the beginning of S-Ducts application for civil and military vehicles, they have been studied in many different ways. In the past, the only way to study these intakes was to build prototypes and test them physically. This method though, slow and expensive, have been outclassed from CFD simulations with the help of more powerful computers and software through the years. These software simulate the flow with mathematical models previously validated through empirical experiments.

2.2.1 Experimental Researches

Weske [12] carried out the first experiments on ducts with a bend. His work aimed to understand the characteristics of the flowfield, thus the nature of the flow separation and of the vortices. He performed various test with different velocities of the stream, from 30 to 90m/s and different shapes of the cross-section of the ducts. An other important work was done by Bansod [13] because he was the first that showed the presence of two counter-rotating vortexes at the outlet. He also analyzed several parameters ad how they changed going through the separate region, like static and total pressure. the most relevant research in this field was carried out by Wellborn [10] in 1993. He investigated the compressible flow through an S-Duct at NASA Lewis Research Centre, providing a big amount of data about the behavior of the flow. His work is very useful for this research because he explained how the flow separation occur, and in fact, to validate the CFD software the Wellborn's work is taken as reference. On top of that, the geometry he used is very similar to the one used on for these optimizations, making this research even more important.

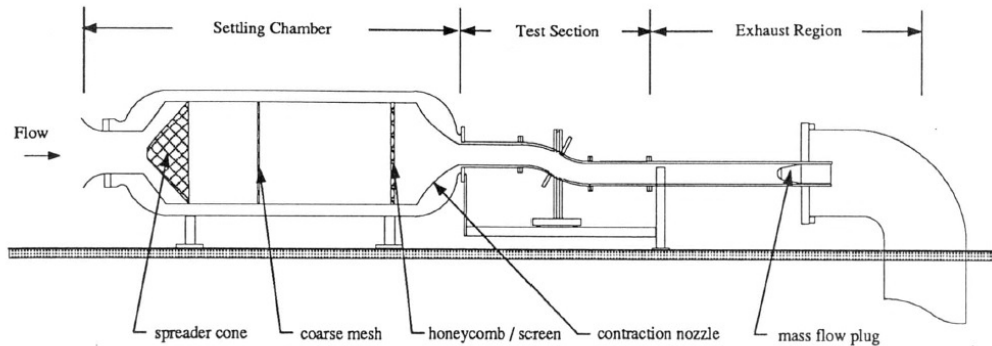


Figure 2.11: Facility scheme of Wellborn's experiment [10]

Figure 2.11 is the representation of the facility used by Wellborn for his experiment. At the beginning the flow has to go through a series of wires, meshes and screens within

the settling chamber, where the non universality of the flow are reduced as much as possible. First of all a perforated spreader cone mix the inlet flow, then the air goes through a coarse mesh screen and at the end a combination of honeycomb screens remove a large scale of the turbulence fluctuations. So, as the name already say, the settling chamber is used to set the flow for the s-ducts downstream. After this, the flow needs to be accelerated, and that is done with a simple contraction of the chamber. An area restriction of 59 was used to ensure a low turbulence intensity flow field at the inlet. The real test is performed in a diffusing S-Duct preceded and followed by a circular constant area extension 76.2 cm long. Then, an exhaust region containing a circular pipe, a mass flow plug and a sub-atmospheric plenum was used to ensure no downstream influence.

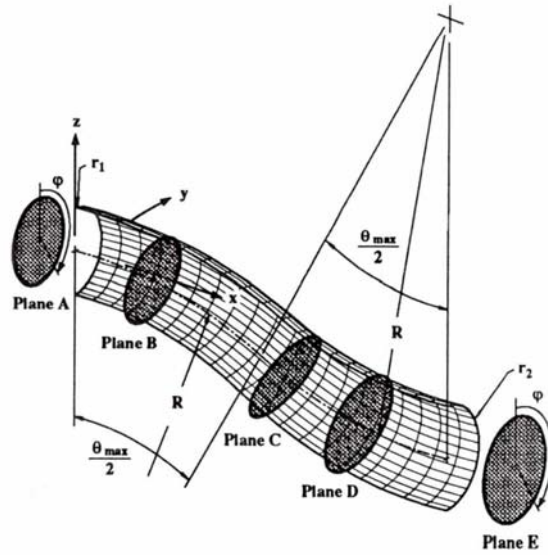


Figure 2.12: *Half shell of the circular diffusing S-Duct* [10]

Figure 2.12 is the description of the geometry used for the Wellborn's research. The S-Duct was designed to produce a number of complex three dimensional flow features, like separation, which happen almost every time with this configuration and the like. To investigate the flowfield behavior five planes, perpendicular to the centerline, was chosen starting from the beginning of the curvature. The centerline of the duct was defined by two planar circular arcs with identical radii, R , of 102.1 cm. The duct inlet radius, r_1 , was 10.21 cm, whether the exit one, r_2 , was 12.57 cm, which produce an area ration of 1.52. The duct was made from two block of aluminum and it was polished to remove all the imperfections. The duration of the test at the desired conditions was of ten minutes with an inlet centerline Mach number of 0.6 and a Reynold number of 2.6×10^6 . The flow was investigated by a variety of methods. The internal flow was measured with three and

five-hole probes, and the data was collected in the five planes already mentioned. To study the flow near the walls a fluorescent oil was applied to the surface to visualize the flow lines. For the surface static pressure instead, a total of 220 taps were positioned axially at constant angles $\phi = 10^\circ, 90^\circ$ and 170° and also circumferentially at the first four planes.

Three conclusion were made from the streakline patterns observed. First, the flow was indeed symmetric. This was confirmed because applying different color dye to each symmetric half of the intake shown that each part remain separate. Second, a large region of separated flow existed, as shown in figure 2.8 and Figure 2.9. The separated region consisted of two saddle points, occurring on the duct split line, and two spiral nodes, lying in each symmetric duct half. Third, the boundary layer cross flows are present.

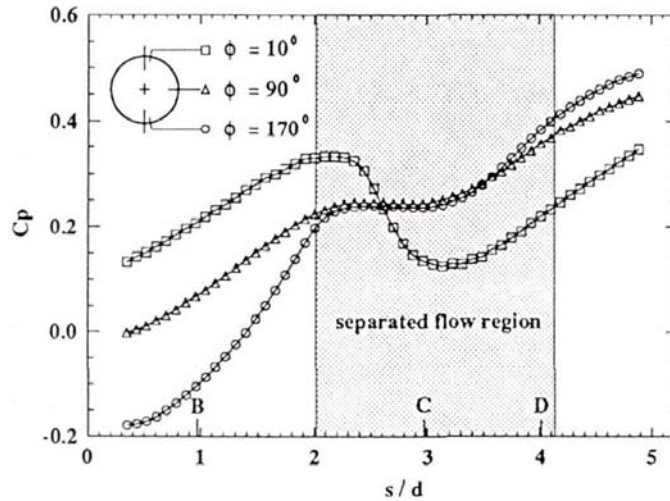


Figure 2.13: Axial distribution of surface static pressures for three circumferential positions [10]

Figure 2.13 represent the static pressure variation moving from the inlet to the outlet at three different "heights" within the duct. Two things appear clearly from the chart, the effect of the diffusion which increase the pressure, and the effect of the flow separation at the first bend. As expected the pressure of the upper section of the duct drops because the velocity of the flow is increased, as already said, due to the blockage of the flow separation bubble.

Figure 2.14 displays total pressure contours distribution in the five planes taken as reference. It is quite clear that a large region of low momentum fluid is developing in the last three planes. Here the boundary layer detach from the walls and goes towards the duct core, obstructing a big part of the cross section. As the flow progress towards the outlet, especially in the final plane, a pair of strong counter-rotating vortices appears,

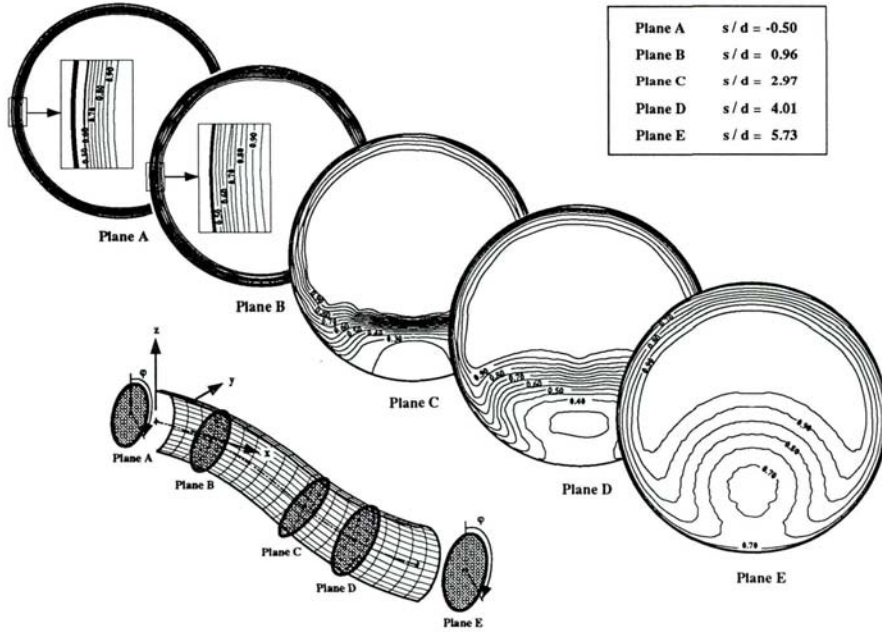


Figure 2.14: Total pressures in the five measurement planes [10]

pushing the low momentum fluid from the boundary layer to the center decreasing even more the univormity and the magnitude of total pressure.

2.2.2 Computational Research

Nowadays CFD solver has allowed to study the behavior of the flowfield through the duct with accuracy, and in fact, in the last few years a lot of studies on S-Ducts can be found in literature. As it is with the empirical experiments, even with the CFD ones there are several previous researches useful for this thesis. These works helped to define the best meshes, turbulence model and CFD settings for a proper flow simulation. For example Smith [14], in 1992, compared the CFD simulation of a S-Duct between the H-Grid and the O-Grid meshes, founding similar results. An other important investigation to understand which are the best settings for a CFD simulation was published by Delot [15] in 2006. She used a scaled up model from the Wellborn's one, maintaining the same geometry ratios. Figure 2.15 shows the scheme of her experiment. Using pressure taps she did steady and unsteady measurements during the tests. Afterwards a lot of different computational tests were carried out to find the best set up, able to obtain results as near as possible to those obtained from the experiments. During this tests a lot of different solver was tried, and among all the solvers, FLUENT matches the separation region better then the others. On top of that, this software predicted extreamly well the low pressure region on the

lower surface. However it was not so accurate at the outlet, where the boundary layer behavior in the upper surface is under-estimated, causing a pressure recovery higher than the experimental data.

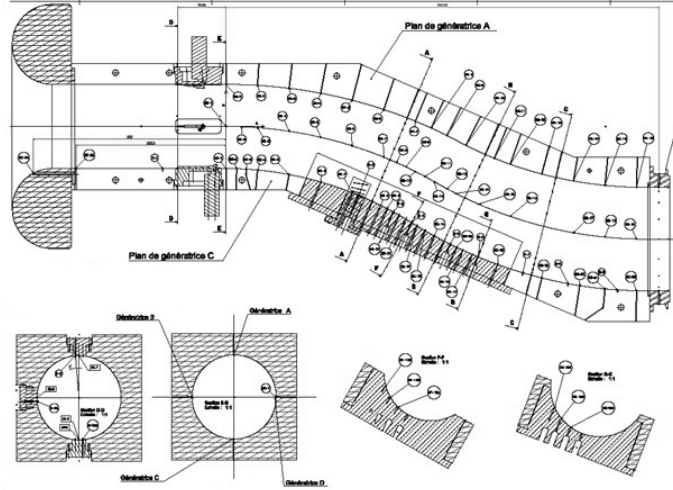


Figure 2.15: Delot's S-Duct model

One of the most recent publication about S-Duct intake simulation is the one carried out by Fiola and Agarwall [16]. They used only ANSYS FLUENT as solver for their tests. Their main goal was to compare different type of turbulence model, like Spalart-Allmaras, $k-\epsilon$, $k-\omega$ SST and transition shear stress transport model. In order to choose the best one, they compared the results with the Wellborn's experimental data, and they decided the $k-\omega$ SST as the most accurate, especially for PR leakages along the centerline of the duct.

A very important work for this thesis is the Optimization work done by Aurora Rigobello [17]. She improved the geometry parametrization of the previous optimization works increasing the number of decision variables in order to explore new configurations, and her model has been used for this research. She started from Guglielmi's work [18], which actually was the starting point for other two thesis carried out by Enrico Manca [19] and Marco Barison [20]. Guglielmi used two objective function, 1-PR and α , the same of this work and Rigobello one, and he analyzed three different type of S-Ducts which were optimized respectively for the best pressure recovery, for the lower swirl angle of the flow and the best trade-off solution.

Enrico Manca's work [19] focused on unsteady simulations of flow (DDES) through an S-Duct, especially for an optimized intake with the NSGA-II algorithm, to determine if the optimization has really improved the shape of the diffuser or not, since it was carried out

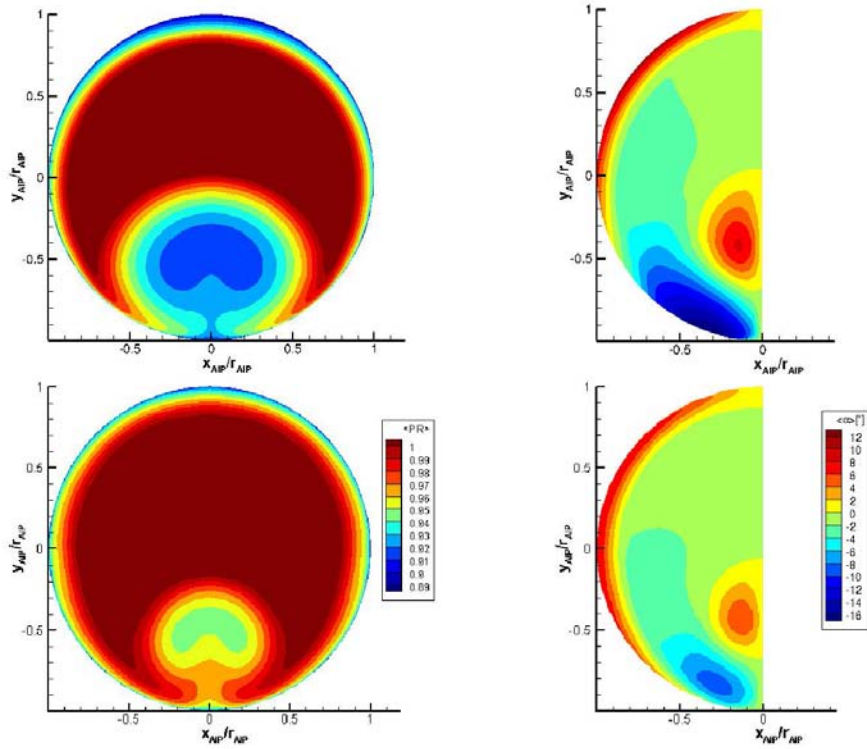


Figure 2.16: *PR distributions (on the left) and Swirl distributions (on the right) comparison between the baseline and an optimized design.* [18]

only with steady simulations. He found that the simplifications used for that optimization, like using only half of the duct, reducing the inlet length or using a medium size mesh didn't compromise the goodness of the results. Marco Barison's work [20] instead, studied how well surrogate models can predict the flow behavior within an S-Duct. He used those models to evaluate the objective functions for a multi-objective optimization with a genetic algorithm.

All the previous works focused on improving the parametrization of the problem, the model used to evaluate the objective functions or to find the best solver to simulate the flow through the duct, but a comparison to understand which optimization algorithm suits better this kind of research has never been done. Different algorithms have been used on these works, but all of them do not start from the same description of the problem like in this thesis, making the comparison of the results from one to another almost impossible. Thus the aim of this thesis is to use the most recent geometry parametrization and the best tools available today after all of the researches, and compare the results from two very different optimization algorithms.

Chapter 3

Multi-Objective Optimization

Optimization is a method of obtaining the best results under given circumstances, and because of the growing complexity in engineering design activities, there is no single method for all the optimization problems efficiently. That is the reason why this thesis what to compare two different algorithms to see which one is better for S-Duct intakes with the parametrization presented in the next chapters. An Optimization can have only one goal, for example to minimize the cost of production or to maximize the efficiency of the production. However the most common cases are the one in which two ore more goal has to be achieved simultaneously, they are called Multi-Objective Optimization Problems (MOOP). These goals can be normally expressed as mathematical functions named *Objective Functions*, which depends on a set of project variables. These variables describe the problem entirely, and the whole problem can be solved changing this parameters. Differently from a single objective function problem, a MOOP does not have an optimal solution capable of meeting both of the objective functions. Generally MOOP present two or more goal which has to be minimized, while some constrains limit the solutions range, meaning that the decision variable cannot change freely. However if the MOOP need to maximize its objective function, it can be easily transformed in a minimization problem since the maximum of a function $f(x)$ is exactly the minimum of $-f(x)$. Let's see how a MOOP

can be mathematically expressed:

$$\begin{aligned}
 &\text{Find } \mathbf{x} = x_1, \dots, x_n \text{ that} \\
 &\text{Minimize } f_j(\mathbf{x}) \quad j = 1, \dots, m \\
 &\text{Subject to :} \\
 &g_l \geq 0 \quad \forall l = 1, \dots, L \\
 &h_k = 0 \quad \forall k = 1, \dots, K \\
 &x_iL < x_i < x_iU \quad \forall i = 1, \dots, n
 \end{aligned} \tag{3.1}$$

g_l and h_k are the constraints, and $f_j(\mathbf{x})$ are the objective functions. A visual representation of a MOOP can be seen in Figure 3.1, where x_1 and x_2 are the two design variables.

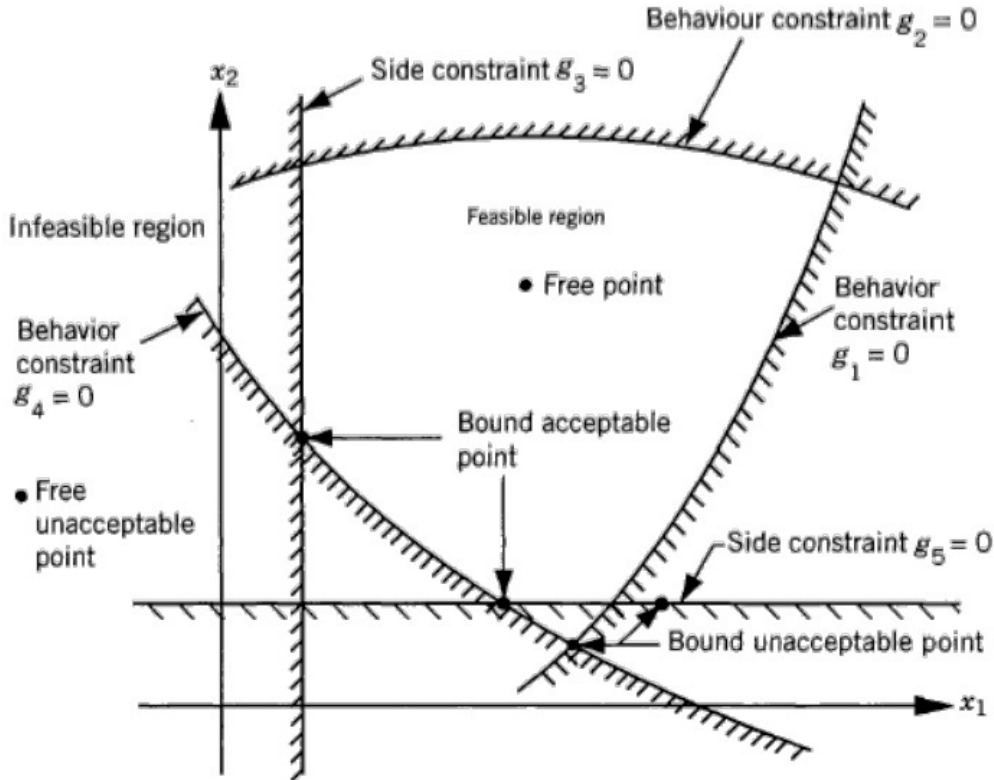


Figure 3.1: Design space of a MOOP divided in feasible and infeasible space

Constraints on the decision variables are called *Side Constraints*, and they define the boundaries for each project parameter. Therefore the design space is divided into two regions, the *Feasible* and the *Infeasible*.

3.0.1 Concept of Dominance

As already mentioned it is easy to find the best solution for a single objective function problem, and it is equally easy to rank its solutions. This cannot be said for MOOP, in which there are conflicting targets that prevent to univocally define the best solution. For this reason it is necessary to introduce the concept of Dominance and Pareto Front. The dominance between two solutions is defined as follow:

In a minimization problem the solution **A** dominates a solution **B**, if these statements are verified:

- The solution **A** is not worst then the solution **B** in each objective function. Which means $f_j(\mathbf{A}) \leq f_j(\mathbf{B}) \quad \forall j = 1, \dots, m$ where m is the number of the objective functions.
- The solution **A** is strictly better then the solution **B**, in at least one objective function. Which means that $f_k(\mathbf{A}) < f_k(\mathbf{B})$ for at least one k in $1, \dots, m$;

It is possible that **A** does not dominate **B** and vice versa. In this case both of them are *non-dominated* solutions. In Figure 3.2 it is possible to understand this concept. f_1 and f_2 are the objective functions.

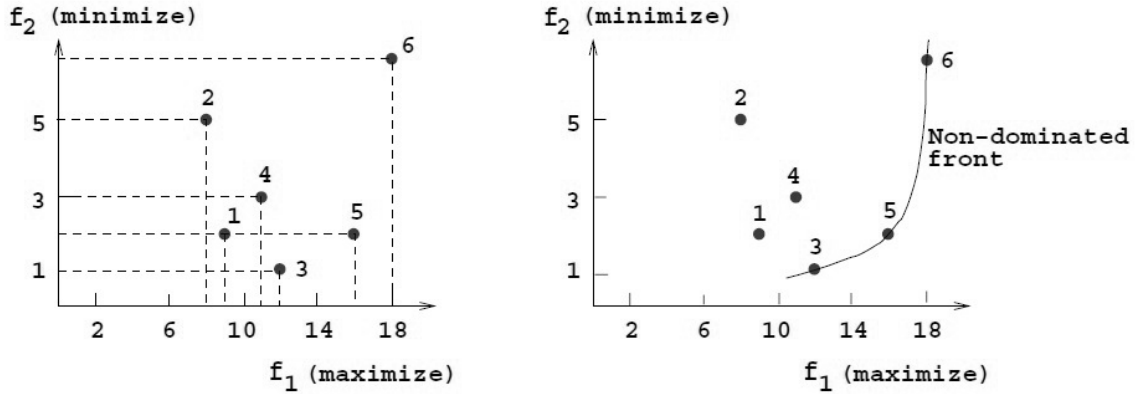


Figure 3.2: Visual representation of the Pareto Front

During the optimization, the best solutions are those that are not dominated by any other point of the current solutions. Those points that belongs to the non-dominated set are considered as optimal. Once taken a non-dominated solution, it is not possible to improve one of its objective functions without worsening the others. Two definitions are necessary to make this concept more clear, and to understand better the next chapters:

- **Pareto Optimal Set:** It is the set of non-dominated solutions, it is a subspace of the decision variables domain.
- **Pareto Front:** It is the image set of the Pareto Optimal set, it contains the objective functions corresponding to the non-dominated solutions. In Figure 3.2 the non-dominated points 3,5 and 6 belongs to the Pareto Front.

3.0.2 Resolution methods

There are several optimization procedures with different advantages and drawbacks in order to solve a MOOP. There are classical methods, but they tend to converge very slowly because they transform the MOOP in a single-objective problem, finding only one optimal solution. Genetic Algorithms (GA), Simulating Annealing (SA) and Tabu Search (TS) instead are the most advanced and efficient one. These algorithms are able to predict the real Pareto Front, and they can provide to the engineers a lot of different trade-off solutions, giving them the possibility to choose the final solution among the Pareto Optimal set. In the next chapter, two advanced methods will be described, respectively Tabu Search and a Genetic Algorithm (GDEA-II).

Chapter 4

Optimization Algorithms

An optimization process is a loop that goes through a lot of steps that can be organized according to their function as shown in Figure 4.1. This scheme is a simplification of the optimizations carried out in this thesis, which should help to understand the entire process. It can be read starting from the geometry creation, where a python code create it from the definition of 36 variables, then ICEM creates a mesh for the geometry, FLUENT simulate the flow through it, and then α and $1 - PR$, the two objective functions, are evaluated. These two results are communicated to the optimizer which associate them with the variables that describes that specific intake. Based on the previous results, the optimizer choose how to change those 36 variables to obtain an intake with better performances, and then the loop start again. Even if these two algorithms are not yet described, it is obvious that the brain of the entire optimization process resides into the optimizer, which in fact is the most important part of the loop. That is why in the next sections Tabu Search and GeDEA-II will be seen in depth.

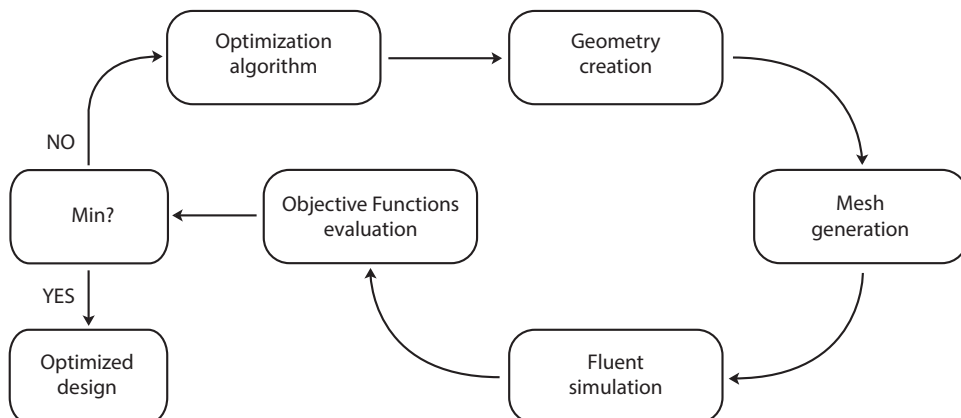


Figure 4.1: Simplification scheme of the optimizations carried out in this thesis

4.1 Tabu Search

The basic concept of tabu search is a meta-heuristic applied to the other heuristic [23]. The overall approach is to avoid entrainment in cycles by forbidding or penalizing moves which take the solution, in the next iteration, to points in the solution space previously visited (hence "tabu") [24]. This method is based on procedures thought to cross the boundaries of feasibility or local optimality, which were usually viewed as a barrier, in order to find a descent direction faster. The term meta-heuristic is commonly employed to refer to a procedure able to guide an heuristic into the design space in a more efficient way. An heuristic may consist in fact in a simple set of instructions which explore the space without a logical path. This approach is obviously inefficient and resource consuming. The design space is usually too big to be explored easily, even if a HPC based on a cluster is used for the optimization cycle. Tabu Search as other meta-heuristics introduce a component of intelligence in this search. According to the Tabu Search designer, problem solving can be defined as "intelligent" only if it takes advantage of an adaptive memory (that has the following attributes: recency, frequency, quality and influence [21]) and a responsive exploration (diversification and intensification). Tabu Search has both these qualities. Specifically it uses three types of memory in order to adjust continuously its research path. Short Term Memory (STM) for the *Tabu Elements*, Medium Term Memory for intensification and Long Term Memory for diversification.

4.1.1 Multi-Objective Tabu Search (MOTS)

The interest for Tabu Search as Optimization Algorithm is due to the fact that most of the real-world optimization problems are solved using Genetic Algorithms or Simulated Annealing [22]. For this reason Jaeggi *et al.* [22] proposed an adaptations of the single objective Tabu Search implementation of Connor and Tilley [25], for multiple objectives problems, called MOTS (Multi Objective Tabu Search). With respect to the single objective method, the multi-objective ones are modified in the following parts:

- Search point comparison: based on the idea of Pareto dominance.
- Hooke and Jeeves (H&J) move.
- Optimal point archiving and MTM.
- Search Intensification and Restart strategy.

This method employ short, medium and long term memories and it is based on the Hooke and Jeeves move (*H&J*), which will be explained in the next section. MOTS uses a

straightforward solution. It groups at random all the new points ($2 \cdot n_{var}$) into equal subsets and analyzes one group at a time. As it finds a Pareto-dominant point in a group it uses this new point for the following *H&J* move, discarding the other subsets.

4.1.2 Algorithm Overview

Tabu Search works in an iterative manner. It start from a single point and then it search for the next one in the search space available, which has as many dimensions as the numbers of variables. TS moves from one point to the other with a specific action called ”*The Hooke and Jeeves move*”.

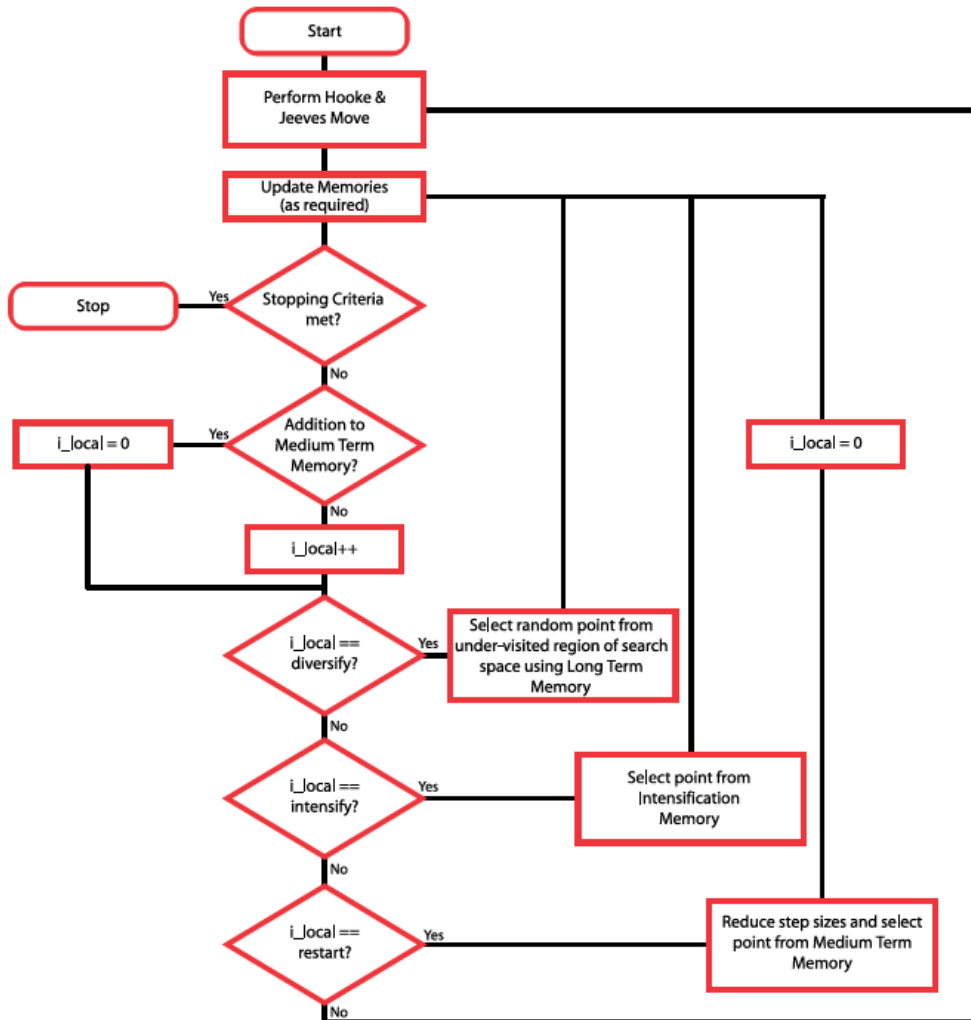


Figure 4.2: Flow chart for the $MOTS_{ii}$ algorithm.

The algorithm needs to remember its previous steps in order to know where to search in an intelligent way and to allow storage for the best solutions found. To understand better how the algorithm works is useful to follow the explanation on the flow chart in Figure 4.2. The design space is at first divided into a given number of regions. This operation is easily realized by dividing the range of each variable in smaller intervals. A local index is assigned to each region and is incremented every time a dominated point is found in that region. Recently visited points are stored in the *short memory term* (STM) and these are tabu, meaning that the search is not allowed to revisit these points. Optimal or near-optimal points are stored in the *medium term memory* (MTM) and are used for intensification. In this way the algorithm knows where are the areas with known good objective function values. The *long term memory* (LTM) instead, records the regions of the search space which are under-explored. A visual representation of the memories usage can be seen in Figure 4.3. Thus all the points are evaluated (through their objective functions) and discarded or stored in the appropriate memory as explained above. The next point is then selected among the Pareto-dominant, equivalent or dominated respectively. More in detail, once all the points have been analyzed the algorithm operates according to the following sequence:

1. A stopping criterion is checked immediately after the computation of the objective functions.
2. For each point taken into account by the *H&J* move its dominance is verified. If it is Pareto dominant or equivalent is stored in the MTM and the *i_local* index is set to zero. If Pareto-dominated it is discarded and the local index incremented by one.
3. Intensification criterion. It is usually set for a lower value of unsuccessful points (for example when $i_local = 15$). When the equality is verified, the corresponding region is demonstrating as not fruitful, as a consequence a Pareto dominant point from the MTM is randomly selected and chosen as the new element for the search. Local indexes are not reset, since if intensification should fail, diversification will be the next alternative. The *H&J* move is then substituted by this random choice.
4. Diversification criterion is verified. It is applied only if *i_local* is equal to a given threshold (i.e. $i_local = 30$). In this example intensification is actuated only after 30 consecutive points have been judged dominated. When this happens a point is selected from an under-visited region (according to the LTM) and the process continues from this new point. The *H&J* move is again substituted by this random choice. The *i_local* indexes are not reset. If the index value is different (smaller or greater) than the diversification value the algorithm move to the following step.

5. Restart and refining the step size. When both intensification and diversification do not reach a result, the restart option is actuated (for higher threshold, for example $i_{local} = 60$). A point is selected from the MTM and the step size reduced. Only in this last case the i_{local} counter returns to zero.

When a point pass through this sequence and none of the strategies is activated it becomes the new point and another $H&J$ move is made. Actually a pattern move is coupled to the $H&J$ move, as a consequence the same move is repeated two times before entering in the algorithm sequence. [22]

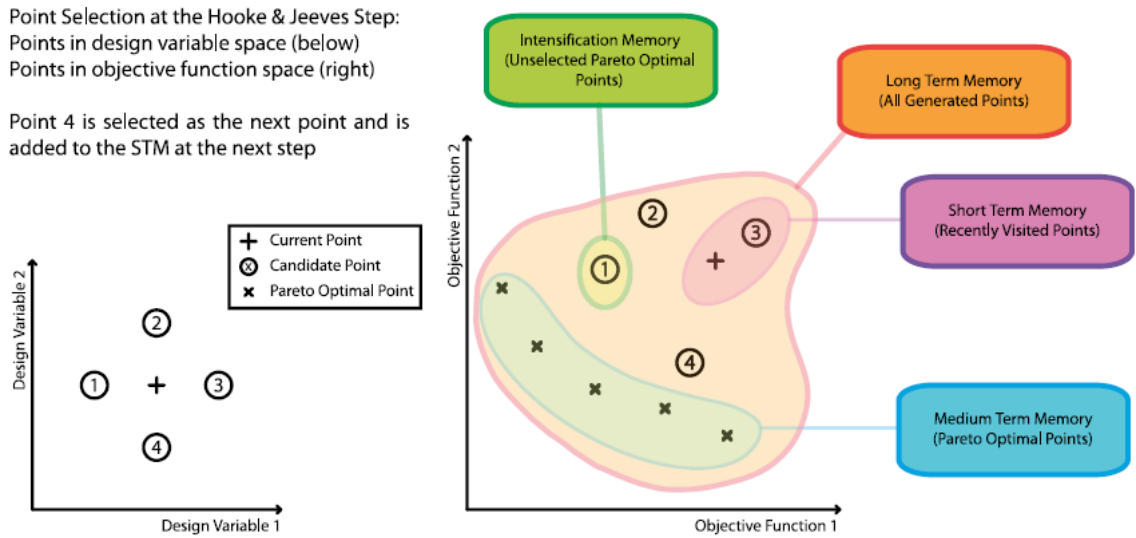


Figure 4.3: Point selection for the Hooke and Jeeves move and Tabu search memories [22]

4.1.3 Hooke and Jeeves Move

A quick look to this key characteristic of the algorithm need to be made. This move is made every iteration, and $2n_{var}$ new points are generated incrementing and decrementing each of the n_{var} design variables by a given step around the current point. However not all of these points are investigated, those that are Tabu are removed immediately, and only a sample of random points from those that remain are evaluated. This random choice is made to avoid any directional bias. The objective functions for these new points are evaluated and they are considered as candidates to be the next point in the search. In a single objective function problem, it is easy to evaluate which objective function is better then the others, but for a multi-objective optimization a similar logic can be applied only introducing the possibility of multiple points being Pareto-equivalent and optimal must be allowed for. If there is a single dominating point, it is automatically taken as the new

point. However if there are multiple dominating points, the dominated points within the group are removed and only one is selected at random for those remaining. The other points become candidates for intensification (discussed below). If there are no dominating points, the same procedure is applied to those candidates which are Pareto-equivalent to the current point. If there are none Pareto-equivalent, other points from those picked up for intensification are evaluated and the comparison is repeated. If all the points new points don't dominate the first one, a non dominated points is selected and the process start again. This move then is repeated an other time, and the new point is compared to the current, and, if it is better, it is taken as the new point. With this procedure the search may be accelerated along downhill directions. The basic search pattern is shown in Figure 4.3. [22]

4.1.4 Intensification Memory

It has not being said what happen to those points that are discarded when there are multiple dominating points during an $H&J$ move. It seems wasteful to not keep track of these points, and for this reason an intensification memory (IM) has been incorporated into this algorithm. This IM contain a set of Pareto-equivalent Points and it is updated every $H&J$ step. Every time that one point become dominated is removed, thus the IM should always hold points which are on, or near to, the current Pareto-optimal front. [22]

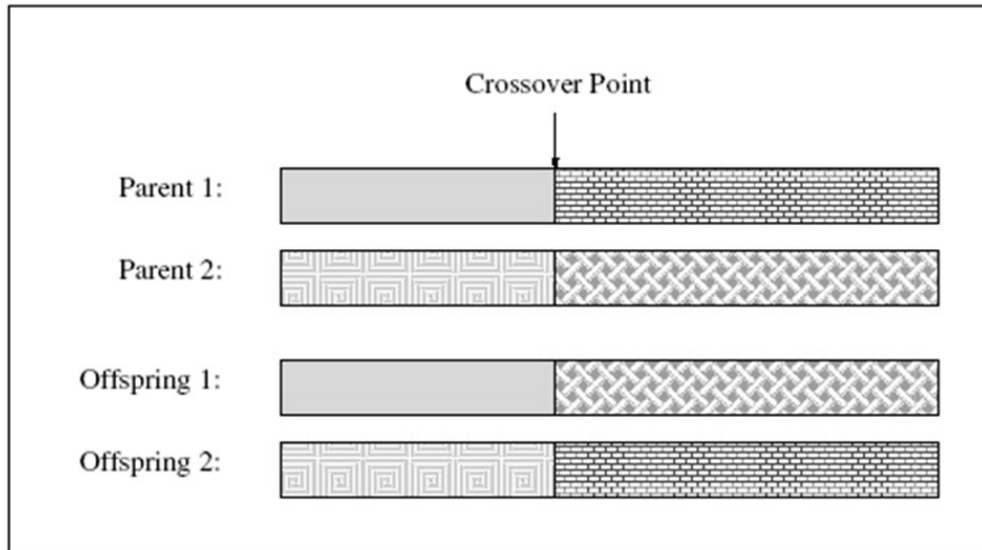
4.2 GDEA - II

The optimizer discussed in this section is a genetic algorithms that belongs to the larger class of evolutionary algorithms (EA). GDEA is the acronym of Genetic Diversity Evolutionary Algorithm, that is the algorithm built around a specific method used to solve MOOP, the Genetic Diversity Evaluation Method (GeDEM). In this section I am going to explain what are the GeDEM and the GDEA, and finally which changes have been done from the first version to the last and second one, the GDEA - II.

4.2.1 Multi-Objective Evolutionary Algorithms

Before discussing about GDEA there are some notions which must be introduced. Genetic algorithms (GA) are nature oriented methods, and for these reason some evolution rules has been defined in order to evolve the initial designs to the optimal ones. GA starts with a random set of solution which is also known as initial population. Then these solutions are evaluated and the respective fitness value is assigned to each one of them. Afterwords these values are checked to see if whether or not a termination criterion is reached. If not, the current generation is subjected to the following operations:

- **Selection:** This operator randomly chooses two solutions, called *parents*, which will generate two child, called *offspring*, using the next operator. The choice is not completely random, because those individuals with a better fitness value have a better chance to reproduce and to give birth to new children for the next generation. In this way the next generation should be better then the previous one.
- **Crossover:** Once the selection is made, the crossover operator mix the genetic information coming from the parents in order to create two children. During this operation it is important to not lose any genetic material, so the genome of the parents is only recombined. If the encoding is binary, an easy method is the single point crossover shown in Figure 4.4. The Crossover point, also called the break point, in both parents is randomly chosen, and the two part of the genome is recombined to create the offspring. It is possible to have different type of crossover, like the double point or the non homologous one. In case the encoding is real, the cross-over is a linear combination.
- **Mutation:** The main goal of this operator is to maintain genetic diversity of the generation. For each offspring, a small percentage of its genetic information is randomly modified. In this way the algorithm can escape easier from local minima, but it does not give always children better then the parents.

Figure 4.4: *Single point crossover*

These operations are repeated sequentially until a new generation is created.

4.2.2 The Genetic Diversity Evolutionary Method (GeDEM)

The multi-objective optimization process with evolutionary algorithms has two main objectives:

- the convergence to the Pareto-optimal set;
- the maintenance of genetic diversity within the population.

The method GeDEM is based on the idea to use these two objectives during the evaluation phase of the objective functions, and to rank the solutions emphasizing the non-dominated solutions as well as the the most genetically different one. In this way the selection pressure drives the optimization towards the Pareto front while keeping it spread as much as possible. Therefore this method helps with two very important issues that Multi Objective Evolutionary algorithms (MOEAs) have to face: the exploration of the search space, and prevent premature convergence. In fact, an individual's worth is not considered to rely only upon its objective function values, but also on its ability, when mating with other individuals, to generate offspring that are dispersed enough in the decision variable space, in order to explore the latter efficiently and exhaustively [26]. To understand if an individual has more chance to produce, when mating, an offspring in regions of the search space not well explored, its distance from the others of the population is checked. The bigger the

distance, the higher the chances. However measuring the distance in a multi-dimensional space is not simple, and there are different ways to express it. The most commons ones are the *Euclidean distance* in the decision variable or objective function space and the *Hamming distance* in the string space. On top of this the measures of the diversity of an individual include the sum of its distances from all the other individuals, or the minimum distance from another one.

That begin said, the GeDEM ranks the solutions maximizing:

- the ranks scored from individuals with better values of the objective functions, as in the original MOOP, giving to the non-dominated the highest rank;
- the ranks assigned to those individuals with higher genetic diversity, according to the chosen distance metric.

In Figure 4.5 is shows how GeDEM ranks the solutions. The individual marked with the label "lethals" are those who have a poor rank and a some neighbor around them, meaning that they will not be helpful for the next generations.

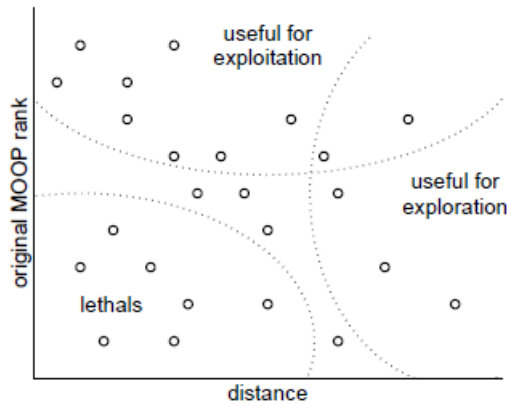


Figure 4.5: *GeDEM fitness assignment* [26]

When using GeDEM the Pareto dominance definition given in the previous chapter has to be slightly modify. In the original MOOP ranks system, individuals with the same solutions should not dominate one another, independent of their distance-based diversity measure. Thus, the definition of dominance used with this method is: Vector $\mathbf{u}=(rank_u, dist_u)$ dominates vector $\mathbf{v}=(rank_v, dist_v)$ if and only if $rank_u > rank_v \wedge dist_u > dist_v$.

Figure 4.6 shows in a simple way how this modification affect the Pareto front.

An other feature of GeDEM different from the other methods is the approach with clones. Clones are individual genetically identical, therefore they represent the same solution, they will have the same objective functions values and even the same measure of

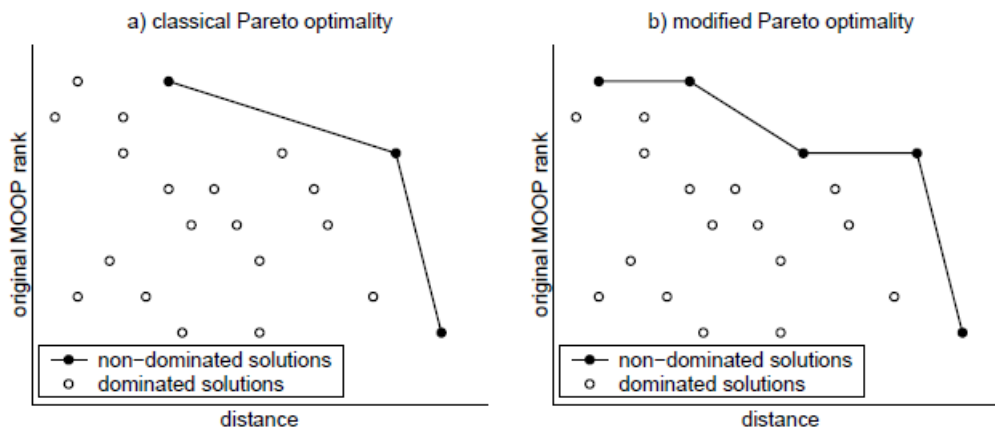


Figure 4.6: *Modification of the Pareto optimality definition in the application of GeDEM [26]*

diversity. Their presence is harmful for the optimization because it takes an useless computation effort to re-evaluate the objective functions of the same individuals more than once, and it affects also the rank of the single solution. GeDEM in fact will give a lower rank to two identical solutions than the same non-cloned, because of the diversity drop due to their proximity. This choice of removing clones helps to avoid having them in the next generations because the recombination of clones will give an offspring that is a clone again. To be more specific clones are not removed, but replaced with a randomly generated individual to encourage the exploration of the search space.

4.2.3 The Genetic Diversity Evolutionary Algorithm (GDEA)

The Genetic Diversity Evolutionary Algorithm (GDEA) is a framework strictly designed around GeDEM to exalt its characteristic [26]. This algorithm follows the basic features of GeDEM when it comes to replacing clones or elitist mechanisms (Schwefel, 1995; Bäck, 1996). The structure of this algorithm follows the Evolution Strategy ($\mu + \lambda$) (Schwefel, 1995; Bäck, 1996), and these are the main steps that are repeated for every generation: [26]

Step 1: An initial population of μ individuals is generated at random.

Step 2: A mating pool of 2λ individuals is formed, each individual having the same probability of being selected.

Step 3: λ offspring are generated by crossover. Some bits of the offspring are also randomly mutated with a probability p_{mut}

- Step 4:* The whole population of $\mu + \lambda$ individuals is checked to discover possible clones. These clones are removed and replaced with new randomly generated individuals (this is done to encourage the exploration of the search space and also to have the algorithm evaluate, for convenience, new λ different offspring every generation; still the occurrence of clones' birth is not so frequent if clones are removed generation after generation).
- Step 5:* The objective function values of the $\mu + \lambda$ individuals are evaluated and the non-dominated sorting procedure by Goldberg (1989) is performed to assign the ranks to the solutions according to the objectives of the MOOP.
- Step 6:* The whole population of $\mu + \lambda$ individuals is processed to determine the value of the distance-based genetic diversity measure for each individual.
- Step 7:* GeDEM is applied according to the ranks scored in Step 5 and the values of the diversity measure assigned in Step 6. The non-dominated sorting procedure by Goldberg (1989) is used to assign the ranks.
- Step 8:* The best μ solutions among parents and offspring, according to the ranks assigned in Step 7 by GeDEM, are selected for survival and the remaining λ are eliminated.
- Step 9:* If the maximum number of generations is reached then stop, else go to Step 2.

4.2.4 Differences from GDEA to GDEA - II

An improved version of GDEA was used for this thesis, named GDEA - II, and it was made with a novel crossover operator, The Simplex-Crossover (SPX), a novel mutator operator, the Shrink-Mutation, and its new Tournament-Selection operator. While the optimization algorithm was improved, GeDEM operator was left unchanged and completed using the non-dominated-sorting based on crowding distance. [27] GDEA-II aimed at reducing the potential weakness of its first version and competitors, without losing its superior performances which consist in a good balance between exploration and exploitation.

Simplex-crossover is the main improvement of this algorithm, and it is the novel crossover function. This operator was already proposed in [28] as a new multi-parent recombination operator for real-coded GAs. In that research it showed to perform well on functions having multimodality and/or epistasis with a medium number of parents. However it was not employed in multi-objective problems, and it was not thought to take into account the fitness of the objective functions as the driving force of the simplex. Thus

this is the first genetic algorithm to integrate an operator based on SPX with some new features.

To understand better the SPX exploited in the GDEA-II, it is better to spend some word to elucidate the Simplex algorithm. A simplex in n -dimensions is a construct consisting of $n+1$ solutions x_k , $k = 1, 2, \dots, n+1$. [29] In a two dimensional plane it is easy to visualize its shape, in fact it corresponds to a triangle. The solutions are evaluated in each step and the worst solution \mathbf{w} , which could be the one with the highest value of the objective functions, is identified. Then the centroid \mathbf{M} of the remaining points is defined as $\mathbf{M} = \frac{1}{n} \sum_{k=1}^n x_k$ (k identifying the two best solutions) and a new solution \mathbf{r} , which replaces \mathbf{w} , is obtained by reflection, $\mathbf{r} = \mathbf{M} + (\mathbf{M} - \mathbf{w})$. Figure 4.7 shows the reflection step applied to a problem in \mathbb{R}^2 .

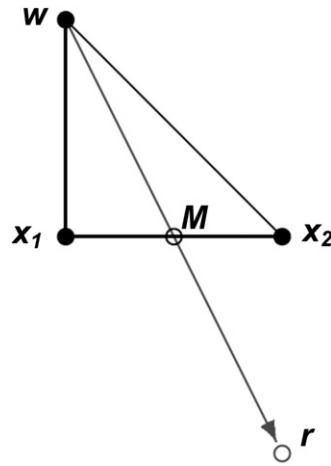


Figure 4.7: The reflect step of the simplex algorithm applied to a problem in \mathbb{R}^2 [27]

\mathbf{w} as the worst point is replaced by \mathbf{r} , and \mathbf{M} is the centroid of the two other points \mathbf{x}_1 and \mathbf{x}_2 .

GDEA-II employ the Simplex concept just discussed as the crossover operator, in order to make the evolution process faster. Crossover functions are very important for EAs since they combine two individuals, or parents, to get back a new child for the next generation. Since the Simplex is an optimization algorithm itself, the children should have better fitness values compared to the one of the parents. The Simplex used with GDEA-II however requires only two parents to form a new child (two is the minimum number for a simplex). These two parents are selected according to the selection procedure from the previous population, and combined following the guideline of the simplex algorithm.

The Shrink-Mutation operator is introduced in GDEA-II. Normally, in genetic algorithms, mutation operator makes small changes in the individuals in order to create

mutated children and to explore the search space better. In the GDEA algorithm some bits of the offspring were randomly mutated with a probability p_{mut} . With the Shrink-Mutation instead, a random number taken from a Gaussian distribution with mean equal to the original value of each decision variable of the parent vector is added.

$$Shrink_{i,ignr} := Shrink_{i,ignr-1} \cdot \left(1 - \frac{ignr}{ngnr}\right)$$

Where $Shrink_{i,ignr}$ is a number representing the current mutation range allowed for the i th design variable, $ignr$ represent the current generation and $ngnr$ the total number of generations. The shape of the shrinking curve was decided after several experimental tests. [27] An other important feature of this kind of mutation is that the variation is zero at the last generation. This mean that the mutation allows to explore, as much as possible, the design space during the first part of the optimization, while exploiting the non-dominated solutions during the second part. When an individual is picked up to be mutated, one decision variable is randomly selected and it is mutated according to the following formula (results of an experimental campaign):

$$Child_{i,mut} := Child_{i,cross} + [(\sqrt{Child_{i,ignr}} \cdot random) \cdot Child_{i,ignr}]$$

where $Child_{i,mut}$ is the mutated decision variable, $Child_{i,cross}$ is the i th decision variable of the child generated by the previously introduced crossover operator and $random$ is a random number taken from a normal distribution between -1 and +1. [27]. Therefore an initial *Shrink* factor is defined (for example it can be equal to ten percent of the variation tange of the design variables) and it will decrease with the advancement of the optimization until it will be equal to zero at the last generation. Then, unlike the crossover operator which generate all the offspring, the mutation is applied only to some individuals of the offspring population in order to deeply explore the design space.

The Tournament Selection is the selection operator that is based on the tournament concept [30]. During the optimization it will not be efficient to undergo to the crossover and to the mutation operators only the last generation, because the genotype of the best individuals of the previous generations will be lost, and the computational effort to evaluate all the population every time is too high. To solve this problem there must be a selection that chooses the individuals eligible for the crossover and the mutation modifications. The Tournament Selection uses a random process in which several parents are chosen from the last generation to play a tournament. The number of parents that take part of this "game" is called *tournament size*. The ranking of the individuals playing are compared, and those with the higher one will be retained to be subjected to crossover and

mutation. In case of ties the parent selected first is retained, while the second is moved into a temporary holding array. This tournament continues until the number of individuals will equal the generation size.

The last modification from the first to the second version of this algorithm, it is not made on the algorithm itself, but on its diversity preservation mechanism, the GeDEM, which is the same as the one used for GDEA. However GDEA-II uses not only this method, but also the *non-dominated sorting* based on crowding distance. Both of the mechanisms are employed for the optimization because each one of them can take advantages from the other.

The main difference between these methods is the definition of dominance. For GeDEM dominance is defined as:

Vector $\mathbf{u}=(rank_u, dist_u)$ dominates vector $\mathbf{v}=(rank_v, dist_v)$ if and only if:
 $rank_u > rank_v \wedge (dist_u \geq dist_v)$.

While for the non-dominated sorting based on crowding distance the definition of dominance is:

Vector $\mathbf{u}=(rank_u, dist_u)$ dominates vector $\mathbf{v}=(rank_v, dist_v)$ if and only if:
 $(rank_u > rank_v) \vee [(rank_u = rank_v) \wedge (dist_u \geq dist_v)]$.

The difference is not huge, but it is clear that the condition of GeDEM is more restrictive, thus, it will create less non-dominated points. This characteristic drives the evolution faster towards the Pareto front, but on the other hand, the non-dominated sorting tends to create more non-dominated individuals meaning that it will result in a better Pareto front coverage. For this reason the diversity preservation is accomplished by means of GeDEM for the first three quarters of the generations, whereas for the rest of the generations the non-dominated sorting mechanism is exploited.

Chapter 5

Methodology

The work in this thesis want to compare the optimizations carried out with two different optimization algorithms, GeDEA-II and Tabu Search. The shape of the duct is optimized in order to obtain the best design capable of minimizing total pressure leakage and flow distortions. In fact, the objective functions for these optimizations are the pressure recovery coefficient and the swirl angle at the Aerodynamic Interface Plane (AIP). Both of the processes required a huge amount of computational power, and for this reason two clusters were necessary to reduce the time needed for each iteration, Astral for GeDEA-II and Grid for Tabu Search. Both of these two clusters are located in Cranfield University. As mentioned before, this research starts from the Rigobello's work that has been very useful, especially for the parameterization of the geometry. This chapter will explain in detail the tools and the models used to achieve these optimization processes.

5.1 Geometry

In this section there will be the description of the geometry baseline used. It is the one presented by Delot [15] in 2006, designed at ONERA, French Center of Aerospace Research and it is a scaled up version of the Wellborn's duct of 1992 [10] visible in Figure 5.1

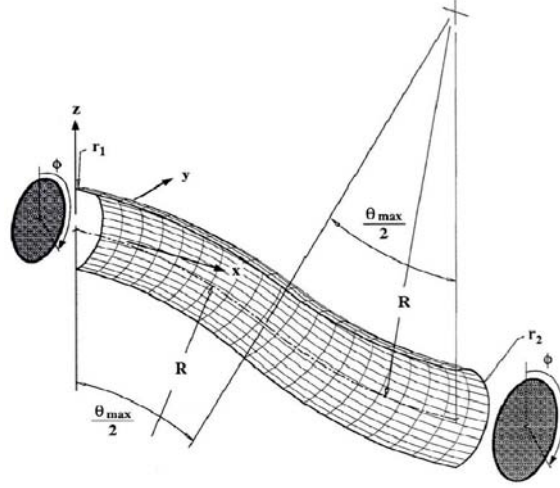


Figure 5.1: *S-Duct geometry* [10]

The centerline is defined by two circular arcs lying on the $y - z$ plane with identical radii, R , of 665mm and a subtended angles, θ_{max} , of 60° . The centerline coordinates, given by the Equations (5.1), (5.2), and (5.3), are indicated by the dashed line in Figure 5.1.

$$x_{cl} = 0 \quad (5.1)$$

$$y_{cl} = \begin{cases} R \sin \theta & 0 \leq \theta \leq \theta_{max}/2 \\ 2R \sin\left(\frac{\theta_{max}}{2}\right) - R \sin(\theta_{max} - \theta) & \theta_{max}/2 \leq \theta \leq \theta_{max} \end{cases} \quad (5.2)$$

$$z_{cl} = \begin{cases} R \sin \theta - R & 0 \leq \theta \leq \theta_{max}/2 \\ 2R \cos\left(\frac{\theta_{max}}{2}\right) - R(1 + \cos(\theta_{max} - \theta)) & \theta_{max}/2 \leq \theta \leq \theta_{max} \end{cases} \quad (5.3)$$

R is the duct curvature, and θ is the arc angle.

The increasing duct radius, instead, is described by this equation:

$$\frac{r}{r_1} = 1 + 3 \left(\frac{r_2}{r_1} - 1 \right) \left(\frac{\theta}{\theta_{max}} \right)^2 - 2 \left(\frac{r_2}{r_1} - 1 \right) \left(\frac{\theta}{\theta_{max}} \right)^3 \quad (5.4)$$

All the cross sections perpendicular to the centerline are circular. The inlet has a radius r_1 , while the outlet has a radius r_2 as shown in Table 5.1. In Figure 5.2 it is possible to see that the geometry used is formed by four main parts. The first one is a straight duct before the inlet, with the same diameter. This part is four times longer than the diameter d_1 , and its main function is to uniform the condition of the flow before the stream arrives to the inlet.

Parameter	Value
θ_{max}	60°
R	415.16 mm
r_1	66.50 mm
r_2	82.00 mm
Offset	324.50 mm
Lenght	658.47 mm

Table 5.1: *S-Duct baseline geometry parameters*

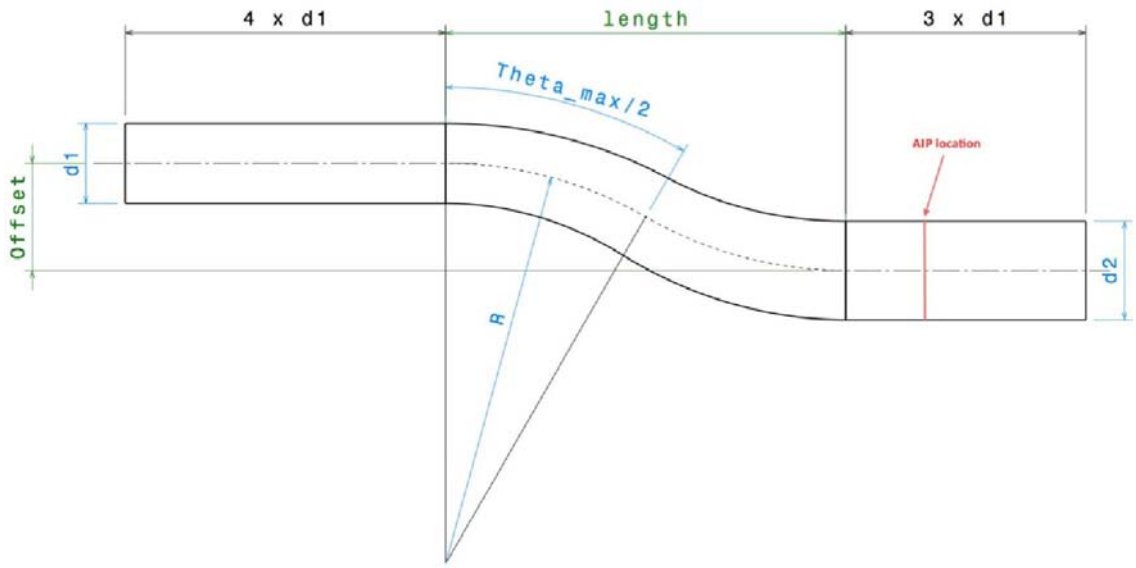


Figure 5.2: S-Duct baseline scheme

The second and the third parts are two curves, each one of them with a curvature R . An important characteristic of these curves is the growing cross sectional area. The ratio between A_1 and A_2 , respectively the area before the first curve and the area after the second one, is equal to 1.52. The fourth and final part is an other straight duct, with a diameter equal the outlet one, and its main function is to prevent the AIP conditions to influence the flow upstream.

Non-dimensional Parameter	Value
A_2/A_1	1.52
Offset/Lenght	0.48
R/r_1	6.24
Offset/ d_1	2.44

Table 5.2: *S-Duct baseline geometry non-dimensional parameters*

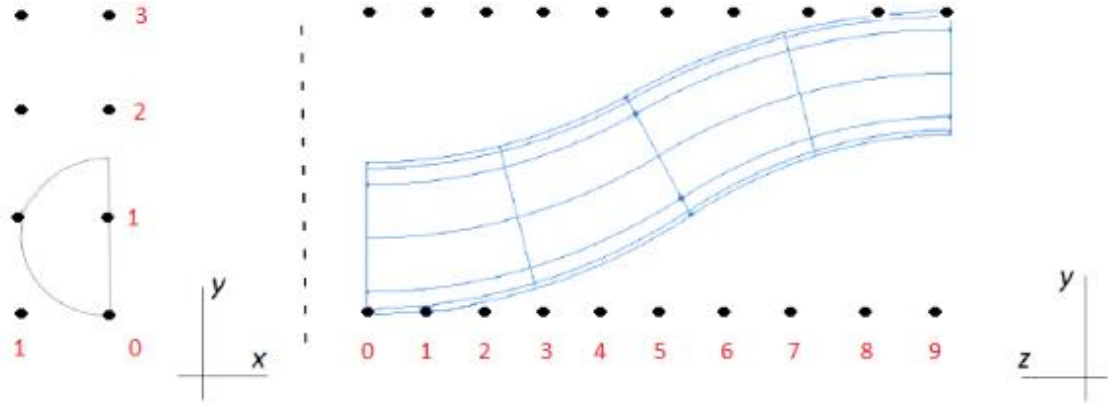
The other parameters indicated in Table 5.2 are the offset and the length, which are represented in Figure 5.2. The length is the distance along the x direction between the inlet and the outlet planes. The offset instead is the distance between the center of those two planes along the z direction. Finally the position of a point inside the duct is specified by the polar angle ϕ , as it is the angle measured from the y axis over a cross section (Figure 5.1).

5.2 Parametrization

Parametrization is a mathematical process consisting of expressing the state of a system, process or model as a function of some independent quantities called parameters. It is not a purpose of this thesis to analyze the parameterization used to describe the geometry, but it is a very important feature to understand. During the optimizations in fact, the deformation of the baseline geometry depends on a vector of 36 elements. The vector contains all the parameters needed to describe the geometry of the intake. In this way all the optimizer has to do to modify the geometry is just to change the vector elements, without worrying about 3D complex tasks or manipulations. It is interesting to understand how these 36 variables can modify the shape of the intake. Free-Form deformation method is the answer. As shown in Figure 5.3 the geometry is inscribed in a rectangular control volume, and a 80 control points are placed externally on its surfaces. These control points are connected to the geometry points and their bond is proportional to the distance between them.

2 control points are located along the x direction, 4 along the y direction, and 10 along the z direction. Without any considerations the variables which would describe the geometry should be 240, because of 80 control points that could move in 3 directions. Thanks to some simplifications made in the Rigobello's work [3] the number was reduced to 36:

- The points cannot move in the z direction because of manufacturing constraints.
- The points on the symmetry plane cannot move in the x direction, to prevent

Figure 5.3: *Control Points.*

asymmetric flow.

- 16 points at the inlet and outlet are fixed as a result of manufacturing constraints and the presence of the compressor.
- 16 points between the inlet and outlet cannot move in the y direction in order to guarantee tangential conditions, and to ensure the first derivative's continuity.

Then, she decided to limit even more the number of points in order to make the optimization faster. These points were those ones too far from the surfaces, so their change in position would not have brought great changes to the geometry:

- Sections 3 and 4: the upper volume may move, the other points are fixed.
- Sections 5 and 6: the middle volume may move.
- Sections 7 and 8: the lower volume may move.

This type of parameterization was an improvement if compared to previous works and it helps to fully explore the S-Duct profile configurations.

5.2.1 Constraints

Theoretically the position of a control point can be moved freely without any limitations, but in order to avoid design problems it was limited for two main reasons. The first one is because the optimization can produce geometries with outer bumps which would interfere with other parts of the aircraft, especially because intakes are mostly used when the engine is embedded into the fuselage. The second one instead, come from previous

analysis [18] in which was stated that the variation grater then $80mm$ always produce solver divergence or leads to a too high time-cost flow simulation, due to flow complexity. In this research, in order to give to the optimization algorithms the widest search space to work with, the maximum variation range was set to $80mm$. Thus during the optimization the value of the position of the control points, hence the value of the 36 elements which describes the geometry of the S-Duct, could be between 0 and 1. When an element value is 1 means that the control point is moved away $80mm$ away from the baseline position towards the positive direction of the axis, if it is 0 it means that is $80mm$ away along the negative direction.

5.3 Mesh Description

The mesh is one of the most important things of a CFD simulation. It is not possible to solve the continuous non-linear fluid-dynamics equations, so it is necessary to discretise the domain with nodes and elements. Only then it the equation can be solved algebraically. The quality of this discretization influences the time of the calculation and the reliability of the simulation. One bad cell can compromise the entire simulation, so it is very important to make it carefully. Basically there are two types of meshes, structured and unstructured, and both of them have pros and cons. Then we can have different levels of refinement, the mesh can be coarse, medium or fine. More refined it is, more accurate the results will be but at the expense of slower simulations. So, if the designer need a quick preliminar analysis of the flow behavior, an unstructured coarse mesh should be chosen. However if the results needs to be very accurate, a structured and fine mesh is needed. First of all lets see the differences between structured and unstructured meshes:

- **Structured mesh:** This kind of mesh is shown on the left side of Figure 5.4 and feature a regular connectivity between nodes and the shape of the cells is quadrilateral or hexahedral. For 2D problems in the former case, for 3D problems in the latter. Some advantages of this mesh are the low memory required to store it, and the low propagation of numerical errors due to the fact that the cell are aligned to the flow direction. However structured meshes are not easy to obtain and usually the time needed to create it is way longer then the unstructured one. In some cases, if the geometry is strongly distorted is impossible to mesh it in a structured way.
- **Unstructured mesh:** This kind of mesh is shown on the right side of Figure 5.4 and feature an irregular connectivity and additional information are needed to store the data, so it gets heavier in terms of files dimensions. Compared to structured

meshes, these ones are way faster to build, but the main drawback is that the results usually are less precise.

- **Hybrid mesh:** This kind of mesh is a mix of the previous ones. For example in an 2D airfoil problem we can find a structured mesh that covers the boundary layer and the unstructured mesh for the rest of the flow field.

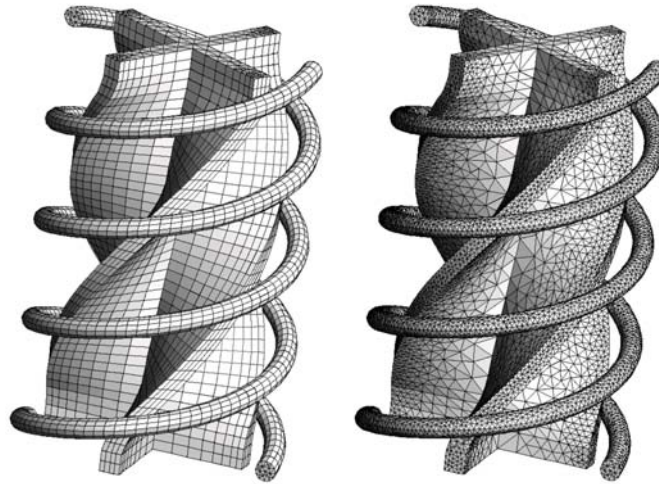


Figure 5.4: *Difference between structured (on the left) and unstructured meshes (on the right)*

The reader has to know that the quality of the mesh influence the convergence, while its refinement influence the reliability of the results.

5.3.1 Mesh Generation

Once the geometry is created it is imported and meshed automatically in ANSYS ICEM by a journal file written by Python. This journal file (*.rpl*) has all the instructions to pick up the right files that comes next. So the the geometry is taken from a bunch of *.dat* files and ICEM gives back the mesh as a *.msh* file that will be read later from ANSYS FLUENT. This process has to be automatic for optimization problems.

Some considerations have been done to reduce the time needed from each simulation:

- Since the flow is axisymmetric only half of the duct is considered, and previous works proved that this hypothesis does not influence the results in a relevant manner.
- In this chapter has being said that the geometry has to straight ducts before and after the inlet and the outlet as shown in Figure 5.2. Since the flow on the first

part is almost the same no matter what the shape is of the intake downstream (with our constraints), the straight duct length was reduced to $2/3 \times d_1$ upstream and all the flow parameters coinciding with cut was saved in a *.dat* file. In this way the simulation time for each duct was reduced.



Figure 5.5: *Half geometry mesh*

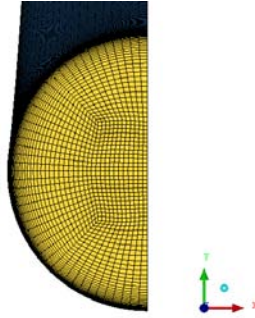


Figure 5.6: *H-O grid at AIP, the outlet of the intake*

In this research the mesh used is an H-O hybrid mesh of 1.1×10^6 nodes, and it is shown in Figure 5.5 and 5.6.

5.4 Ansys Fluent Settings

The flow was simulated with ANSYS FLUENT solver. It imports the mesh generated by ICEM and then it uses the settings studied by Guglielmi [18]:

- **CFD model:** The model used was RANS (Reynolds Average Navier Stokes). The objective of this turbulence model is to compute the Reynolds stresses, which can be done by three main categories of RANS-based turbulence models: Linear eddy viscosity models, nonlinear eddy viscosity models and Reynolds stress model (RMS). It solves simplified time-averaged Navier-Stokes equations, achieving considerable results in a reasonable amount of time.
- **Air model:** the flow is obviously considered compressible with an ideal density gas configuration. For the viscosity model instead, Sutherland model was applied.
- **Turbulence:** $k-w$ SST model was set up. For Delot and Penin's [31] considerations, this model is the one that better describe the reparate region in S-Ducts. Then, compressibility effects, viscous heating and curvature correction were selected with this model.
- **Solver:** Pressure-based solver was set. This type of solver suits the flow because the maximum Mach number is about 0.6, and it is the most used in these cases. The flow was solved coupling continuity and momentum equations together, while, for the gradients, Green-Gauss Node-based method was used.
- **Initialization:** Full Multi Grid (FMG) was chosen to make the convergence faster, initializing the simulations from a good solution.
- **Iterations:** After the iteration sensitivity stdy, it was decided to run the simulations for 7000 iterations at second order of accuracy, as a compromise btween convergence and time-cost. This set the residual at the end of the simulations about 10^{-6} for continuity, k, velocity, omega and energy.
- **Boundary conditions:** These come from the Delot's studies, and are shown in the Table below:

And from the values above, it is possible to calculate the following parameters:

All the simulations were carried out in the same way: the first 200 or 300 iterations run with the first order of solution accuracy for all the flow parameters. Then until the 7000th all the parameters were set to the second order.

Parameter	Value
Inlet total pressure	88.744 Pa
Inlet total temperature	286.2 K
Inlet static pressure	69.575 Pa
Outlet static pressure	78.982 Pa

Table 5.3: *Boundary conditions parameters for the simulations*

Non-dimensional Parameter	Value
Inlet Mach number	0.6
Massflow	2.43 Kg/s
Outlet Mach number	0.37

Table 5.4: *Flow parameters*

All the simulations, were run using a cluster of parallel processor which is available at Cranfield University. The optimization with GeDEA run on the cluster Astral, while Tabu search run on Grid.

5.5 Objective Functions

The goal of this research is to find the best designs for the S-Ducts, but the aerodynamic performances of it depends on several flow phenomena. Therefore in order to determine which are best designs, two of only two of them have been chosen to determine whether or not a duct is better then another. Our main concerns was to obtain a flow as uniform as possible at the outlet, because after it the compressor blades are designed for an undisturbed flow. For these reasons the two chosen parameter, which will be called *Objective Functions*, are:

- **Swirl**
- **Pressure losses**

Even if the physics of these two Objective functions have already been described, in this section our focus will be on their mathematical definitions and representation in these two optimizations.

The reader has to keep in mind that based on the purpose of this research to compare two different algorithms, the objective functions definition is exactly the same for both of them.

5.5.1 Swirl

The swirl, as already explained in chapter 2, is basically the twist of the flow due to the bubble of separation caused by the bend of the intake. Now, the behavior of the flow through the duct is very complex, so analyze the swirl for the entire length is time consuming and we don't want need that. That is why our objective functions are evaluated only at the AIP, which is the plane corresponding to the duct outlet or the compressor inlet. The swirl is calculated as an area-averaged parameter as it shown in Figure 5.7.

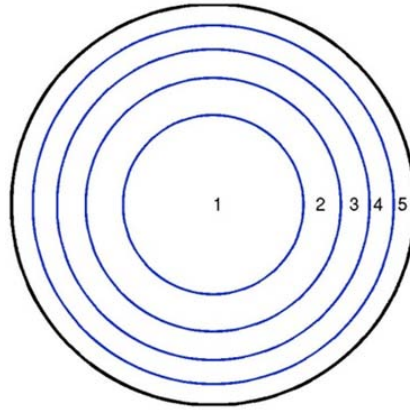


Figure 5.7: *Area-Averaged swirl parameter*

The outlet cross-section is divided in five circles with the same area. Swirl is calculated over this sections and then the five swirl values are averaged, obtaining the final AIP swirl. Swirl angle is defined as follow:

$$\alpha = \arctan \left(\frac{U_{\theta}}{U_z} \right)$$

While the radial velocity is defined as:

$$V_{\theta} = u \sin \theta - v \cos \theta$$

Where u, v and w are the flow velocity components respectively along x, y and z axes. Thus our first objective function is the area averaged swirl angle:

$$f_1 = \alpha$$

,

5.5.2 Pressure Losses

Even in this case, like it was for swirl, this is also calculated through an area-averaged calculation. In this particular case we want to obtain as much pressure recovery as possible, because higher is the pressure increase at the intake, lower will be the work of the compressor. Therefore the second objective function will refer to the pressure losses, in order to keep the optimization as a minimization problem. That is why this objective function is defined as:

$$f_2 = 1 - PR$$

Where PR represents the non-dimensional area-averaged Pressure Recovery:

$$PR = \frac{P_{t,outlet}}{P_{t,inlet}}$$

Hence, the goal of the optimizations is to bring this value as low as possible. In the ideal case without any pressure losses we will have the same total pressure both at the inlet and at the outlet. In reality, we just aim to have the total pressure at the outlet as close as possible to the one at the inlet.

5.6 Optimizations loop

In this section it will be presented the explanation of the loop thanks to which the optimization is made possible. The spine of this work is based on *Python codes* which generate and read all the files needed and produced from the different software. A Python code which needs only the 36 parameters that define the position of the control points is able to generate all the *.dat* files which describe the geometry. On top of that it creates even two very important *replay files*. Thanks to one of them ANSYS ICEM can create the mesh, while with the other, once the *.msh* file of the mesh is created, ANSYS FLUENT is able to simulate the flow through the intake without any help from the user. It has to be pointed out, that the boundary conditions are taken from Delot's experiment and from the *p0profile.prof* which contains the data thanks to which we were able to shorten the first straight part of the duct before the intake. Depending on which cluster we are using, each simulation lasts from 2 to 6 hours and even more in some cases. Then comes the turn of Python again, which analyzes all the FLUENT files produced, and evaluates the objective functions which are stored even on simple *.csv* files (comma separated values). Finally these results are communicated to the optimizer, which is GDEA-II or MOTS, and then a new set of parameters will be available to repeat the loop over and over.

5.6.1 Nimrod

Using two different algorithms required two different approaches to optimize their loop, because they had to be run on different computers and because of their different structure. That is why there is the need to introduce Nimrod.

Nimrod has been used for MOTS and it is a specialized parametric modeling system developed by Monash University. It provides a simple declarative language which allow the user to create an optimization which can be executed across distributed computers. Nimrod manages the whole experiment and provide the tools to automate the process. It make it run, it produce the files to monitor its execution, and it gives back the results for every step it does.

In particular *Nimrod/O* is an open source tool for "distributed optimization" [32]. It needs a schedule file, that is a declarative plan in which is defined:

- The domain and the type of the parameters.
- Any constraints imposed on the solution.
- The tasks to perform the experiment.
- Which and how many are the objective functions.
- Which optimization method has to be used, and in this case is MOTS.

All the files needed are placed in a working directory of the cluster in which there are the schedule file, and all the files required to execute the optimization loop. Once it is started it create a temporary folder in which one step of the optimization is accomplished, which mean one folder per each duct. When the objective functions of one duct are evaluated, the folder is deleted and the report files are updated. This report files are placed in the working directory, and they keep track of the evaluation already done, the value of the parameters used and so on. Using MOTS means that there will be even a folder in which will be updated the long, medium, and short memory.

APPENDICE

Chapter 6

Analysis and Discussion of the Results

This chapter is dedicated to analyze the results obtained throughout this project. The main goal for both of the optimization algorithms used was to reduce a particular flow distortion, the swirl, and to reduce the pressure losses within the S-Duct. Therefore the aim of this research is to compare these two optimization algorithms to determine which one perform better, or at least to analyze their advantages and disadvantages. Before going on, it is important to remind the reader that this project has been possible thanks to previous works of students like me that worked on their thesis at Cranfield University. The FLUENT simulation were carried out with a $k - \omega$ *SST* turbulence model because of the studies of Penin [31] regarding mesh and turbulence model sensitivity. The geometry total length was reduced in order to spend less time for each simulation thanks to the results of Guglielmi [18] and finally the parameterization used was the one developed and tested by Rigobello [3]. Therefore in order to save time and dedicate more attention to the optimization codes, it was avoided a further researches concerning these parameters. In this chapter there will be an analysis of the results obtained by the algorithms separately, and then they will be compared

6.1 Baseline Analysis

The first step of these optimizations is the validation of the baseline used. Since it is the duct studied by Delot, it was possible to compare his results with the one obtained with solver settings as described in chapter 5.4. Figure 6.1 shows the AIP pressure recovery distribution in three different cases.

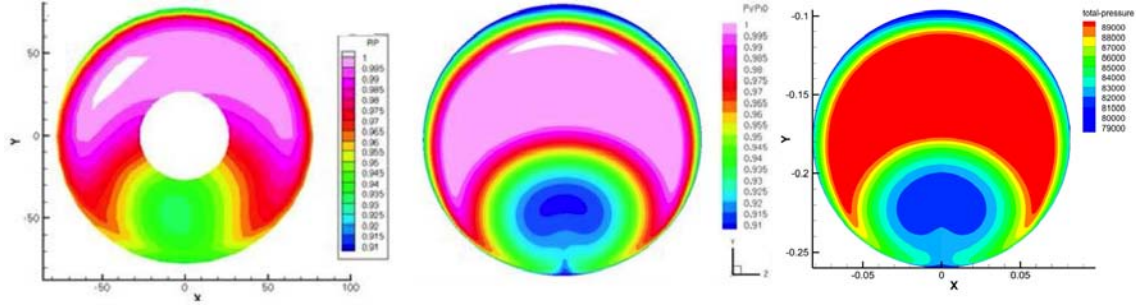


Figure 6.1: *Baseline PR Comparison:*
Left: Delot Experiment, Center: Delot Simulation, Right: Our simulation

On the left it is represented the empirical experiment carried out by Delot, on the center there is the results of his CFD simulation, and on the right we have our results. As the reader can see, our results are coherent with the experiment data. In fact Delot measured a value of the area-averaged pressure recovery equal to $PR = 0.9711$, while our simulations produced a value of $PR = 0.9685$, meaning that the percentage error is 0.27%. Therefore we can assume that the prediction of the pressure recovery is reliable and the error is negligible.

Then comes the turn of the validation of the flow distortion. This parameter is much more complicated to measure experimentally, so, as it was done in previous works from Delot to Rigobello, in order to validate this parameter it was decided to use the flow separate region location. Delot and Penin studies showed that the turbulence model $k-\omega SST$ matches more accurately then others the flow separation region. Since pressure recovery and flow separation cause the swirl, once we are sure that they are reliably evaluated, we can assume our model validated and verified. The values obtained from the baseline simulation which will be used as a reference points for our further discussions are presented in the table 6.1.

	Area-Averaged Pressure Recovery	α	1-PR
Baseline	0.9685	3.41424	0.03151

Table 6.1: *Baseline geometry objective functions*

Furthermore, Figure 6.2 shows the swirl angle α values at the AIP. As expected two vortices are clearly visible, as discussed in the Literature Review chapter.

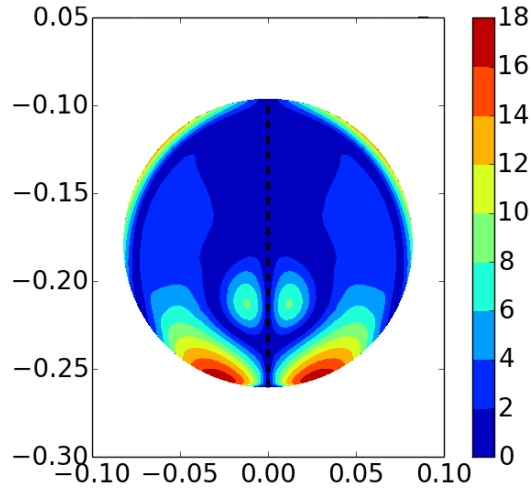


Figure 6.2: *Swirl representation at baseline AIP*

To have a better representation of this phenomenon, looking to the velocity vector could be even more explanatory. In fact Figure 6.3 shows even more clearly the presence of these two counter rotating vortices, but it shows also the fact that at the symmetry plane there is not rotation, endorsing the symmetrical behavior of the flow.

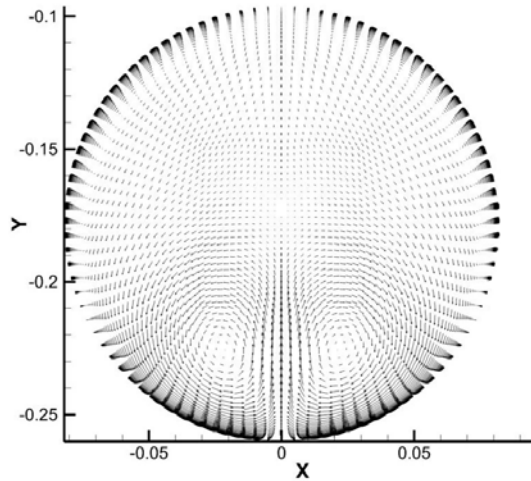


Figure 6.3: *Velocity vectors at baseline AIP*

Figure 6.4 present the symmetry plane of the S-Duct, and it very useful because it shows quite clearly the flow separation region after the bend. On top of that it is even represented its wake, which cover about half of the duct symmetry plane.

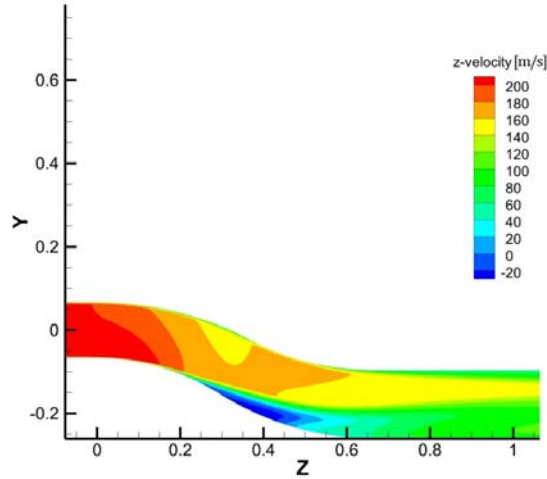


Figure 6.4: *z-velocity in baseline symmetry plane*

Figure 6.5 instead, shows the main function of the intake achieved by increasing cross-section area, which is gaining static pressure.

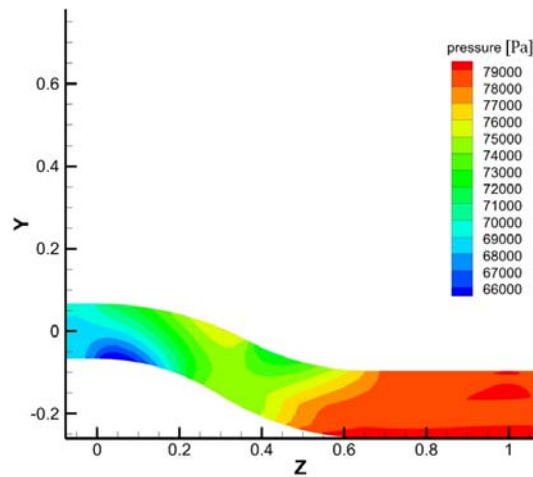


Figure 6.5: *Pressure fluctuations in baseline symmetry plane*

6.2 Optimization Results

In this section will be presented and discussed the results obtained with the genetic algorithm GDEA-II and MOTS. It is important to remind that the design vector is composed by 36 components, like in the Rigobello's work. This mean that the duct has 36 degrees of freedom, hence the search space will have 36 dimensions. Moreover, these parameters have a range that goes from 0 to 1, and the maximum variation which they can produce on the geometry is 80 *mm*.

6.2.1 GDEA-II Optimization Results

In particular, speaking of GDEA-II, the size of its first generation, also called starting population, is equal to 108 individuals, while the generations that followed has only 20 of them. Due to the long time needed for each duct simulation, every generation took about three days to complete, so it was not possible to evaluate more then 12 generations, for a final amount of 348 individuals.

Starting population

As said in GDEA-II theory, a first random generation, named *Generation 0* or *starting population*, is required to probe the search space.

The optimizer produced a matrix with 108 rows and 36 columns, it was given to Python which created the geometries and then through ICEM and FLUENT they were evaluated and post processed. Figure 6.6 shows the evaluation of this first generation and compare it with the baseline represented with the orange square. The Swirl angle is in the ordinate axis, and $1 - PR$ in the abscissa axis, whereas the points connected with the red lines indicate the Pareto front. In Figure 6.6 it is clear that there are already some improvement, both on swirl (f_1) and pressure recovery (f_2). Respectively an improvement of 19% for the former and a 19,5% for the latter objective function, although these values don't refer to the same individual, but they are values taken from the edges of the Pareto front. In fact, the duct with 19% improvement of the swirl angle, performed worse then the baseline about pressure leakages. This highlights the fact that in a multi- objective optimization, it is difficult to improve both of the objective functions at the same time. The individual with a good improvement in pressure recovery instead, performed a little bit better even for the swirl angle, but the improvement was not significant. We have been pretty lucky with this generation, because as it will be seen later, the Pareto front didn't move forward for a while.

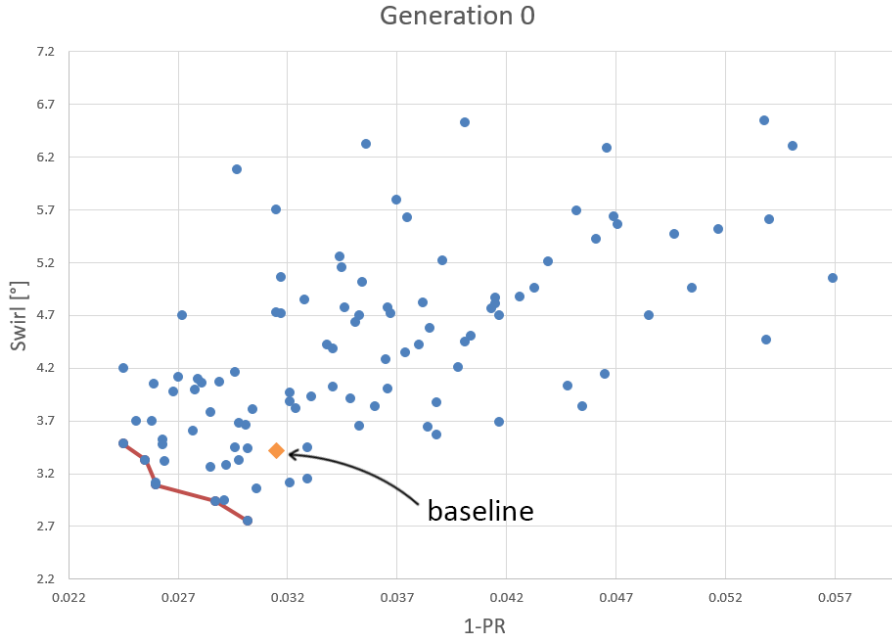


Figure 6.6: *Objective functions values of the Generation 0, the orange square is the baseline evaluation*

Optimized Generations

After all 108 individuals of the starting population have been evaluated, two columns can be added to the matrix built with the parameter of each duct. So we started with a matrix with 108 rows and 36 columns, and we ended up with a matrix with still 108 rows, but two more columns, i.e 38. This updated matrix has to be read from the genetic algorithm which gives back the new generation composed by 20 new individuals. Again they has to be simulated and the process was repeated until the generation 12. In total, the new duct generated after the random ones are 240.

Figure 6.7 shows the second generation, hence the *Generation 1* which performed very badly. In every graphs the orange square represent the baseline, and if it is taken as a graphic reference point, it is easy to understand that the Pareto front of this generation didn't improve at all. Not only the we didn't get better solutions then the previous generation, but not even one duct performed better then the baseline. However these data are still important for the optimization algorithm because they give information to the optimizer about where not to search, because solutions next to them are unlikely to be good. In Table 6.2 the reader can appreciate the performance of the individuals on the Pareto front of the first generation compared to the baseline solution. In this table the solutions are recalled with tags like 1_5, which means this is the 5th individual of the first

generation.

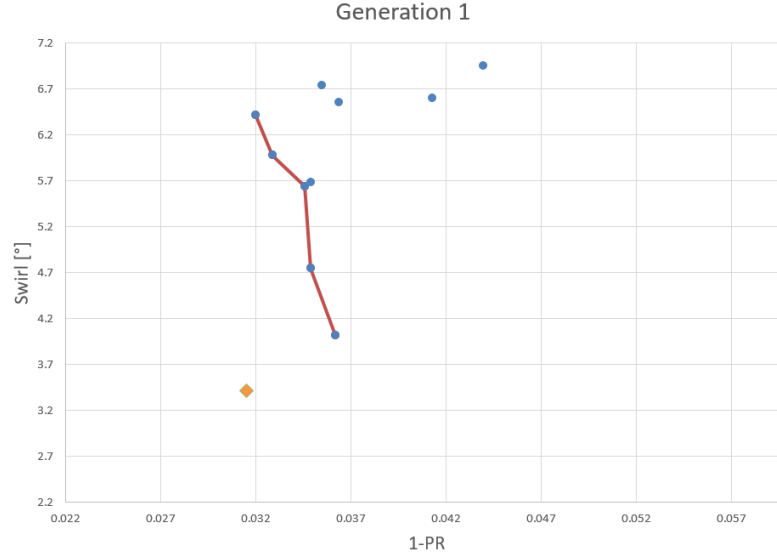


Figure 6.7: *Generation 1 Pareto front*

	1-PR	Improvement	α [°]	Improvement
Baseline	0.0315		3.41	
1_5	0.0349	+10.8%	4.75	+39.1%
1_6	0.0346	+9.8%	5.64	+65.2%
1_11	0.0329	+4.4%	5.98	+75.1%
1_12	0.0320	+1.5%	6.42	+88.0%
1_17	0.0362	+14.9%	4.02	+17.7%

Table 6.2: *Baseline compared to Generation 1 Pareto front*

The following generation, hence the third one, improved the previous Pareto front, but still the randomly generated generation performed better. Even in this case all the solutions were worse than the baseline. In Figure 6.8 and Figure 6.9 there are all the generations with their Pareto front highlighted by a red line, and the scale of the graphic is always the same in order to allow the reader to see the advancement of the Pareto front.

Generation 3 didn't improve the results obtained from the previous one, whereas *Generation 4* found a better solution, and finally the overall Pareto front moved for the first time since the starting population. Afterwards, *Generations 6-7-8* gave good solutions, adding new points to the Pareto front, while *Generations 5-9-10-11* didn't performed as we wanted. Only the last generation produced a new intake to be added on the Pareto front.

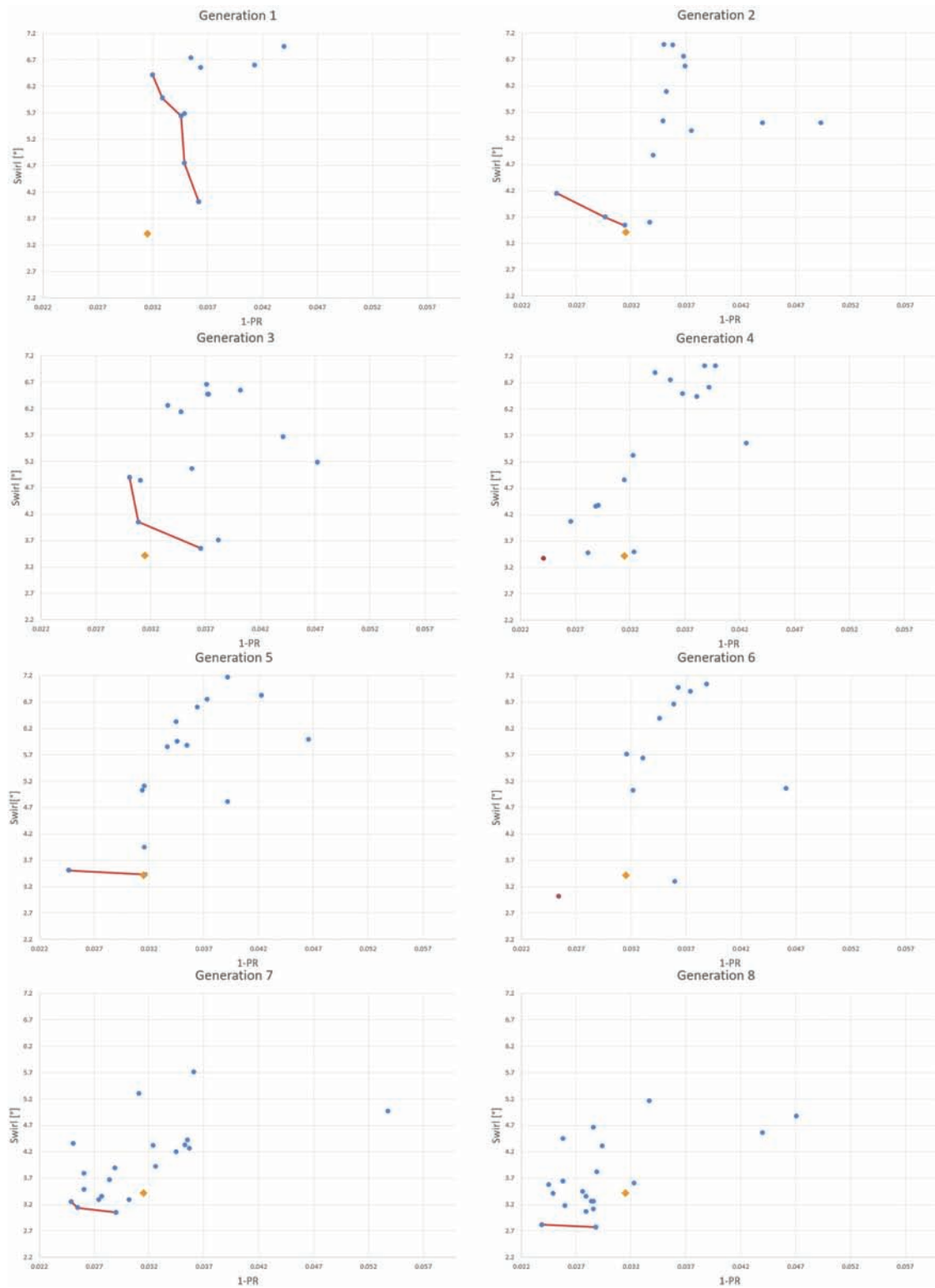
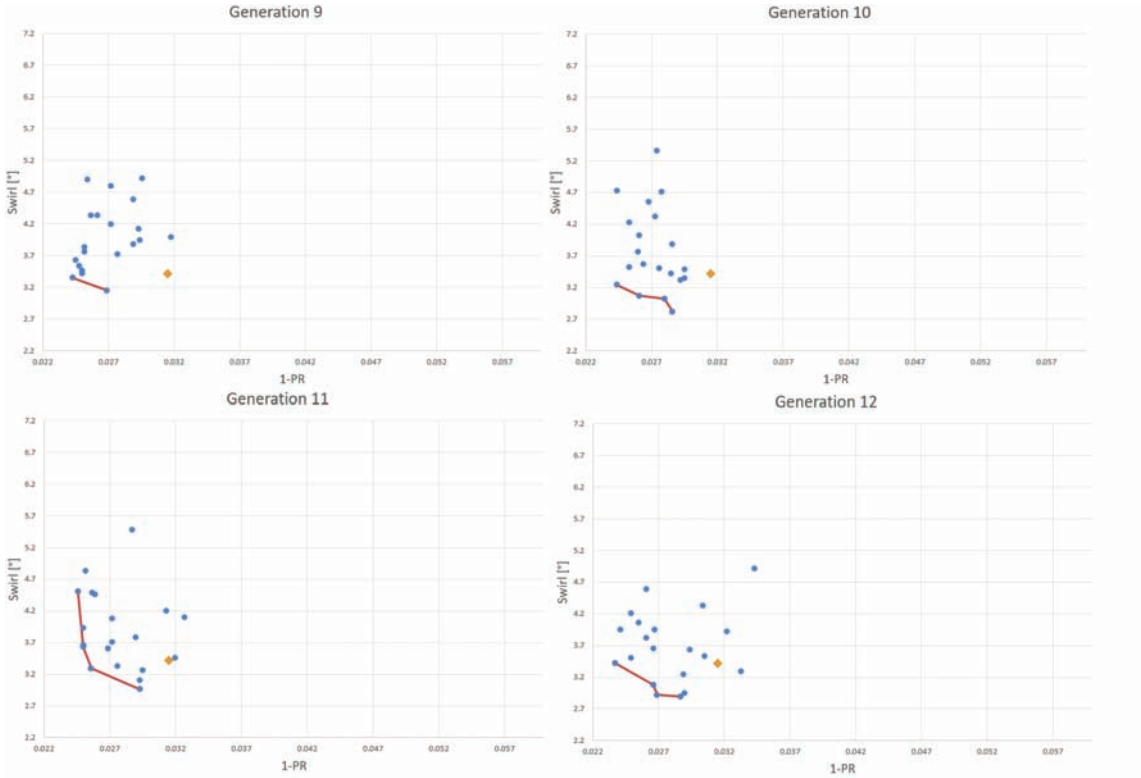


Figure 6.8: *Generation 1 Pareto front*

Figure 6.9: *Generation 1 Pareto front*

Even though three consecutive generations, from the ninth to the twelfth, hence 60 individuals, didn't add any new solution to the Pareto front, from the graphs can be seen how these generations kept going closer and closer to the Pareto front with all their individuals.

At this point 348 S-duts has been simulated and evaluated, and the final results can be seen in Figure 6.10 and the improvements achieved are shown in Table 6.3.

	1-PR	Improvement	α [°]	Improvement
Baseline	0.0315		3.41	
0_51	0.0302	−4.2%	2.75	−19.5%
8_6	0.0239	−24.2%	2.82	−17.4%
8_10	0.0288	−8.6%	2.77	−18.9%
12_0	0.0237	−24.8%	3.42	+0.2%

Table 6.3: *Baseline compared to the final Pareto front of GDEA-II*

Solution 1.51 is the better in term of swirl angle, but the improvement on pressure recovery were not so big. Solution 12.0 instead is very good on pressure recovery while the swirl is even worse then the baseline one. Between them the two trade-off solutions are

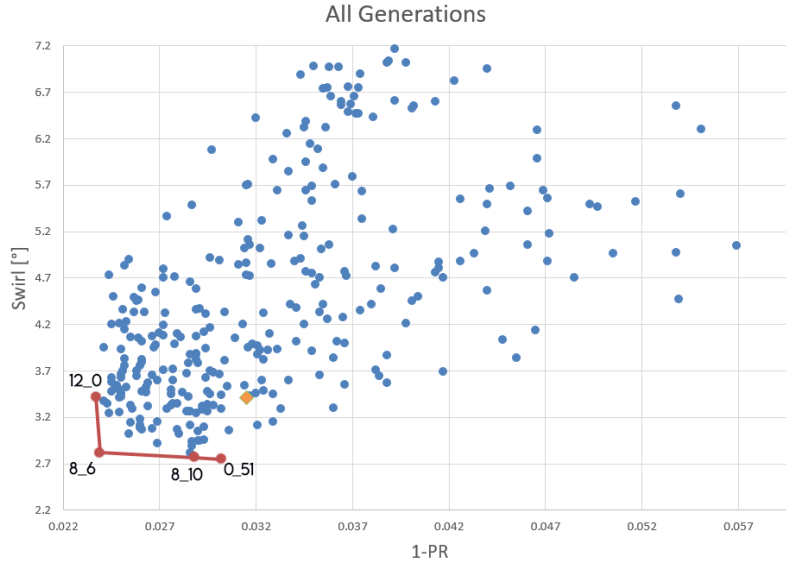


Figure 6.10: *Final Pareto front after Generation 12*

the one from generation 8. In particular the S-duct 8.6 performed very well reducing the pressure leakages about the 24.2% and the swirl angle about 17.4%, because the others couldn't be so effective contemporary on both of the objective functions.

6.2.2 Tabu Search Optimization Results

Unlike Genetic algorithms, Tabu Search doesn't start from a generation of individuals, but from a single solution. In this case the starting solution can be created randomly, or it can be defined from the user. Because the purpose of this thesis is to compare two algorithms that work differently, it was decided to start from a randomly generated individual, since GDEA-II started with a randomly generated generation.

Because of the complexity of the automation of this algorithm, and because of the slower cluster available (Grid) compared to the one in which GDEA-II worked (Astral), it was possible to evaluate only 53 individuals.

Advancement of MOTS

Tabu Search leaves to the user the possibility to tweak some of its settings, and one of the most important is the variation imposed on the parameters from one solution to another. In this work it was set up equal to 0.1, with the parameters ranging from 0 to 1. Later it will be clear why this variation parameter is so important.

The optimization with MOTS could continue to run indefinitely, or until a stopping

criterion is reached. With the Cranfield’s clusters though, it is possible to run the job for a finite amount of time. Grid, the cluster used, allowed only 170 hours of continuous work, and for this reason the Figure 6.11 highlights the three run carried out. It is important to understand that those are not generations, but the image help to understand the path chosen from the algorithm in order to find a better Pareto front.

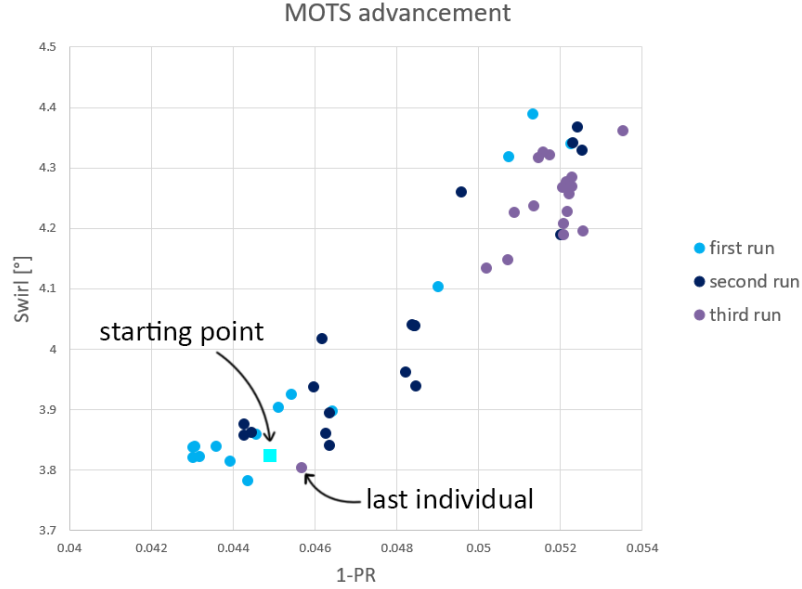


Figure 6.11: All the individuals evaluated with MOTS: “**first run**” are the individuals from 0 to 15, “**second run**” are the individuals from 16 to 33, and “**third run**” are the last 17 individuals, from 16 to 51.

Even if it seems that the optimization went in the wrong direction, with the first solutions better then the last ones, it even shows a good behavior. The last individual, highlighted in the picture, in fact, went back on the right track showing that, probably, with more time, the optimization could have improved the Pareto front soon.

	1-PR	Improvement	α [°]	Improvement
Baseline	0.0315		3.41	
solution 4	0.043941	+39.5%	3.814	+11.7%
solution 6	0.043036	+36.6%	3.820	+11.9%
solution 7	0.043031	+36.6%	3.838	+12.4%
solution 11	0.044369	+40.8%	3.781	+10.7%

Table 6.4: Baseline compared to MOTS Pareto front solutions

Figure 6.12 though, shows that all the solutions found with MOTS where not even close to the baseline. Table 6.4 describes even better how much worse they are compared

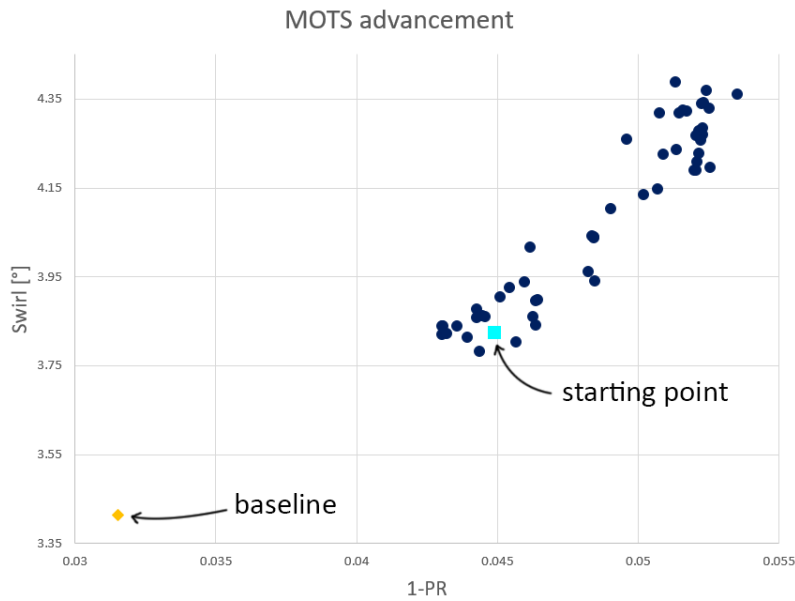


Figure 6.12: All the individuals evaluated with MOTS compared to the baseline geometry

to the baseline. All the individuals present pressure leakages from a 36.6% increase to a maximum of 40.8%, while the swirl angle increasing goes from a 10.7% to 12.4%. As it was already mentioned, the first solutions performed better than the last one, and in Table 6.4 it is confirmed by the fact that the Pareto solutions are within the 11th evaluation.

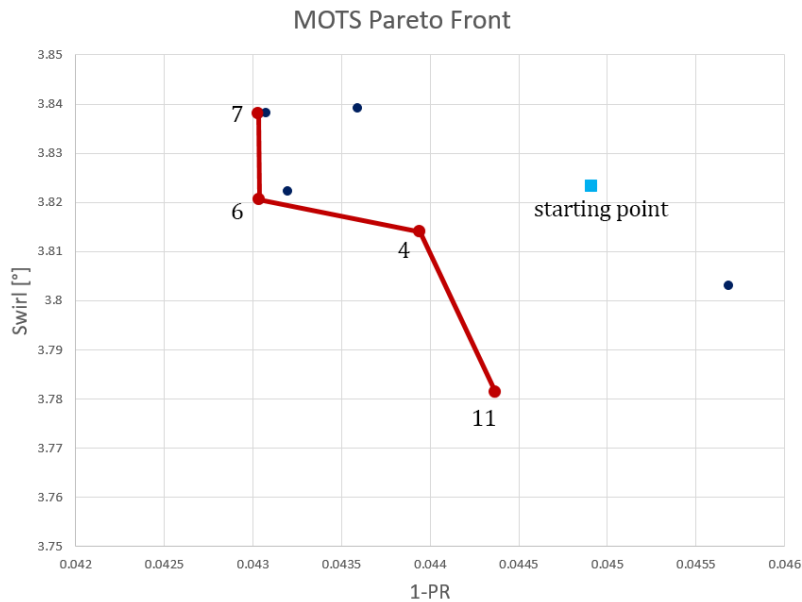


Figure 6.13: Close-up on the Pareto front of found from Tabu Search

Chapter 7

Algorithms Comparison

In this chapter there will be a comparison between the results obtained from the genetic algorithm GDEA-II, provided from the University of Padova, and the ones from Tabu Search, provided by the University of Cranfield. Both of the optimizations were carried out in Cranfield with the help of two clusters, Astral [4] and Grid [5], which were necessary due to the huge amount of computational power required for these tasks.

7.1 Pareto Front Comparison

As we discussed in the previous chapters, in a multi-objective optimization there are conflicting targets that don't allow to define a unique solution that is better than the other in every aspect. Therefore the best solutions found from our optimizations are collected on their Pareto front, and they are shown in Figure 7.1. In this image it is pretty clear that GDEA-II performed way better than MOTS, but this is not a fair comparison since the former ran for much longer, evaluating 348 individuals, while the latter had the chance to evaluate only 53 of them. For this reason we need to compare more deeply these two algorithms, and to do so, we can start from their way of starting the optimization.

Both of these optimizations started randomly, GDEA-II from a randomly generated generation, MOTS instead, from a randomly generated solution. Figure 7.2 shows the two different approaches. The genetic algorithm required a big amount of evaluations before the optimizer could start to make "intelligent" choices and aim towards the optimal solutions. As we can see on Figure 7.2, the starting generation of the genetic algorithm probes the search space in a very wide area, giving the optimizer a good idea of where the best solutions could be, and giving the opportunity to reach areas otherwise difficult to find. In fact, solution 0.51 from the starting population, kept its dominance even after the following 12 generations, and it is part of the final Pareto front as shown in Figure 7.3.

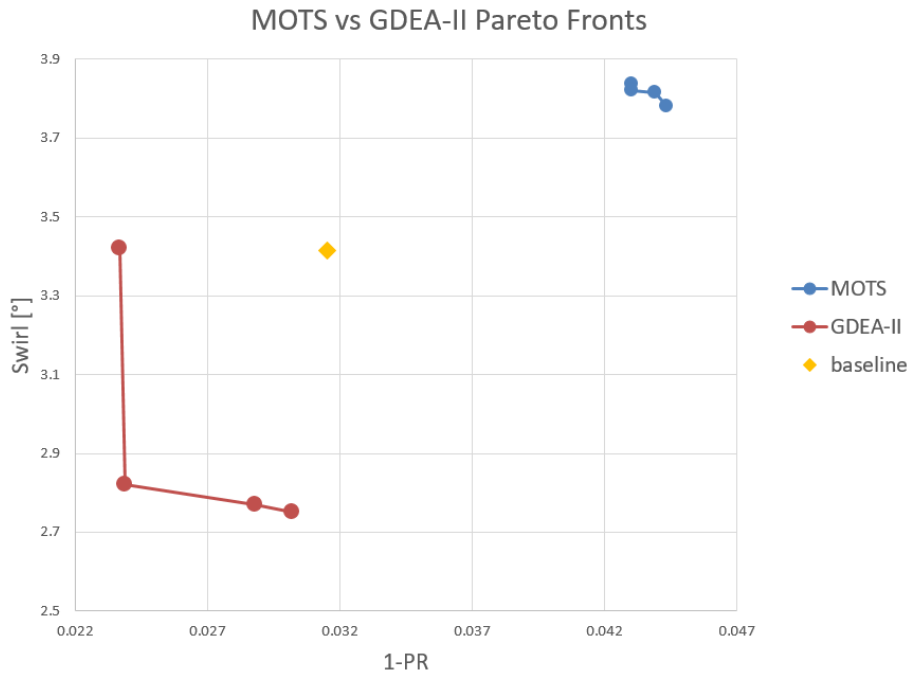


Figure 7.1: *Pareto fronts comparison between GDEA-II with 13 generations, and MOTS after 52 individuals*

MOTS on the other hand, should be more efficient in this first part of the optimization, because it leave only the first evaluation to chance. However in this case, the starting point didn't work in its favor, raising the difficulty of finding the right path towards better solutions. Therefore starting from the baseline could be a big improvement to achieve better results with less evaluations.

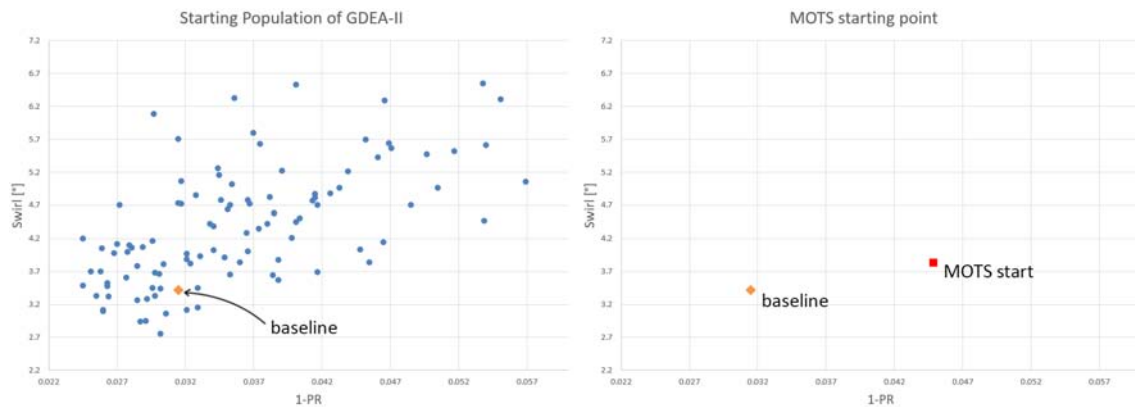


Figure 7.2: *Comparison between the starting points of GDEA and MOTS*

In order to do a fair comparison it would be helpful to take a look at Table 7.1. In

	Individual	1-PR	Improvement	α [°]	Improvement
MOTS	Starting point	0.044913		3.823	
	solution 4	0.043941	−2.2%	3.814	−0.2%
	solution 6	0.043036	−4.2%	3.820	−0.1%
	solution 7	0.043031	−4.2%	3.838	+0.4%
	solution 11	0.044369	−1.2%	3.781	−1.1%
GDEA-II	Baseline	0.0315		3.41	
	0_1	0.0255	−19.0%	3.33	−2.4%
	0_6	0.0245	−22.2%	3.48	+2.1%
	0_22	0.0260	−17.5%	3.09	−9.4%
	0_38	0.0291	−7.6%	2.95	−13.5%
	0_51	0.0302	−4.1%	2.75	−19.4%

Table 7.1: *Improvements of MOTS from its starting point, and improvements of GDEA-II from the baseline, withing the first 53 solutions for both of the algorithms*

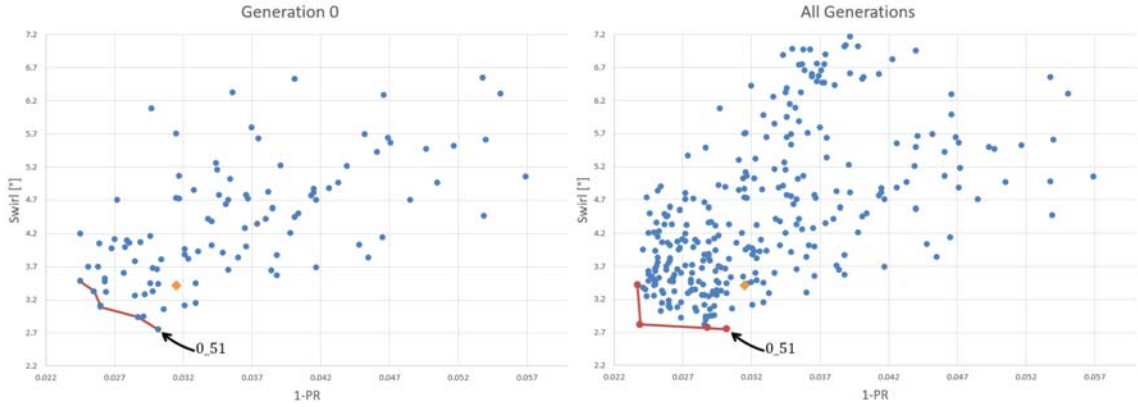


Figure 7.3: *Highlighting the fact that the individual 0_51 from the starting population is a non-dominated solution even after 12 more generations*

this table all the MOTS solution have been taken into account, but only the same amount of solutions have been considered from GDEA-II, i.e. 53 individuals. On top of that, the improvement have been calculated from two different reference points, the *starting point* for MOTS, and the *baseline* for GDEA-II. In this way MOTS can be compared more equally since its starting point, far away from the baseline, was a big disadvantage.

However, even if we tried to make the comparison as fair as possible, GDEA-II still performed better. Table 7.1 shows that with the same amount of evaluations, hence about the same time, GDEA-II reached very good improvements in total pressure leakages, reaching even a decrease of 22.2%, and quite good improvements on flow distortion, reducing the swirl more than 2.1% in three solutions out of four. MOTS instead, couldn't reach more than the 4.2% reduction on total pressure leakages, and 1.1% on swirl angle distortion.

With a similar approach we can compare the Pareto Fronts with only the first 53 individuals of GDEA-II, since the comparison seen at the beginning of the chapter was considering a very different amount of solutions between the two algorithms. Figure 7.4 shows the different searches carried out by the two algorithms. Even in this case GDEA-II performed way better than MOTS.

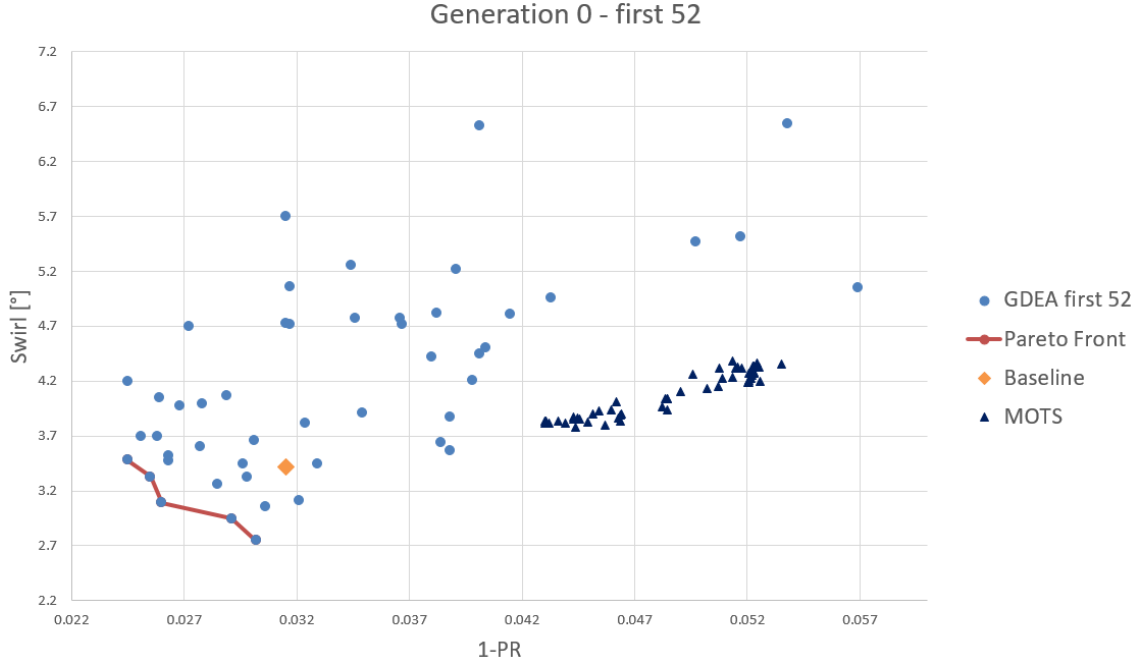


Figure 7.4: Comparison between the first 53 solution of both of the algorithms

Looking at the chart make easy to understand that GDEA-II searched on a wider area since the beginning, while the solutions of MOTS are confined only in a small area. This could be the sign that the variation setting previously named, makes a huge difference on the optimization, and an higher value in this case would be helpful, even though, it is not sure if it will be a right choice with the baseline as the starting point.

7.2 Optimum Geometries Comparison

In this section will be analyzed the Pareto front geometries. Therefore we will focus on the designs with the lowest total pressure leakage, lowest swirl angle distortion and the best trade-off solutions.

7.2.1 Lowest Total Pressure Leakages Solutions

As previous discussions showed, the best solution in terms of pressure recovery is solution 12_0 from GDEA-II and the 7th solution from MOTS. Let's see the comparison between these results in Figure 7.5.

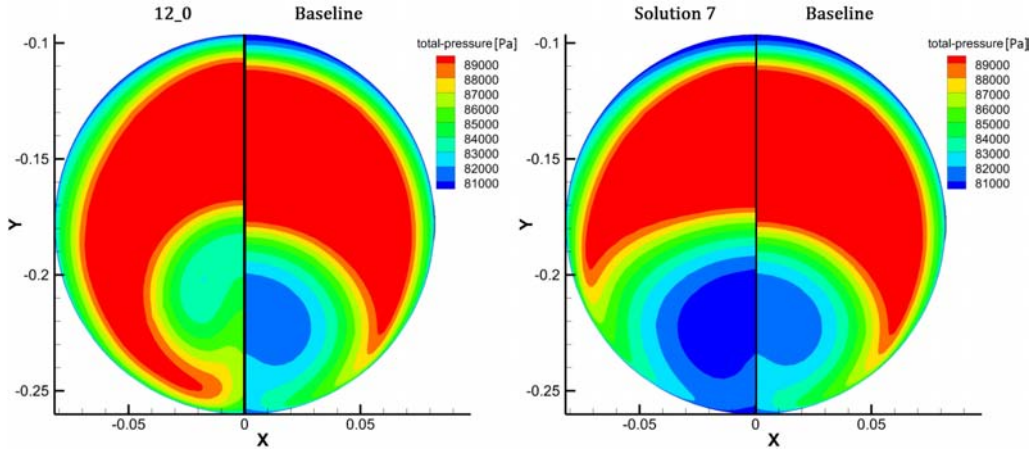


Figure 7.5: *Left: Solution 12_0 from GDEA-II, Right: 7th solution from MOTS; both compared against the baseline.*

Both of them are compared to the baseline solution in order to enhance the improvements made from the optimizations. As we can see a major improvement has been done with GDEA-II. The objective function 1-PR (f_2) in this case is reduced by 24%, from 0.0315 of the baseline, to its value equal to 0.0237. Contrariwise, MOTS didn't have enough time to improve the baseline configuration, and it couldn't find a better design. Its best solution is the 7th one and it increased f_2 by the 36.6%. Figure 7.5 shows clearly how the area of low total pressure is much bigger then the one of the baseline, while in the solution 12_0, is almost completely gone. As we did before, if we compare the results considering only the first 53 evaluations of GDEA-II, the outcomes are almost the same. Solution 0.6 in fact, reduced f_2 by the 22.2%.

With regard to Figure 7.6, we can analyze the velocity distribution on the symmetry plane of the optimum geometries from the two optimizations. On top of the figure there is the individual 12_0 obtained with GDEA-II, on the bottom instead, there is the 7th solution from MOTS. The most important difference highlighted from this comparison is the dimension of the separation region, which is very big on the MOTS solution, and it is the main reason of its bad performances on total pressure leakages. On top of these flow observations, in this graph we can appreciate the different shape obtained from the two algorithms.

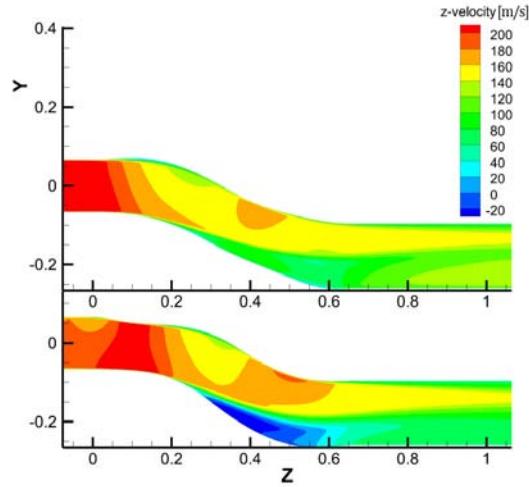


Figure 7.6: Separate region comparison: Top: 12_0, Bottom: 7

Apart from pressure recovery and separation region, the last parameter to analyze is the flow distortion, since a design that reduce f_2 tend to increase it. In Figure 7.7 is presented the swirl distortion improvements of the solution 12_0 (on the left) with respect to the baseline (on the right). With the aera-averaged method used to evaluate the swirl angle, the result for this duct is 3.42° , hence the 0.2% more then the baseline configuration. However the figure seems to tell a different story, in which the optimized geometry has a less flow distortion. The area with a swirl angle different from zero is a little bit larger, but also more uniform. Thus, this case shows that probably the area averaged evaluation is not the best way to consider this objective function, and some improvements can be done in this area.

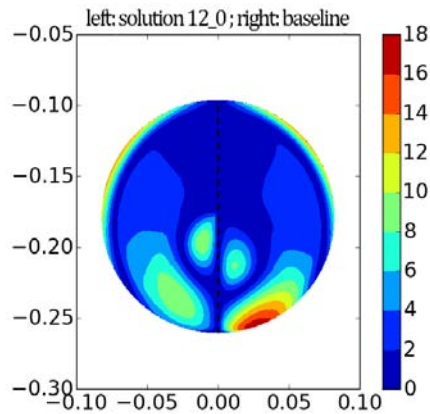


Figure 7.7: Swirl angle at AIP

On the other hand, the best solution of MOTS performed very badly, as the big separation region in Figure 7.6 suggested. The swirl angle calculated for this geometry is 3.838° and it is 12.4% more then the baseline one.

7.2.2 Lowest Swirl Angle Solutions

Now it is time to analyze the solutions with the lower swirl angle, and they are solution 0.51 from GDEA-II, and the 11th solution from MOTS. In order to analyze them with the same procedure followed before, we will begin comparing the pressure leakages. Figure 7.8 shows the comparison of f_2 (1-PR) between the optimized solution and the baseline.

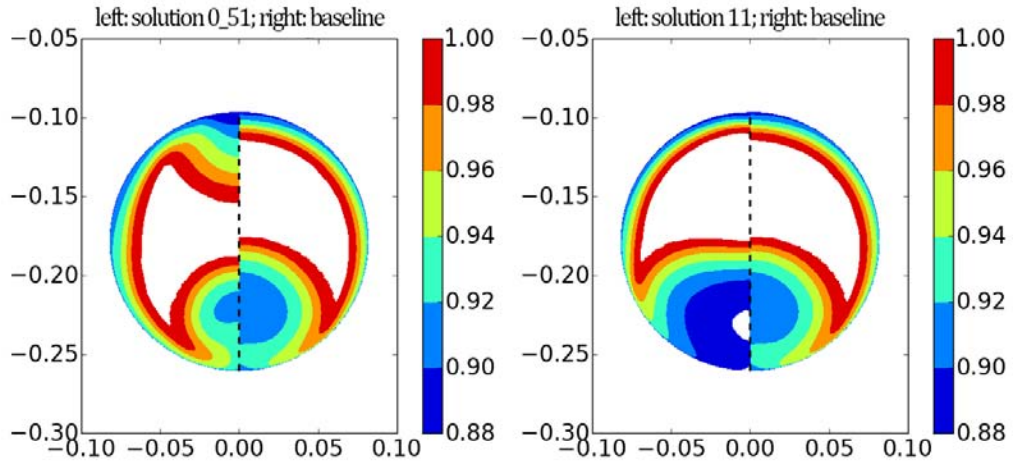


Figure 7.8: *Left: Solution 0.51 from GDEA-II, Right: 11th solution from MOTS; both compared against the baseline.*

As we can see the behavior of the two geometries is very different from each other. Firstly we take a look at the image on the left of the figure. The solution 0.51 improved f_2 by 4.1% from the baseline, with a value equal to 0.0302. The 11th solution from MOTS even in this case produced a geometry worse then the baseline, raising f_2 by 40.8%. The comparison from the two is straight forward, and the best solution in every aspect is 0.51. What is interest to observe is how this solution reduced the wake of the separation region on the bottom of the intake, but a new one appeared on top. To understand better this behavior it is helpful to take a look at the symmetry plane of this geometry.

Figure 7.9 shows the velocity along the z axis on the symmetry plane on both of the optimized geometries. The one on top is the solution 0.51 and the one on the bottom is the 11th solution from MOTS. As we can see, the image on top has a bump before the bend, and this is a characteristic found even on previous works that helped to reduce the swirl angle. Since this bump creates a small separation region on top, the pressure

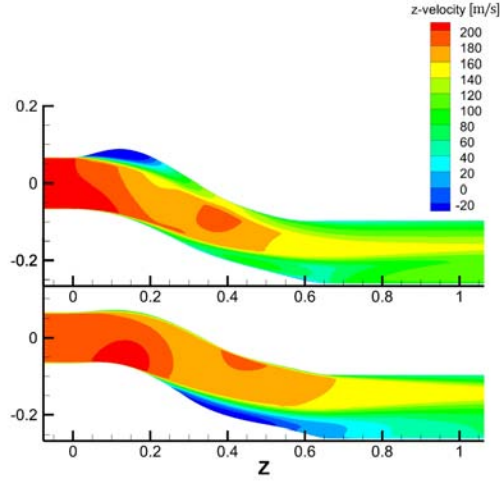


Figure 7.9: Separate region comparison: Top: 0-51, Bottom: 11

leakages cannot be improved a lot. However its wake inhibits the effect of the separation region on the bottom that usually is the main reason of the swirl perturbation. In fact Figure 7.10 shows how well the swirl is reduced. The swirl angle reduction achieved with this design was reduced by 19.4%, from 3.41° of the baseline, to 2.75° .

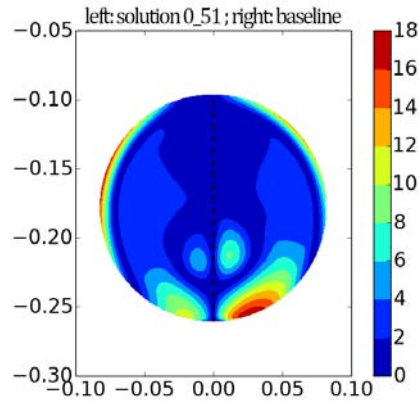


Figure 7.10: Swirl angle at AIP, solution 0.51

The solution created by MOTS cannot handle the comparison again, with an increase of the swirl angle from the baseline equal to 10.7%.

7.2.3 Best Trade Off Solutions

This section will analyze the best trade-off solutions found with GDEA-II and MOTS. In order not make the comparison too chaotic, even if the trade-off solutions on the Pareto

front for both of the optimizations are two, it was chosen to analyze only one of them for GDEA-II because there are clearly one better than the other. GDEA-II trade-off solutions are shown in Table 7.2, and the chosen solution for this comparison is 8_6. It has a little less reduction on the swirl angle if compared to solution 8_10, but the pressure leakages are way better.

	1-PR	Improvement	α [°]	Improvement
Baseline	0.0315		3.41	
8_6	0.0239	−24.2%	2.82	−17.4%
8_10	0.0288	−8.6%	2.77	−18.9%

Table 7.2: *trade-off solutions of GDEA-II*

Speaking about MOTS instead, we will take into account both of the trade-off solutions, but the graphs are so similar that the difference is not noticeable, and for this reason we will refer to the same image for both of them. As always we start by comparing the reduction of f_2 , and Figure 7.11 shows the differences between the two algorithms.

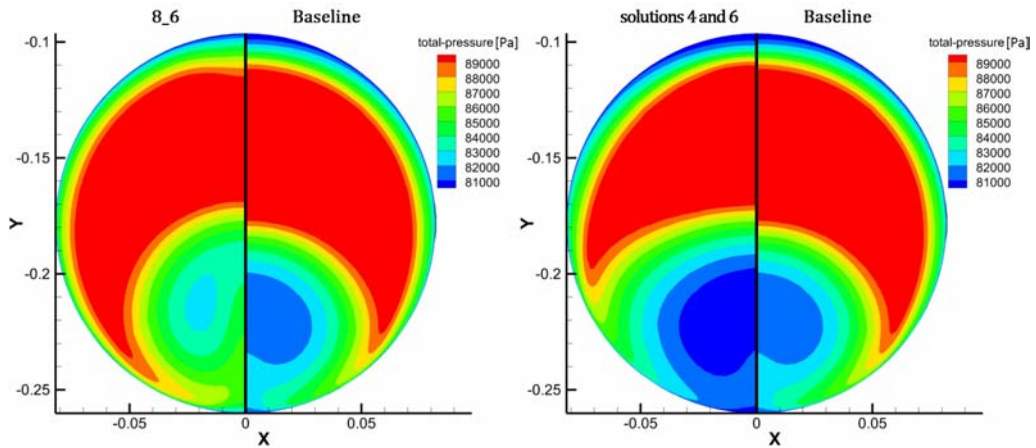


Figure 7.11: *Left: Solution 8_6 from GDEA-II, Right: 6th solution from MOTS; both compared against the baseline.*

As always, the longer run of GDEA-II produced better results, with a reduction of this objective function equal to 24.2%, i.e. 0.0239, from the baseline geometry which had f_2 equal to 0.0315. But still, even if we consider only the first 53 evaluation in order to have the same amount of individuals as MOTS, the trade-off solution 0_22 reduced the pressure leakages more than 17%. Contrariwise the trade-off solutions of MOTS couldn't reduce the objective functions from the baseline values. Solution 4 increased 1-PR by 39.5% and solution 6 increased it by 36.6%.

In fact, as we can see in Figure 7.12, solutions 4 and 6 have a not regular shape, and

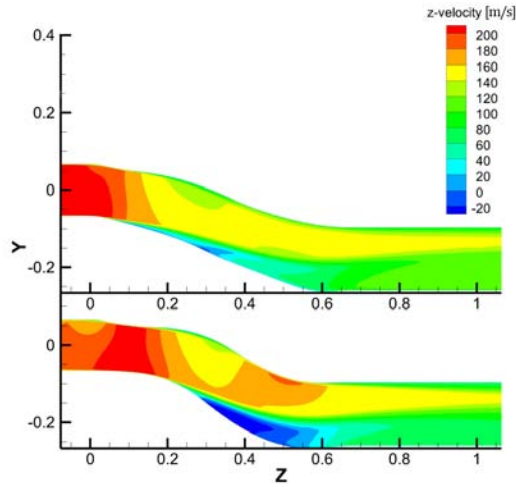


Figure 7.12: Separate region comparison: Top: 8_6, Bottom: 4 and 6

this affected both of the objective functions badly. It might be more interesting to speak about the fact that the shape of the ducts from the 4th to the 6th solution didn't change enough to make it visible on these charts. This is a clear sign that the variation parameter imposed to Tabu Search was too small, and it should be helpful to increase it from the current value (0.1), even because the algorithm itself reduce this variation during the optimization.

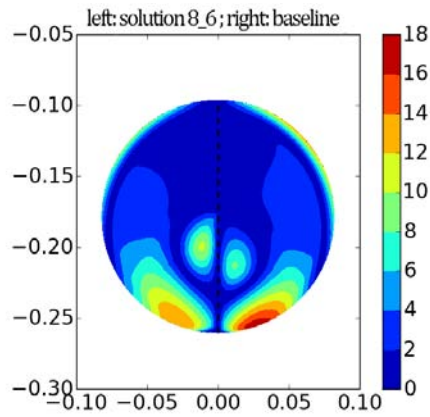


Figure 7.13: Swirl angle at AIP, solution 8_6

Finally, as we saw before, the same thing happen when we take a look at the flow distortion. The swirl angle reduction for the GDEA-II solution is about the 17.4%, while the MOTS solution increased it respectively by 11.7% and 11.9% for the solutions 4 and 6.

Chapter 8

Conclusions

The main topic of this thesis is the comparison of the optimization of an S-Duct shape, carried out using two algorithms, a genetic one called GDEA-II, and the Multi-Objective Tabu Search (MOTS) algorithm. The aim of these optimizations was the same, and consisted in the reduction of flow distortions and pressure losses at the compressor inlet. The main objectives were to:

- Adapt the framework used for previous optimizations with genetic algorithms in order to use GDEA-II.
- Automate completely the optimization loop with the Tabu Search algorithm. Something that was never done before.
- Compare the results produced by the two optimizations which had to be carried out with the same parameterization.

For this thesis, the parameterization developed by Rigobello [3] was chosen. It uses 36 variables to describe the geometry, and the Free Form Deformation (FFD) method to change its shape. The geometry is deformed in a similar way to that of former studies: inlet, outlet and also their nearby area were fixed as before.

Thanks to previous researches, some simplification were used to reduce the computation time required from each CFD simulation. Only half of the geometry of the ducts were simulated because of the symmetrical flow behavior, and furthermore, the inlet domain was reduced. A coarse H-O mesh of 1.1×10^6 nodes, the k-w SST turbulence model and a RANS methodology were applied on ANSYS Fluent to simulate the flowfield. Then the baseline geometry was compared and validated with Delot's experiments.

The automation of both of the optimization is done with Python codes arranged specifically for this research. The code is made by several subroutine and scripts that allows

to save time, and to make possible the communication between different software. Moreover, these subroutines sometimes are used to make very important steps without the help of any other software, like for the creation of the geometry, or for the evaluation of the objective functions.

The objective functions chosen were the total pressure leakages at the Aerodynamic Interface Plane (AIP), and the swirl angle at the AIP. These two intake characteristics are considered the ones that most influence the flow uniformity at the AIP.

Both of the optimizations were carried out with a variation range equal to 80mm. The optimization with the algorithm GDEA-II run until the 13th generation, which meant 348 ducts; with MOTS instead only 53 ducts were evaluated due to the limited amount of time available after its complex automation was complete. After all of these evaluations the algorithms were compared about the path they followed to reach the optimum solutions, and about their best solutions, hence the ones on the Pareto fronts.

GDEA-II resulted to be the best algorithm with settings used during this research, but MOTS potential is still very high because the results of this work showed which of its settings has to be tweaked to improve its performances. The GDEA-II and MOTS solutions on their Pareto fronts are showed in Table 8.1.

	Individual	1-PR	Improvement	α [°]	Improvement
	Baseline	0.0315		3.41	
MOTS	solution 4	0.043941	+39.5%	3.814	+11.7%
	solution 6	0.043036	+36.6%	3.820	+11.9%
	solution 7	0.043031	+36.6%	3.838	+12.4%
	solution 11	0.044369	+40.8%	3.781	+10.7%
GDEA-II	0_1	0.0255	-19.0%	3.33	-2.4%
	0_6	0.0245	-22.2%	3.48	+2.1%
	0_22	0.0260	-17.5%	3.09	-9.4%
	0_38	0.0291	-7.6%	2.95	-13.5%
	0_51	0.0302	-4.1%	2.75	-19.4%

Table 8.1: *Improvements of MOTS and GDEA-II from the baseline*

The project achieved all the goals initially set, and for the first time an optimization with Tabu Search was automated successfully.

8.1 Recommendations for Future Researches

From the outcomes obtained and the experience gained from these optimizations, some suggestions are proposed for further studies.

Firstly it is recommended to enhance the number of generations with GDEA-II, in order to push the Pareto front at its limit. In fact one of the non-dominated solutions was produced on the last generation.

The same suggestion can be said about MOTS. The time required to automate its very complex process left a limited amount of time for the optimization itself, and for this reason a bigger number of evaluations is needed to investigate the potential of this algorithm.

Then it is suggested to change the starting points of Tabu search and changing it from a random one to the baseline configuration. Starting from a random solution was a big disadvantage for this algorithm. Moreover it would be helpful to see if it can achieve the same results starting from different solutions, in order to see how much it depends from its starting point. With regards of MOTS the last suggestion is to tweak the variation parameter in order to speed up the convergence towards a better Pareto front. It would be even better to set a different variation setting per each design parameter.

Finally, a different objective function evaluation model should be implemented. The area-average evaluation for both of the objective functions is rough, and it works well in the first part of the optimization, but if we want more precise results, a more refined model is needed.

Bibliography

- [1] R. H. Liebeck R. 1, "Design of the Blended Wing Body Subsonic Transport", Huntington Beach, California : Journal of Aircraft, 2004, Vol. 41.
- [2] A. P. Plas, M. A. Sargeant, V. Madani, D. Crichton, E. M. Greitzer, T. P. Hynes, C. A. Hall, "Performance of a Boundary Layer Ingesting (BLI) Propulsion System", 45th AIAA Aerospace Sciences Meeting and Exhibit 8 - 11 January 2007, Reno, Nevada
- [3] A. Rigobello, "A Multi-objective shape optimization of an S-Duct intake through NSGA-II genetic algorithm", 5 December 2016, Padova
- [4] A., Gittings, "Astral-2 cluster user guide". Cranfield University : s.n., 2012.
- [5] Richard Stanworth, "GRID cluster user guide". IT Department, Cranfield University : September 2016
- [6] Ahmed F. El-Sayed and Mohamed S. Emeara, "Intake of Aero-engines: a case study", The internal conference of Engineering Sciences and Applications, Aswan, Egypt, 31 - 01 - 2016
- [7] "A methodology for assessing inlet swirl distortion". Technical report, SAE Aerospace, International Group, 2007.
- [8] Anderson, John D. (2004), "Introduction to Flight", Section 4.20 (5th edition)
- [9] "Clancy, L.J., Aerodynamics", Section 4.14
- [10] Wellborn S.R., Reichert B.A., Okiishi T.H. "An experimental investigation of the flow in a diffusing s-duct". Technical report, NASA Technical Memorandum 105809, AIAA, 3622(92), 1992.
- [11] Clinton Jefferson Farmer, "Inlet distortion, vorticity, and stall in an axial-flow compressor", 1972
- [12] Weske, J.R., "Pressure loss in ducts with compound elbows". Technical report, NACA Wartime Report, 1943.
- [13] Bansod P., Bradshaw P., "The flow in s-shaped duct". Aeronautical Quarterly, 13:131-140, 1972.
- [14] Harloff G. J. Smith C. F., Bruns J. E. and DeBonis J. R. "Three- dimensional compressible turbulent computations for a diffusing s-duct". Tech. rep., NASA Contractor

- Report, page 4392, April 1992.
- [15] Delot, A.L., Scharnhorst, R.K., "A comparison of several CFD codes with experimental data in a diffusing s-duct". 49th AIAA Joint Propulsion Conference, 2013.
 - [16] Fiola C, Agarwal R., Simulation of Secondary and Separated Flow in diffusing S-ducts. *Journal of Propulsion and Power*, 31(1):180-191, 2015.
 - [17] A.Rigobello, "A Multi-objective shape optimization of an S-Duct intake through NSGA-II genetic algorithm". Cranfield University: MSc Thesis, 2016
 - [18] L., Guglielmi. "Aerodynamic S-duct intake shape multi-objective optimization for distributed propulsion". Cranfield University : Msc Thesis, 2015.
 - [19] E., Manca. "Unsteady Aerodynamic investigation of the flow within an optimized S-duct intake". Cranfield University : MSc Thesis, 2016.
 - [20] M., Barison. "Shape optimization of highly convoluted intakes using genetic algorithms and metamodels". Cranfield University : Msc Thesis, 2016.
 - [21] A. M. Laguna, A J. L. Gonzalez-Velarde, A Search Heuristic for Just-in-time Scheduling in Parallel Machines, *J. of Int. Manu.*, Vol. 2, 1991, pp. 253-260.
 - [22] D.M. Jaeggi , G.T. Parks, T. Kipouros, P.J. Clarkson, "The development of a multi-objective Tabu Search algorithm for continuous optimization problems", *European Journal of Operational Research* 185 (2008), University of Cambridge.
 - [23] R. Battiti, G. Tecchioli, The Reactive Tabu Search, *ORSA Journal on Computing*, Vol. 6, No. 2, 1994, pp. 126-140.
 - [24] Jerzy Balicki, "Tabu Programming for Multiobjective Optimization Problems", Naval University of Gdynia, ul. Smidowicza 69, Gdynia, Poland (2007)
 - [25] A.M. Connor, D.G. Tilley, A Tabu Search method for the optimisation of fluid power circuits, *IMEchE Journal of Systems and Control* 212 (1998) 373-381.
 - [26] A. Toffolo, E. Benini, "Genetic Diversity as an Objective in Multi-Objective Evolutionary Algorithms", Massachusetts Institute of Technology (2003), *Evolutionary Computation* Volume 11, Number 2.
 - [27] C.C. Da Ronco, E. Benini, "A Simplex Crossover based evolutionary algorithm including the genetic diversity as objective", Department of Industrial Engineering, University of Padova, (2013)
 - [28] S. Tsutsui, M. Yamamura, T. Higuchi, Multi-parent recombination with simplex crossover in real coded genetic algorithms, in: *Proceedings of the GECCO-99*, 1999, pp. 644-657
 - [29] J.M. Nelder, R. Mead, " A simplex method for function minimization", *Computer Journal* 7 (4) (1965) 308-313
 - [30] T. Back, Selective pressure in evolutionary algorithms: a characterization of selection

- mechanisms, in: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE PRESS, 1994, pp. 57-62.
- [31] Penin L., Computational analysis of swirl distortion in a diffusing s-duct, MSc thesis, Cranfield University, 2013.
- [32] T. C. Peachey, Timos Kipouris and Monash University, "Nimrod/O Users' Guide", 2001-2012

Appendix A

Nimrod Schedule File

It would be helpful for future works to see the schedule file needed from Nimrod in order to run the optimization with Tabu Search:

```
parameter x1 float range from 0.0 to 1.0
parameter x2 float range from 0.0 to 1.0
...
parameter x36 float range from 0.0 to 1.0

results 2

task main
    copy zippedfiles.zip node:zippedfiles.zip
    node:execute unzip zippedfiles.zip
    node:execute ./OptimizationTABU/geotest_Nimrod.py $x1
    $x2 ... $x36
    node:execute ./meshbash.sh
    node:execute ./fluentbash.sh
    node:execute ./write_results.sh
    copy node:results.txt output.$jobname
endtask

method mots_ii
    starts 1
    resume optimisation 0
    number of regions 4
```

```
size of Short Term Memory 20
intensification 20
diversification 30
stepsize reduction 50
initial step size 0.1
stepsize reduction-factor 0.5
size of sample 2
number of evaluations 300
starting method 0
pattern move mode 1
tolerance 0.001
on error ignore
endstarts
endmethod
```