UNIVERSIDADE
BEIRA INTERIOR

Structural optimization of composite UAV wings

(Versão final após defesa)

Filipe Miguel Jesus Silva

Dissertação para obtenção do Grau de Mestre em

Engenharia Aeronáutica

(Mestrado Integrado)

Orientador: Prof. Doutor Pedro Vieira Gamboa

Dezembro de 2022

Declaração de Integridade

Eu, Filipe Miguel Jesus Silva, que abaixo assino, estudante com o número de inscrição 35176 de/o Mestrado Integrado em Engenharia Aeronáutica da Faculdade de Engenharia, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior.**

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 9 / 12 / 2022

Assinado por: **Filipe Miguel Jesus Silva**
Num. de Identificação: 15413071
Data: 2022.12.09 00:42:07+00'00'

(assinatura conforme Cartão de Cidadão ou preferencialmente
assinatura digital no documento original se naquele mesmo formato)

Acknowledgements

First, I would like to thank to my thesis supervisor, professor Pedro Gamboa, for all the guidance during this work, his availability to meet and discuss as well as for all the ideas suggested for this thesis. I would also like to thank him for letting me make part of the team AERO@UBI for the 2019 edition of the Air Cargo Challenge, since this is one of my personal motivations for this work.

I also would like to express my gratitude to the people that shared some of their time and knowledge with me during these seven years at UBI. A special thanks to Adriana Pinto, Guilherme Farias, Hugo Sousa, Joaquim Reinolds, Ricardo Palmeira, and Ricardo Soares. I learnt a lot from you!

Thank you Catarina.

Last, but not least, I would like to thank my parents and sister for the continued support during these years. Without them, this journey would have not been possible.

Resumo

A massa é um parâmetro essencial durante a fase the projeto de uma aeronave. Uma vez que a asa é um dos componentes mais pesados, uma previsão precisa da sua massa é essencial para a correta definição do seu tamanho e avaliação do desempenho da aeronave. Materiais compósitos compõem frequentemente as estruturas das aeronaves devido à sua elevada resistência específica e facilidade de manufatura. Encontrando a sequência de empilhamento ótima é possível minimizar a massa dos componentes assegurando que constrangimentos como integridade estrutural e deslocamento máximo são cumpridos. Vários estudos mostram que algoritmos evolucionários combinados com análises por elementos finitos são uma boa abordagem para otimizar estruturas em materiais compósitos no que respeita à composição e orientação de cada camada.

Esta dissertação descreve o desenvolvimento de uma ferramenta que executa otimização de asas com estruturas em materiais compósitos de veículos aéreos não tripulados. A motivação para este trabalho assenta na necessidade de melhorar o método de dimensionamento estrutural usado pelas equipas da Universidade da Beira Interior durante a fase de projeto para a competição Air Cargo Challenge. Tipicamente, a asa é uma estrutura em viga bicelular com casca em *sandwich*, almas de longarina e mesas laminadas. No modelo computacional, elementos placa triangulares são usados para representar as almas das longarinas e a casca, e elementos barra para definir as mesas. Para esta estrutura, a sequência de empilhamento da casca deve ser procurada assim como o número de camadas a laminar em cada mesa para minimizar a massa sujeito a constrangimentos de falha e de deflexão (deslocamento da ponta da asa e torção).

Para gerar a malha, a secção transversal da asa é dividida em cinco sub-secções: bordo de ataque, extradorso e intradorso entre as posições das longarinas e almas da longarina principal e da longarina traseira. Com base no número de divisões escolhidos e na técnica de espaçamento para cada sub-secção, são calculados os nós para o problema estrutural. Uma vez que o número de divisões e as técnicas de espaçamento da secção transversal são mantidos ao longo da envergadura, os nós do painel são resultado da interpolação dos nós das secções extremas de cada painel. Após a numeração dos nós, a malha é definida painel a painel, gerando os elementos triangulares para a casca e almas e os elementos lineares para as mesas. As cargas são calculadas usando a teoria da linha sustentadora e transferidas para a malha considerando que são aplicadas na alma e mesas da longarina principal. A ferramenta computacional usa o MYSTRAN para resolver o problema de elementos finitos e avaliar critérios de falha e o algoritmo genético do OpenMDAO para resolver o problema de otimização com inteiros.

Tipicamente, os materiais considerados para o problema são ortotrópicos, incluindo tecidos unidirecionais e bidirecionais. Os constrangimentos de manufatura são

abordados considerando estruturas em *sandwich* simétricas, em que o núcleo é a camada central, e através da orientação dos tecidos unidirecionais das mesas com a direção longitudinal do painel. O número máximo de camadas de cada estrutura laminada e as orientações em que um tecido pode ser aplicado devem ser especificadas pelo utilizador. Para reduzir o número de variáveis de desenho, uma base de dados que contém todas as combinações de material-orientação para cada camada é gerada. Este método permite fixar o número de variáveis de desenho por painel em sete: sequência de empilhamento da casca e das almas das longarinas, e número de camadas de cada uma das mesas de longarina. Devido ao elevado custo computacional do processo de otimização, a opção de realizar processamento em paralelo do OpenMDAO é ativa o que permite a análise simultânea de mais do que um indivíduo da população.

Como caso de estudo, o painel central da asa da aeronave da edição de 2019 do Air Cargo Challenge da AERO@UBI é otimizado, testando a ferramenta de otimização. Uma redução de 16.5% da massa do painel é obtida durante as simulações realizadas. Das diferentes definições de otimização testadas, considera-se que a taxa de mutação de 0.05, tamanho de população de 30 indivíduos e 20 gerações corresponde à combinação que melhor serve este problema de otimização. Dos resultados obtidos, é recomendada a implementação adicional de um constrangimento capaz de medir a diferença entre a geometria deformada e a original no sentido de prevenir perdas aerodinâmicas excessivas.

Palavras-chave

Compósitos, Otimização, Algoritmo Genético, Otimização com inteiros, Tabelas de Empilhamento, Análise por Elementos Finitos

Abstract

Mass is a key factor during the design of an aircraft. Since the wing is one of the heaviest components, an accurate prediction of its mass is essential for a correct definition of its size and evaluation of the aircraft's performance. Composite materials usually compose aircraft structures due to their high specific strength and ease of manufacture. By finding the optimal stacking sequence and layer orientation, it is possible to minimize the mass of components ensuring that constraints such as structural integrity and maximum displacement are fulfilled. Many studies show that Evolutionary Algorithms combined with Finite Element Analysis is a good approach to optimize composite structures regarding layer composition and orientation.

This thesis describes the development of a tool that performs the structural optimization of Unmanned Aerial Vehicles' wings made of composite structures. The motivation for this work lies in the need to improve the structural sizing method used by the University of Beira Interior teams during the design phase for the Air Cargo Challenge competition. Typically, the wing is a two-cell beam structure with sandwich skin, spar webs and laminated spar caps. In the computational model, triangular plate elements are used to represent both spar webs and the skin, and bar elements define the spar caps. For this structure, the stacking sequence of the skin must be found as well as the number of layers of each spar cap for minimum mass subject to failure and deflection (wing tip deflection and twist) constraints.

To generate the mesh, the wings' cross-section is divided into five sub-sections: leading-edge, upper and lower surfaces between spar positions and leading and rear spar webs. Based on the number of divisions and the spacing technique chosen for each sub-section, the section nodes for the structural problem are computed. Since the number of divisions and spacing technique of the cross-section are kept constant across the span, the panel nodes are the result of the interpolation of the section nodes between the extreme sections of each panel. After the node numbering process, the mesh is defined panel-by-panel, generating triangular elements for the panel skin and webs of both spars and linear elements for the spar caps. The loads are computed using the lifting line theory and transferred to the mesh considering that they are applied on the web and caps of the main spar. The solution uses MYSTRAN as the finite element solver to assess failure criteria, and the Simple Genetic Algorithm from OpenMDAO to solve the integer optimization problem.

Typically, the materials considered for the problem are orthotropic, including unidirectional and bidirectional fabrics. Manufacturing constraints are addressed considering symmetrical sandwich structures, in which the core is the central layer, and by orienting the unidirectional fabric of the spar caps with the longitudinal direction

of the panel. The maximum number of layers for each structure and the orientations in which a fabric can be applied must be specified by the user. To reduce the number of design variables, a database containing all possible arrangements of layer's material and orientation for each structure is generated. This method allows to fix the number of design variables per panel to seven: stacking sequences of shell, main and rear spar webs, and number of layers of each spar cap. Since this is a high demanding computational optimization process, the parallel processing option available at OpenMDAO is activated, allowing to simultaneously analyze more than one individual of a generation.

As a case study, the central panel of the 2019 Air Cargo Challenge aircraft wing from AERO@UBI is optimized, testing the optimization tool. A reduction of 16.5% on the panel's mass is achieved during the simulations. From the different optimization settings tested it is considered that a mutation rate of 0.05, population size of 30 individuals and 20 generations is the combination that best suit this optimization problem. From the results obtained, it is recommended to implement an additional constraint able of measuring the difference between the deformed and the original shapes to prevent excessive aerodynamic losses.

Keywords

Composites, Optimization, Genetic algorithm, Integer optimization, Stacking Sequence Table, Finite Element Analysis

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

ACC	Air Cargo Challenge
BDF	Bulk Data File
DCA	Aerospace Sciences Department
FEA	Finite Element Analysis
GA	Genetic Algorithm
LLT	Lifting Line Theory
MPI	Message Passing Information
PSPC	Permanent Single Point Constraint
SF	Safety Factor
SM	Safety Margin
SST	Stacking Sequence Table
UAV	Unmanned Aerial Vehicle
UBI	Universidade da Beira Interior

Nomenclature

b	Wing span [mm]
$bslsd$	Number of divisions between spars on lower surface
$bsusd$	Number of divisions between spars on upper surface
c	Airfoil chord [mm]
c_d	Local drag coefficient
C_i	Constant penalty
c_l	Local lift coefficient
c_m	Local pitching moment coefficient
d_i	Distance metric
E_1	Young's Modulus in direction 1 [MPa]
E_2	Young's Modulus in direction 2 [MPa]
$ETOA$	Expected time of analysis [h]
$f(x)$	Objective function
F_x	Force in x-direction [N]
F_y	Force in y-direction [N]
F_z	Force in z-direction [N]
FT_{Hill}	Tsai-Hill failure theory
$g_j(x)$	Inequality constraints function
$h_k(x)$	Equality constraints function
k	Penalty exponent
ld_i	Longitudinal number of divisions of panel i
led	Number of divisions: leading edge
m	Wing mass [g]
msd	Number of divisions: main spar
N_{gen}	Number of generations
n_{tri}	Number of triangular elements in a given structure
$nodes_i$	Number of nodes of panel i
nwp	Number of wing panels
of	Objective function value
p	Optimization total penalty value
p_{mut}	Probability of mutation
pop_{size}	Population size
rsd	Rear spar number of divisions
S	Wing area [mm²]
S_{12}	In-plane shear strength [MPa]
SAT	Single analysis time [s]
$tot._{elements}$	Total number of elements
$tot._{nodes}$	Total number of nodes

$triangles_i$	Number of triangular elements of panel i
X_c	Compressive strength in direction 1 [MPa]
x_e	Triangular element local x-axis
X_t	Tensile strength in direction 1 [MPa]
x_i	Design variable x
x_{iL}	Lower bound of design variable
x_{iU}	Upper bound of design variable
Y_c	Compressive strength in direction 2 [MPa]
y_e	Triangular element local y-axis
Y_t	Tensile strength in direction 2 [MPa]

Greek symbols

α	Angle of attack [°]
β_i	Violation check of optimization constraints
δ_{tip}	Tip deflection [mm]
γ	Mass per unit area [g/m²]
ρ	Density [kg/m³]
ρ_{air}	Air density [kg/m³]
θ_{tip}	Tip twist [°]
θ_{tri}	Angle between x_e projected on xy-plane and y-axis
$\theta_{sec.}$	Section's incidence angle [°]

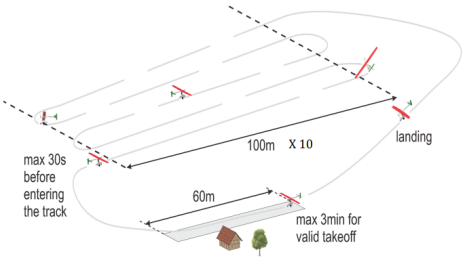
Chapter 1

Introduction

1.1 Motivation

The Air Cargo Challenge (ACC) is an academic competition held every two years in Europe, promoting the concept "Design, Build and Fly". The competition aims to design an unmanned aerial vehicle (UAV) able of carrying the highest payload possible and fly a pre-defined circuit in the shortest time. Since the beginning of the competition in 2003, the Aerospace Sciences Department (DCA) from Universidade da Beira Interior (UBI) recognises the value and importance of this type of challenge in the training process of its students.

Regarding the competition, the score achieved by each team depends on the payload mass carried, and on the time needed to complete a flight course composed of ten legs of one hundred meters each. The aircraft must take-off in 60 m and proceed to execute a flight pattern as observed in Figure 1.1a [1]. The flight path is a set of straights and turns that should be completed in the minimum amount of time. For the 2019 edition of the Air Cargo Challenge competition, held in Stuttgart, Germany, the team from UBI presented a full carbon composite aircraft with a span of 3.89 m (Figure 1.1b). The aircraft's wing is split into five rectangular panels, each one with a length of 745 mm and a chord of 300 m.



(a) Flight path of Air Cargo Challenge competition [1]

(b) UBI's aircraft for 2019 edition of ACC

Figure 1.1: Flight path and 2019 UBI's aircraft

Mass is a key factor in the design process of an aircraft, therefore the need of an accurate prediction of each components' weight is essential to obtain a realistic design. For the ACC competition, as in aeronautics in general, minimizing the mass of the aircraft structure is essential to improve the overall flight score. The development of the aircraft for each edition of the Air Cargo Challenge is subjected to the regulations of the competition that suffered some changes over the years. For the 2019 edition, payload mass and speed

are equally weighted in the flight score equation, so the team needed to develop a fast aircraft able of carrying a considerable amount of payload. The need to fly fast with heavy payload requires higher flexural stiffness and torsional stiffness when compared to the first editions. One can meet these requirements by changing the cross-section and/or the materials used to manufacture the wing. Composite materials have high specific strength, offering flexibility and easiness during the manufacturing process, that combined with a two-cell beam structure composed the solution found by the team in 2019.

Due to the above referred regulations' adjustments, performing an analysis and optimization of the wing structure for the 2019 edition would be, most probably, inapplicable in terms of stacking sequence results for future editions, opening the door to the development of a computational tool able to perform the optimization of composite wings. This approach represents a more flexible solution, allowing future studies and applications in terms of the wings' structural design for a variety of UAVs. Additionally, this soluiton must account for manufacturing constraints in such a way that the structural wing that comes as a result of the optimization process must be possible to manufacture.

1.2 Objectives

The main objective of this thesis is to develop a free, open-source tool able to perform structural optimization of composite UAV wings considering manufacturing constraints. In order to achieve it, several subtasks must be completed:

- Develop a bulk data file (BDF) generator based on essential information, such as material properties, wing geometry and loading, to determine the structural response regarding the displacement and failure;
- Implement the objective function and constraints and integrate those with an external finite element analysis (FEA) software;
- Perform several optimization runs with the implemented tool using different settings, such as mutation rate, pouplation size and number of generations, to study their effects on the optimum design;
- Develop a graphical interface to plot the results from the static analysis.

1.3 Thesis outline

This thesis consists of five chapters, including this one, and the main topics are:

- Chapter 1 presents the motivation for this work and the main objectives;

- Chapter 2 covers the literature review that includes the theoretical background and a brief presentation of similar work;
- Chapter 3 describes the methods used to implement the optimization tool;
- Chapter 4 presents an application of the tool using a case study, the discussion of the results and some considerations regarding the optimization settings;
- Chapter 5 includes the final conclusions and possible future work.

Chapter 2

Literature Review

This chapter provides an overview of related works and introduces key concepts for this thesis. A discussion on similar works is held in first place, followed by a presentation of the terms and definitions associated with optimization and finite element analysis.

2.1 Related work

On the topic of structural optimization of composite structures, some work has been performed using different techniques. A considerable part of the literature covers the analysis and optimization of composite plates subjected to different boundary or loading conditions. In general terms, one can optimize a structure via manual iteration or using an optimization algorithm to search the design space for the best solution.

Regarding the manual design, two examples are found in the literature on the composite UAV wings topic. Silva studied the design of a composite UAV wing using manual adjustements to adapt the structure to the best performance [2]. Rumayshah, Prayoga, and Moelyadi manually designed the wing structure of a high altitude long endurance UAV [3]. In both examples, the approach requires a good understanding of structural design to decide on what type of change must be applied to the stacking sequence in order to improve the structural performance and compliance of the wing.

The more automatic approach usually involves connecting an optimization driver with a structural analysis model. For the case of composite optimization, stochastic methods are frequently applied to solve the problems due to its ability to deal with multivariate functions, discrete variables or non-differentiable functions [4]. Gradient based algorithms can also be applied to composites optimization using, for example, the lamination parameters. This approach ensures a convex design space with continuous design variables. The lamination parameters are usually applied to less complex problems such as the analysis of composite plates or cantilever beams [5]. One drawback of this methodology is the difficulty of converting the resultant lamination parameters into stacking sequences. To solve this issue, Sprengholz et al. [6] studied the conversion process between lamination parameters and stacking sequence of layers. Even though the ply angles and the number of layers may not be restricted, the conversion from the lamination parameters does not lead to a single stacking sequence, meaning that a set of values can be achieved by multiple stacking sequences . Therefore, this approach does not address the manufacturing constraints in the desired manner for the development of

the tool.

Arias-Montaño, Coello and Mezura-Montes reviewed the application of evolutionary algorithms in multiobjective problems of aeronautical and aerospace engineering [7]. This paper presents concrete examples where evolutionary algorithms have been used to optimize designs in distinct areas: computational fluid dynamics (CFD), structural optimization, control design, and physics/shape optimization. These authors refer that the incorporation of optimization techniques during the conceptual and preliminary design phases of an aeronautical/aerospace project is essential to search for viable and manufacturable solutions. Regarding the structural optimization in the aeronautical domain, the design goal is to find lighter and stronger structures, looking for the best compromise between mass and stiffness/strength.

Zhong, Jin and Han implemented a method to optimize a composite wingbox using a master-slave parallel genetic algorithm [8]. In this study, the wingbox was divided along the span and the main goal was to determine the number of layers needed to support the loads without failure in each section. Manufacturing constraints were considered by allowing only predefined orientations.

Hailian and Xiongqing defined the optimization problem of a composite wing from the manufacturing cost point of view [9]. In this study, the goals were to minimize the wing mass and cost determining the best combination of number of spars and its locations, as well as the thicknesses of the skin, spar webs and ribs. Three failure constraints and one geometric constraint were implemented to define the feasible designs. The structural solver used in this work was MSC Patran/Nastran together with a version of a NSGA-II optimization algorithm.

Długosz and Klimek studied the optimal design of a UAV wing structure using composite materials [10]. To this end, they coupled a numerical finite element model of the wing with an evolutionary multi-objective algorithm to perform the optimization of the wing structure. To solve the finite element problem, the MSC Patran/Nastran software was selected. The optimization algorithm used is an evolutionary technique based on the NSGA-II, that employs similar concepts to the genetic algorithm. Twenty four design variables were defined for the problem, and three optimization functionals were selected to compose the objective functions: minimization of the structure's volume, minimization of the maximum equivalent stress value, and minimization of the maximum vertical displacement. As a result of this process, a 27% reduction in the wing's mass value was achieved.

Using a finite element analysis software coupled with an optimization algorithm seems to provide good results regarding mass reduction, however dealing with composites requires attention when addressing the design variables. Stochastic methods in the composites optimization subject, present more flexibility on the definition of the design variables.

To better understand the pros and cons of the optimization of composite structures using gradient or stochastic algorithms one can refer to the work of Ghiasi, Pasini and Lessard [11]. In an attempt to solve the problem associated with the design variables, several solutions were proposed. An example is the development of a dedicated module to help solving composite laminate design of structures using a previous implementation of a genetic algorithm [12]. To avoid the need of developing a complex intermediate step and still address the manufacturing constraints, stacking sequence tables (SST) can be generated for each laminate structure, considering the definition of each layer (material and orientation). This strategy has been applied to the design of blended structures [13] and tested in less complex structures such as the case of panels and stiffened panels [14] ensuring that manufacturing constraints are addressed and respected. When using the SST, each combination corresponds to a line in the table, therefore can be identified by a single integer. The generation of the stacking sequence tables can be performed using different procedures to meet design or manufacturing requirements. In this thesis, all possible combinations for core centered structures are generated, leaving enough margin for the optimizer to explore the design space, trying to find the lightest solution.

From the information presented, it is possible to understand that coupling a finite element analysis software to a genetic algorithm driver is a solution to optimize composite structures. The manufacturing constraints are adressed by generating stacking sequence tables for the laminated structures of the wing.

2.2 Optimization

A brief presentation regarding different optimization techniques can be found in [15] or [16]. Usually, optimization techniques are classified as local or global, commonly associated with gradient-based and non-gradient based (or evolutionary), respectively [15]. Regardless of the optimization type selected, it is necessary to define the optimization problem statement, where the information about objective function ($f(x)$), design variables (x_i) and constraints ($g_j(x)$ and $h_k(x)$) can be found. The general formulation of a constrained optimization problem can be stated as [15][16]:

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && g_j(x) \leq 0,\ j = 1, \ldots, m \\ & && h_k(x) = 0,\ k = 1, \ldots, p \\ & && x_{iL} \leq x_i \leq x_{iU},\ i = 1, \ldots, n \end{aligned} \tag{2.1}$$

In Equation (2.1), $g(x)$ corresponds to the inequality constraints and $h(x)$ to the equality constraints. The bounds of each design variable (x_i) are x_{iL} and x_{iU}, lower bound and upper bound respectively. A feasible solution is such that the values for the design

variables are within the range specified ([x_{iL}, x_{iU}]), and all the constraints are verified.
For the first referred type of optimization method, gradient based, the optimization is performed faster due to the use of information related with the derivatives of the objective function. However, it is not always possible to provide or compute the derivatives of the objective function. In these cases, an evolutionary algorithm might be applied. When using evolutionary algorithms, a penalty term must be added to the original objective function in order to account for the non-feasibility of the designs. Smith and Coit [17] classify the penalties in interior or exterior, depending on wether the penalized solution is feasible or non-feasible, respectively. It is said that with evolutionary optimization techniques, an exterior penalization is commonly selected. Two types of static penalties that are relevant for this work are: constant penalty and distance based penalty.
The constant penalty is a method that penalizes the solutions that violates the constraints by applying a constant penalty. The general formulation of the penalty term (p) for this strategy is presented in Equation (2.2). In this equation, m is the number of constraints, C_i is the constant penalty and β_i is the violation check, meaning that its value is 1 when the *i*-th constraint is violated and assumes the value 0 when it is not.

$$p = \sum_{i=1}^{m} C_i \beta_i \quad (2.2)$$

In the distance based penalty, the magnitude of the penalty term varies with the distance to feasibility. Hence, the more unfeasible design is, the bigger the penalization. Equation (2.3) presents the general formulation for this type of penalty. As presented for Equation (2.2), this penalty term (p) is the result of the sum of the contributions of all *m* constraints, and C_i the constant penalty. d_i^k is the distance metric term, in which k corresponds to the penalty exponent and d_i to the distance metric itself. In the work of Smith and Coit [17], it is also referred that the penalty exponent typically assumes the value of 1 or 2, and the distance metric can be continuous or discrete.

$$p = \sum_{i=1}^{m} C_i d_i^k \quad (2.3)$$

2.2.1 Genetic algorithm

The presentation performed in this section regarding the genetic algorithm is mainly based on "Engineering Design Optimization" by Martins and Ning [16].
A genetic algorithm is a bio-inspired process that uses the concepts of evolution of a population to navigate the search for solutions across the design space. Since it does not use any information about gradients, it is classified as a gradient-free optimization procedure, and can be easily applied to a wide variety of problems. Usually, this optimization algorithm has the ability to handle both integer and continuous variables,

which in part justifies its general use. Examples of the application of the genetic algorithm can be found in the mechanical engineering field [18], control engineering [19], aeronautical engineering [7], and on the optimization of composite structures [20].
This algorithm tries to mimic the evolution process observed in the nature of living beings, therefore the terms associated to this optimization strategy are similar to the ones used in biology and genetics. The evolution process is based on the fitness of the individuals that compose the population, and the three operations associated with this process are: selection, crossover, and mutation. Each iteration of the genetic algorithm is considered a generation and includes the evolution operations that allow to progress from one population to another. The genetic algorithms can also be distinguished by the type of encoding used: binary-encoded or real-enconded. There are some differences concerning the terms and the way some operations are performed between these two types. Since in this work a binary-encoded genetic algorithm is used, the operations described below are more directly related with this type.

Figure 2.1 is a representation of the concepts of individual and population. An individual is defined as the combination of the design variables ($x_1 - x_n$), and can be seen as an array where each design variable is alocated to a slot. Since a binary representation is used to encode the design variables, each slot corresponds to a series of bits. The population corresponds to the group of individuals and its size, pop_{size}, is the total number of individuals, m.

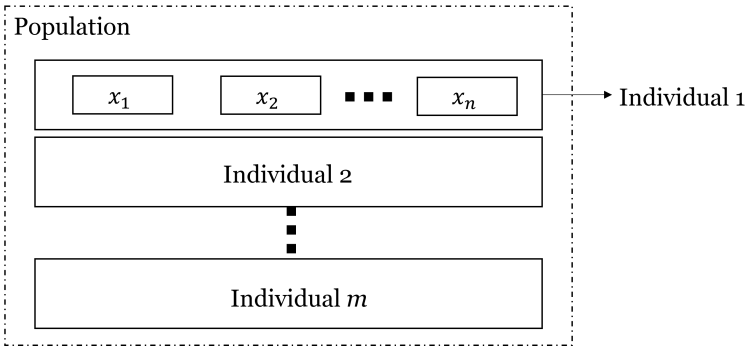


Figure 2.1: Definition of individual and population (adapted from [16])

The selection of individuals is the operation that precedes the crossover and aims to choose the fittest individuals while the diversity of the population is ensured. This process can be performed in different forms, from random selection to the roulette wheel, however the tournament selection is a technique commonly used with the GA's. In the tournament selection process, the individuals in the populations are randomly paired and compared, being selected the fittest element of the pair for the mating pool. A total of $pop_{size}/2$ pairs must be formed for each tournament. This tournament process is repetead until a total of pop_{size} individuals exist in the mating pool, being designated as parents.

Following the selection process, each individual in the mating pool is paired with another element and the crossover operation takes place. The crossover is the process where the information regarding the design variables of both parents is mixed and two new individuals are generated. As for the selection, different crossover techniques exist. However, according to the work of Williams and Crossley [21], the uniform crossover is best for real engineering problems. In this formulation, each bit of a new individual has 50% probability of being from one of the parents.

This set of new individuals will compose the next population after being applied the mutation process. The mutation operation corresponds to the random possibility of changing the value of one bit from 1 to 0 or vice-versa. This operation is applied only if the random value, between 0 and 1, generated for the parameter p of a bit, p_{bit}, is larger than the mutation rate (p_{mut}) defined for the problem. This operation is responsible for the random component of search of the genetic algorithm.

Another feature that is frequently used when applying a genetic algorithm to an optimization problem is elitism. The elite corresponds to the individual of a population with best fitness value. The concept behind the elitism is to prevent the loss of the information associated with the best individual during the optimization due to the crossover and mutation. In this sense, the elite of a population will replace the worst performing point of the next population, ensuring that the information of the current best performing individual is maintained. This feature is only applicable after the first generation.

In the case of genetic algorithms, the user must decide on the maximum number of generations (n_{gen}) that can be performed, since it is not common to have a convergence criteria as usually exists for gradient-based optimizations. The flowchart representation of the genetic algorithm is presented in Figure 2.2. The evolution process stops when the maximum number of generations has been reached. Since an initial random population is needed to initiate the optimization process, the total number of populations studied is always one more than the number of generations selected (n_{gen}). Equation (2.4) presents the total number of individuals that should be analyzed during an optimization process with n_{gen} generations.

$$n_{individuals} = (n_{gen.} + 1) \cdot pop_{size} \quad (2.4)$$

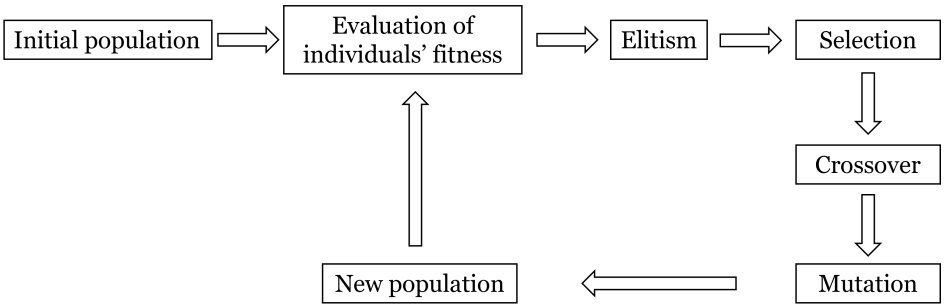


Figure 2.2: Flowchart of the genetic algorithm

2.2.2 Optimizer

Selecting the optimizer is an important step when defining the tool and interfaces. This process must attend to the problem's needs, therefore a series of requirements are defined to select an adequate optimizer:

- Must contain/or be a genetic algorithm driver;
- Should be able of handling integer variables;
- Might allow parallel processing to speed up the optimization;
- An open-source tool is desirable.

The OpenMDAO[1] is a python framework developed to perform multidisciplinary design, analysis and optimization [22]. The development of this library focus mainly on gradient-based optimization techniques, such as the adjoint method, but it also includes a genetic algorithm driver called "SimpleGA". A wide variety of engineering design optimization problems have been studied using this tool and the applications range from a CubeSat electric energy design [22] to low-fidelity wing aerostructural optimization [23].

The genetic algorithm driver is based on the lectures notes of Professor William A. Crossley at Purdue University and uses a binary encoded representation for the design variables, allowing both integer and real types. In the case of a real variable, the precision of its value depends on the number of bits used in its representation. For integer variables, the number of bits needed for a correct representation is automatically computed during the execution of the GA. Regarding the bounds of the design variables, if the upper and lower limits are not a power of two, the variables during the optimization process can assume values outside of the defined range. These elements, whose design variables' values are outside the defined range, are not analyzed affecting the evolution process [24]. Parallel processing is also available and uses OpenMPI as default tool for Message Passing Information (MPI).

[1] All the information about OpenMDAO can be found at `https://openmdao.org/`.

2.3 Structural analysis

The analysis of composite plates is split into two groups: Equivalent single-layer theories (ESL) and Three-dimensional elasticity theory. The first group is a simplification of the second, via assumptions related with the thickness and deformation, which leads to a lower computational cost to solve the problem [4]. One of the ESL theories is the classical laminated plate theory (CLPT), that corresponds to an extension of the Kirchoff plate theory to laminated plates [25]. Since a high number of function evaluations is expected, it is desired to reduce the computational cost of each analysis, justifying the selection of an ESL approach for the plate elements that compose the shell and the spar webs. This theory has been applied in the design of composite plates by different authors (e.g.: [26] [27]).

To assess failure in composite structures, a failure criteria that weighs the stress levels in the different directions is used. The Tsai-Hill criterion is an intralamina analysis of failure and is also considered an extension of the von-Mises analysis for isotropic materials [28]. Equation (2.5) is the formulation of Tsai-Hill criterion for the in-plane failure analysis. Failure is expected to occur when $FT_{Hill} \geq 1$.

$$FT_{Hill} = \left(\frac{\sigma_1}{X}\right)^2 - \frac{\sigma_1 \sigma_2}{X^2} + \left(\frac{\sigma_2}{Y}\right)^2 + \left(\frac{\tau_{12}}{S}\right)^2 \quad (2.5)$$

σ_1 - Applied direct stress in direction 1

σ_2 - Applied direct stress in direction 2

τ_{12} - Applied shear stress

X - Material's strength in direction 1, can be X_c or X_t

Y - Material's strength in direction 2, can be X_c or X_t

S - Material's shear strength

2.3.1 FEA solver

The software to use as solver of the static structural analyses must present a set of features needed to solve the problem:

- Should have plate elements to represent the shell and webs;
- Bar type elements must be available to represent the spar caps;
- Composite structures analysis needs to be supported;
- Import loads that can mimic realistic conditions;

- Perform static structural analysis;
- Open-source/free software;
- Interface using a file, a group of files or a script.

There are some commercial solutions with the capability of performing analysis of composite structures. Two examples of these solutions are Optistruct/Hypermesh from Altair, and ANSYS with the ANSYS Composites PrepPost (ACP) and static structural analysis modules. In fact, the use of one of these softwares could ease the methodology in the sense that they offer a full package concerning the formulation of elements, solvers, and post-processing. However, a license is required to use these softwares. Due to this reason, since an open-source solution is desired, none of these options was chosen.

MYSTRAN[2] is a free finite element analysis software able to solve problems in which the structures' displacementes and stresses vary linearly with the applied loads. The development of this program has been performed mainly by Dr. Bill and uses Fortran 90 as programming language. More recently, a super LU fatorization solver was added allowing faster computation of expensive problems [29].
This solver offers a wide variety of elements that includes triangular and quadrilateral plate elements, rigid elements, bar and rod elements and have composites support based on the classical laminate plate theory. Concentrated loads can be applied on a grid point to simulate the aerodynamic loading. The input for an analysis is a bulk data file that carries all the information related to the case and the standard output is a F06 file that contains the data about the deformed structure and failure theory values and safety margins. Regarding solution types, MYSTRAN is capable of performing four different types of analysis, being a static simulation one of them [30].

[2]MYSTRAN code is available at `https://github.com/dr-bill-c/MYSTRAN`.

Chapter 3

Methodology

This chapter presents the methods used in the resolution of the problem and it is divided into three sections: Models, Optimization and Implementation. The first section presents the methods used to generate the mesh, the materials definition, and to compute and transfer the loads. In the Optimization section, the constraints, mass function, and objective function used in the optimization tool are presented. The last section provides an overview of the software's structure and how the analysis is set using a bulk data file.

3.1 Models

3.1.1 Geometry & mesh

In terms of geometry, the wing is a composition of tapered panels as can be observed in Figure 3.1. Each panel is the result of the linear interpolation of its root and tip sections. The wing's structure may be a single or double-cell beam and two or four spar caps can be selected for the analysis. In any case, the structure is made up of a shell, two spar webs, and spar caps located on the top and bottom of the webs. Plate elements are used to represent the shell and webs, while bar elements are used to define the caps from a structural standpoint. The reference frame selected uses an X-axis oriented towards the trailing edge, an Y-axis pointing in the spanwise direction of the right wing, and a Z-axis pointing upward. The origin of this frame of reference is located at the vertical plane of symmetry of the aircraft, at the chord quarter location, at an height equal to half thickness.

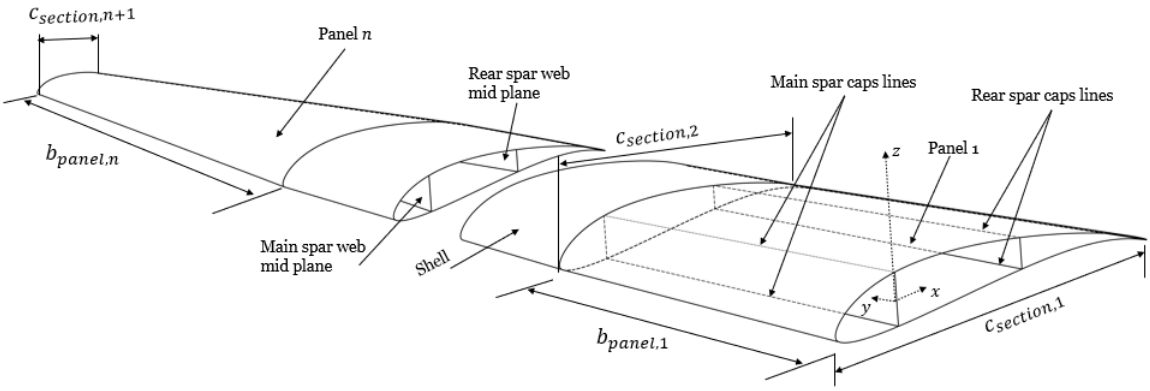


Figure 3.1: Wing's geometry and structure

The wing's cross-section is divided into three areas namely: leading edge, wing box

and flap/aileron (Figure 3.2). For structural purposes, the flap/aileron portion is not considered, so a single cell structure corresponds to an analysis of the wing box, while a double-cell arrangement includes the leading edge portion as well. The spar caps are located on the top and bottom of the spar webs and are centered at the middle of the shell thickness. A two spar cap configuration only uses the main spar caps whereas a four cap setting includes the main and rear ones.

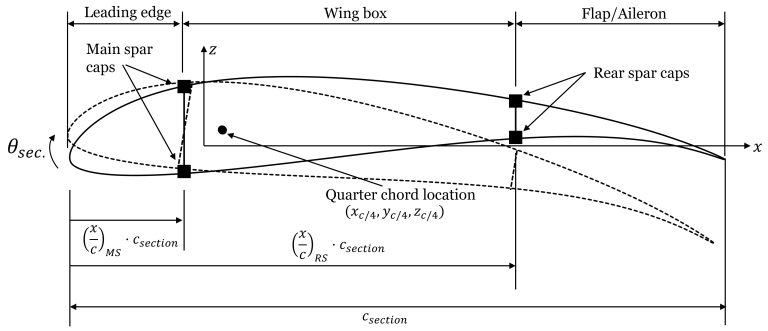


Figure 3.2: Cross-section parameters

Since the wing is defined using linear interpolations of the sections, the user only needs to specify some parameters related to each section such as the airfoil, chord ($c_{section}$), incidence ($\theta_{section}$), main and rear web non-dimensional positions ($\left(\frac{x}{c}\right)_{MS}, \left(\frac{x}{c}\right)_{RS}$) and the chord quarter location.

The meshing process is performed for each panel at a time and is composed of four steps:

1. Compute nodes of the root and tip sections of each panel (these sections are defined by the user);

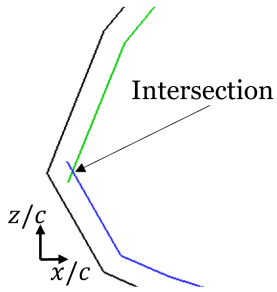
2. Interpolate the nodes coordinates of sections between the root and tip of each panel;

3. Generate the triangular elements;

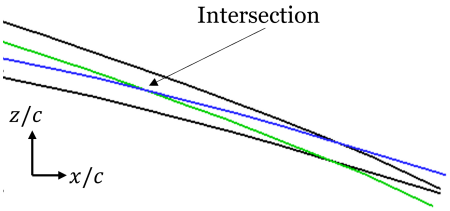
4. Define the linear elements.

The meshing process corresponds to a dedicated implementation developed during this work for the cross-section in use, supported by previous works as observed in the next steps. The first step of meshing is divided into three sub-steps: airfoil offset; scaling, rotation of points and positioning; and cross-sections nodes' interpolation. By the end of this first step, one must have the coordinates for the points of the *n-th* section, $\left\{x_{i,f}, y_{i,f}, z_{i,f}\right\}_n$.

The computation of the airfoil offset uses a procedure similar to the one developed by Sousa et al. [31] to study the effects in the aerodynamic performance of airfoils obtained via a geometric offset. This method computes the inwards' offset of the upper and lower splines and explores different formulations for the leading edge. The inward offset results from applying a translation to the original point towards the inside of the airfoil in the normal direction regarding the local first derivative. One can observe that the leading edge formulation played an important role in the aerodynamic study carried out by Sousa, however it is not particularly relevant for a structural analysis, provided that the mesh is obtained via linear interpolations. Using any of the leading edge corrections presented in his work proved to be complex and far from automatic, which is not desirable considering that the mesh should be generated without the user intervention during the optimization process regardless the offset value. Therefore, one just needs to ensure that the process applied to compute the leading edge does not generate errors, which can be achieved by considering it as "sharp". A sharp leading edge results from trimming the inwards' splines at the intersection point, eliminating the points that present a non-dimensional horizontal coordinate ($(x/c)_{point}$) smaller than the intersection horizontal value ($(x/c)_{inter.LE}$) (Figure 3.3a). During these operations, it is checked if the intersection point is located before the main spar web position to ensure that the geometry generated is correct. For the trailing edge section, the approach is similar, deleting the points behind the intersection (Figure 3.3b), but in this case an additional check is required to verify that the intersection point is located after the rear spar web location. The offset value corresponds to the ratio between half of the shell's thickness and the section's chord.



(a) Leading edge intersection due to inward offset (green and blue lines)



(b) Trailing edge intersection due to inward offset (green and blue lines)

Figure 3.3: Airfoil offset: leading and trailing edge intersection points

The second sub-step begins with scaling for the chord, and re-centering the offset airfoil around the point $(0.25, t_{0.25}/2)$, in which $t_{0.25}$ is the airfoil's thickness value for (x/c) equal to 0.25. This re-centering is required since the user defines the quarter chord location at the input. Equation (3.1) corresponds to this re-positioning and scaling operations, considering that after the offset the i-th point of the airfoil from section n has coordinates

$\left\{x_{i,1}, y_{i,1}, z_{i,1}\right\}_n$.

$$\begin{Bmatrix} x_{i,2} \\ y_{i,2} \\ z_{i,2} \end{Bmatrix}_n = c_{section,n} \left(\begin{Bmatrix} x_{i,1} \\ y_{i,1} \\ z_{i,1} \end{Bmatrix}_n - \begin{Bmatrix} 0.25 \\ 0 \\ t_{0.25}/2 \end{Bmatrix}_n \right) \quad (3.1)$$

After scaling and re-centering, the airfoil points are rotated to match the incidence angle of the section *n*, $\theta_{sec.,n}$, using Equation(3.2). This rotation occurs in the vertical plane, therefore no changes are applied to the spanwise coordinate of the point.

$$\begin{Bmatrix} x_{i,3} \\ y_{i,3} \\ z_{i,3} \end{Bmatrix}_n = \begin{bmatrix} \cos\theta_{sec.,n} & 0 & \sin\theta_{sec.,n} \\ 0 & 1 & 0 \\ -\sin\theta_{sec.,n} & 0 & \cos\theta_{sec.,n} \end{bmatrix} \begin{Bmatrix} x_{i,2} \\ y_{i,2} \\ z_{i,2} \end{Bmatrix}_n \quad (3.2)$$

If the section that is being computed is an interface between two panels, one should account for the difference in the dihedral angles of the panels. In these cases, an adjustment of the spanwise position is applied, using the average value of the dihedral angles (Equation (3.3)). To compute the dihedral angle of the *n*-th panel (Γ_n), the quarter chord positions of the *n*-th and *(n+1)*-th sections are used, as can be observed in Equation (3.4).

$$\begin{Bmatrix} x_{i,4} \\ y_{i,4} \\ z_{i,4} \end{Bmatrix}_n = \begin{Bmatrix} x_{i,3} \\ y_{i,3} \\ z_{i,3} \end{Bmatrix}_n - \begin{Bmatrix} 0.0 \\ (z_{i,3})_n \sin\left(\frac{\Gamma_{n+1}-\Gamma_n}{2}\right) \\ 0.0 \end{Bmatrix}_n \quad (3.3)$$

$$\Gamma_n = \arctan\left(\frac{z_{n+1} - z_n}{y_{n+1} - y_n}\right) \quad (3.4)$$

To adjust the section's dihedral, a rotation about the *X*-axis is applied using the panel's dihedral angle (Equation (3.5)). If the section under calculation does not belong to two panels, which is true for the wing's root section, the array that multiplies the rotation matrix has subscript 3 (referring to Equation (3.2)) instead of 4.

$$\begin{Bmatrix} x_{i,5} \\ y_{i,5} \\ z_{i,5} \end{Bmatrix}_n = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Gamma_n & -\sin\Gamma_n \\ 0 & \sin\Gamma_n & \cos\Gamma_n \end{bmatrix} \begin{Bmatrix} x_{i,4} \\ y_{i,4} \\ z_{i,4} \end{Bmatrix}_n \quad (3.5)$$

Before interpolating the cross-section nodes, it is necessary to move the points for the desired location, using the coordinates provided for the quarter-chord position of each section.

$$\begin{Bmatrix} x_{i,f} \\ y_{i,f} \\ z_{i,f} \end{Bmatrix}_n = \begin{Bmatrix} x_{i,5} \\ y_{i,5} \\ z_{i,5} \end{Bmatrix}_n + \begin{Bmatrix} x_{c/4} \\ y_{c/4} \\ z_{c/4} \end{Bmatrix}_n \tag{3.6}$$

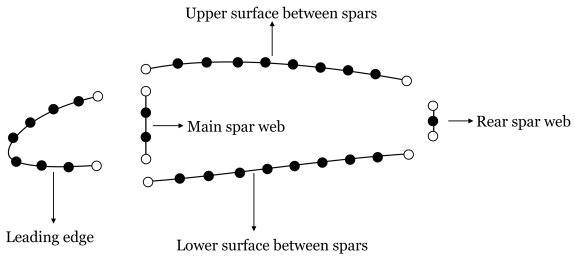
As a final substep of the first step of meshing, the cross-section nodes of the root and tip section of each panel are interpolated from the group of points calculated in the previous two substeps. Figure 3.4a shows that the cross-section is divided into five segments for the double-cell structure, and four segments for the case of a wingbox analysis. Four spacing techniques are available (Equation (3.7)) to compute the nodes of each cross-section's segment.

$$\begin{cases} S1(i, ndiv) = (i-1)\frac{1}{ndiv}, & \text{uniform} \\ S2(i, ndiv) = 1 - \cos\left(\frac{\pi}{2ndiv}(i-1)\right), & \text{cosine refined at first end} \\ S3(i, ndiv) = \cos\left(\frac{\pi}{2} - \frac{\pi}{2ndiv}(i-1)\right), & \text{cosine refined at second end} \\ S4(i, ndiv) = \frac{1-\cos\left(\frac{\pi}{ndiv}(i-1)\right)}{2}, & \text{cosine refined at mid} \end{cases} \tag{3.7}$$

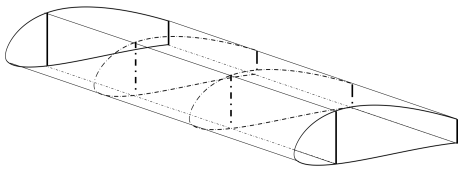
Table 3.1 presents the nomenclature selected for the number of divisions of each segment. These terms are used later in this section to compute the number of nodes and elements.

Table 3.1: Nomenclature of number of divisions

Nomenclature	Segment
msd	Main spar web
rsd	Rear spar web
bslsd	Lower surface between spars
bsusd	Upper surface between spars
led	Leading edge
ld_i	Longitudinal (spanwise) *i*-th panel



(a) Cross-section divisions



(b) Spanwise linear interpolation of sections (interpolated sections are represented using dashed lines)

Figure 3.4: Cross-section and longitudinal interpolations

In Equation (3.7), $ndiv$ corresponds to the segment's number of divisions, and i to the point that is being computed, with i within the range $[0, (ndiv + 1)]$. The result of applying any spacing technique is a series of points with values between zero and one,

that corresponds to percentages of the segment's length (ℓ). The nodes are positioned in the segment performing the product between the segment's length and the result of the spacing technique. The node's coordinates are obtained via interpolation of the points calculated through Equations (3.2) - (3.7), using the absolute position inside the segment as reference. The absolute position of the *i*-th point inside of a segment, corresponds to the distance between that point and the beginning of the segment (s_i), and is calculated by adding the distances between points until reaching it (Equation (3.8)). Figure 3.5 is the representation of a general segment of length ℓ, in which three nodes are computed (two divisions).

$$s_i = \sum_{i=2}^{i} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (3.8)$$

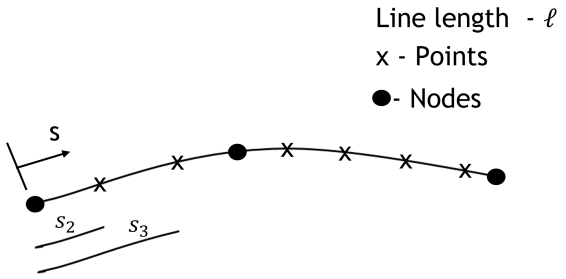


Figure 3.5: General representation of interpolation of nodes in a cross-section segment

The second step of the meshing process, computing the nodes of the interpolated sections of a panel, is achieved applying an equal procedure to the one presented to calculate the nodes in the cross-section (Figure 3.4b). After computing all the nodes of the structure, the nodes numbering process starts on the wing's root section, with the nodes that belong to the lower suface between spars and a clockwise route around the section's shell is performed, followed by the rear spar and the main spar webs' nodes. After numbering the nodes of the root section, it advances to the next interpolated section and the process is repeated. Equation (3.9) is the definition of the total number of nodes per panel ($nodes_i$) for both structural configurations available, wingbox and double-cell, given the number of divisions of each segment.

$$nodes_i = \begin{cases} (msd + rsd + bslsd + bsusd + 1) \cdot (ld_i + 1), & \text{Wingbox} \\ (msd + rsd + bslsd + bsusd + led) \cdot (ld_i + 1), & \text{Double-cell} \end{cases} \quad (3.9)$$

The third step of the meshing process is the generation of the triangular elements. Each triangle is defined by assigning three nodes. Figure 3.6 shows that the definition of the triangular elements and its local referential (x_e, y_e), depends on the local numbering of the triangle's nodes (Node 1, Node 2, and Node 3). The x_e axis points from the local node

1 towards local node 2, while the y_e axis points in a normal direction with respect to the x_e axis, and is considered positive if pointing in the sense of local node 3. The local reference frame is in the plane defined by the three nodes.

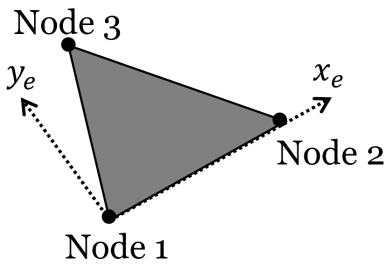


Figure 3.6: Local reference frame and numbering of triangular elements

The numbering of triangular elements occurs in similar way to the nodes numbering, and is performed panel by panel, starting with the numbering of the elements that belong to the shell, then the elements that are part of the rear spar web, and finally the elements that belong to the main spar web. The methodology applied to number the triangular elements is presented in Figure 3.7.

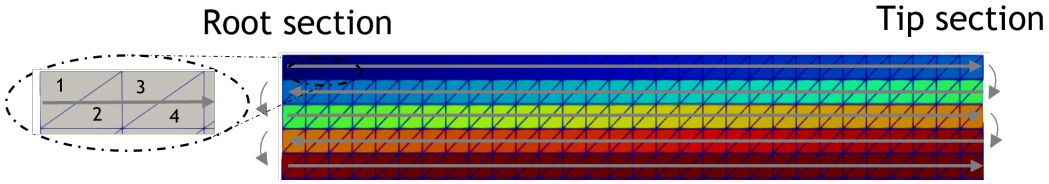


Figure 3.7: Numbering of triangular elements (range starts in dark blue and goes up to dark red)

Equation (3.10) gives the number of triangular elements used to represent each panel ($triangles_i$) for any structural arrangement.

$$triangles_i = \begin{cases} 2ld_i \left(\frac{nodes_i}{ld_i+1} - 1\right), & \text{Wingbox} \\ 2\left(nodes_i - ld_i - \frac{nodes_i}{ld_i+1}\right), & \text{Double-cell} \end{cases} \tag{3.10}$$

To find the orientation of each triangle with respect to the panel's frame of reference, ($\theta_{element}$), Figure 3.8, one uses the dot product between the versors of x_e and y (Equation (3.11)). Knowing this angle is important to later adjust the orientation of each fabric to the element, ensuring that the material is correctly rotated to match the desired orientation.

$$\theta_{element} = \arccos\left(\frac{\hat{x_e} \cdot \hat{y}}{\|\hat{x_e}\|\|\hat{y}\|}\right)\frac{180}{\pi} \tag{3.11}$$

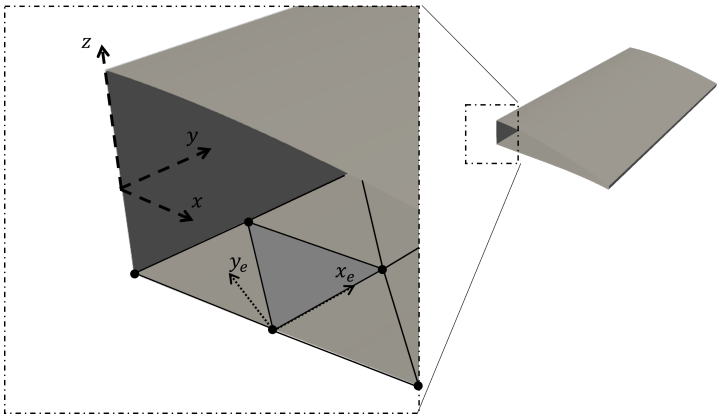


Figure 3.8: Relation between local reference frame of triangular elements and panel's global frame of reference

The fourth and final step of the meshing process corresponds to the definition of the bar elements that represent the spar caps. To set a bar element, one must define the two nodes that correspond to the element's ends. The numbering of these elements depends on the number of caps selected, but the general procedure is to start numbering, from the root towards the tip, the main spar caps, followed by the rear spar caps (if applicable). In any case, one begins with the upper cap and after finishing this structure proceeds to number the lower cap. As expressed in Equation (3.12), the number of bar elements per panel is independent of the number of cells, being dependent only on the number of spar caps activated, either two or four, and on the spanwise number of divisions (ld_i).

$$bar_i = \begin{cases} 2ld_i, & \text{Two spar caps} \\ 4ld_i, & \text{Four spar caps} \end{cases} \tag{3.12}$$

The computational cost of the finite element analysis is set by the total number of nodes ($tot._{nodes}$) and elements ($tot._{elements}$). Equation (3.13) shows that these values are obtained by summing the number of nodes and elements of each panel.

$$\begin{cases} tot._{nodes} = \sum_{i=1}^{nwp} nodes_i \\ tot._{elements} = \sum_{i=1}^{nwp} (bar_i + triangles_i) \end{cases} \tag{3.13}$$

3.1.2 Materials, spar caps, and laminated structures

Two different types of structures compose the wing: spar caps, and laminates. Different materials must be defined based on the structural purpose of each of these structures. For the spar caps, since they are made of unidirectional fibers, isotropic materials are selected provided that these components must carry the direct stresses, mainly, due to

bending. Orthotropic materials must be considered for laminated constructions, which are divided into two categories: cores and fibers.

Figure 3.9 is a representation of a spar cap. Typically, a cap has a rectangular cross-section with a total thickness t_{cap}, and width w_{cap}. This structure is the result of stacking n_{layers} layers of a unidirectional material with thickness per layer equal to t_{layer}. The spar caps are represented using "BAR" elements (Table 3.7) and failure is assessed via the safety margin parameter (SM) (Equation (3.14)).

$$SM = \frac{\sigma_{applied}}{\sigma_{allowable}} - 1 \quad (3.14)$$

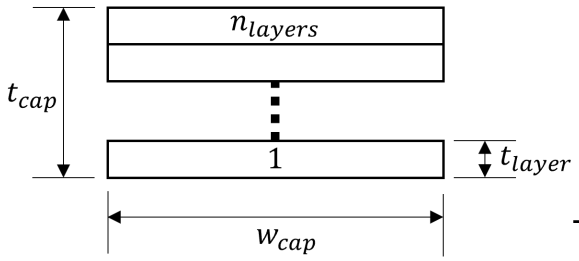


Figure 3.9: Cap nomenclature

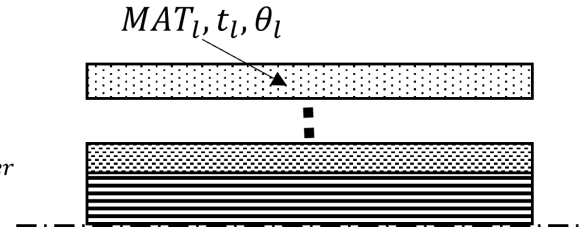


Figure 3.10: Nomenclature of layered composite

In which concerns the laminated structures, shell and spars' webs, to define the stacking sequence tables it is required to know the materials assigned to each structure as well as the maximum number of layers. Each line of the table is a unique combination of the materials available for a given number of layers. Each layer of a laminate is then defined by a material (MAT_l), thickness (t_l) and orientation (θ_l) (Figure 3.10). Since a fiber fabric or a core might assume more than one orientation, each material-orientation arrangement is considered a different material. The size of an SST, n_{comb}, is then given by Equation (3.15), in which n_{cores} represents the number of material-orientation combinations for core materials, n_{fibers} the total number of arrangements for fiber materials, and $n_{max.layers}$ is half of the maximum number of layers allowed for a given structure. For the cases in which the total number of layers is an odd number, this parameter is computed considering the next integer.

$$n_{comb} = n_{cores} \times \sum_{i=1}^{n_{max.layers}-1} n_{fibers}^{i} \quad (3.15)$$

The technique applied to generate the stacking sequence tables focuses symmetric core centered structures and considers all the materials available for a given structure. Due to symmetry, just one half of the sequence is computed, therefore the sequence's length

starts on two layers and ends at $n_{max.layers}$.
To ease the explanation of the applied method, an example of an SST for a set of materials and layers is presented. Let us consider a structure for which the user wants to allow a maximum number of layers of six and the materials are defined in Table 3.2. A unique identifier (SST ID) is assigned to each material-orientation arrangement .

Table 3.2: Materials for SST example

Type	Material	Orientation [°]	SST ID
Core	Balsa	0	1
Fiber	CFRP	0	2
Fiber	CFRP	15	3
Fiber	CFRP	45	4

According to Equation (3.15), there are twelve sequences considering three different fiber materials, one core and $n_{max.layers}$ equal to three. Table 3.3 presents the respective stacking sequence table for this example. Each sequence is a unique combination of materials and number of layers as intended. At this stage, the mass per unit area of the composite ($\gamma_{composite}$) is also computed for each sequence, using Equation (3.16). The contribution of each layer to the composite's mass is weighted using the ratio between the layer's thickness (t_i) and the total thickness of half sequence ($t_{total,seq.}$). The thickness of the central core layer is considered to be half of the layer's thickness value. $\gamma_{material,i}$ is the mass per unit area of the material used in the *i*-th layer.

$$\gamma_{composite} = 2 \sum_{i=1}^{layer} \frac{t_i}{t_{total,seq.}} \cdot \gamma_{material,i} \quad (3.16)$$

Table 3.3: Example of an SST

Sequence	Layer 1	Layer 2	Layer 3
1	1	2	-
2	1	3	-
3	1	4	-
4	1	2	2
5	1	2	3
6	1	2	4
7	1	3	2
8	1	3	3
9	1	3	4
10	1	4	2
11	1	4	3
12	1	4	4

A correction to the layer's angle must be applied to ensure that each ply is properly oriented in the structure. This need appears from the fact that each layer should be oriented with respect to the element's frame of reference. Considering that a triangular

element can be rotated $\theta_{element}$ degrees relative to the global reference frame (Figure 3.8), and that the values defined by the user in Table 3.2 are relative to the panel's reference frame ($\theta_{material,SST}$), Equation (3.17) gives the final orientation of each ply ($\theta_{final,material}$). Figure 3.11 exposes this correction from a graphical point of view. In this picture, (M1, M2) are the material axes, (x_e, y_e) the local frame of reference of the triangular element, and (X, Y, Z) the global reference frame.

$$\theta_{final,material} = \theta_{material,SST} - \theta_{element} \quad (3.17)$$

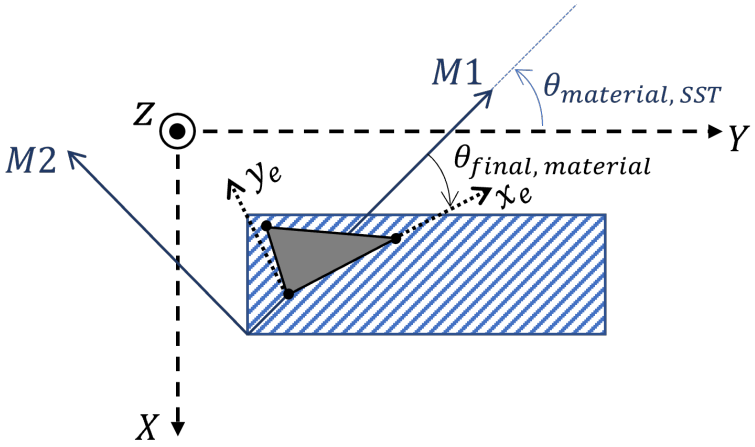


Figure 3.11: Relation between local reference frame of triangular elements and global frame of reference of the problem

3.1.3 Loads and boundary conditions

Since the structure is optimized for a set of operating points, that can be predicted using a V-n diagram, for example, the input related with loads includes an aerodynamic part. Additionaly, a section to insert a static load is available which allows to search for solutions that comply not only with flying loads but also with static loads due to testing, operation requirements or loads from other structures. Both loadings are defined along the span and transferred, as point forces in the X, Y and/or Z directions, to the nodes that belong to the bar elements of the main spar.

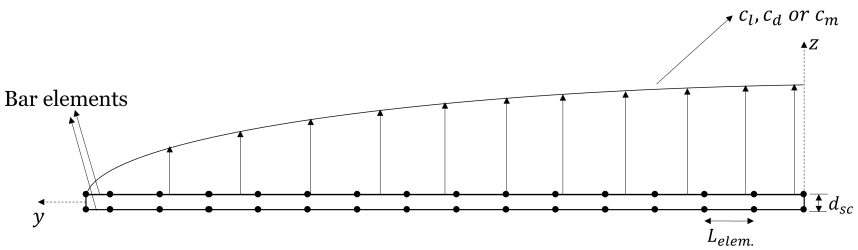


Figure 3.12: Aerodynamic load over wing

Figure 3.12 is an example of an aerodynamic loading over a wing. In this work, the distributions of lift, drag, and pitching moment coefficients are predicted using the lifting line theory. The distributions computed are given at the chord quarter location along the wing. Before computing the loads at each node, one must convert the non-dimensional distributions of coefficients of lift (c_l) and drag (c_d) into non-dimensional distributions of coefficients of force in the X and Z directions, $\{C_x, C_z\}_{aero}$ (Equation (3.18)). This rotation uses the local angle of attack (α) to relate both frames of reference.

$$\begin{Bmatrix} C_X \\ C_Z \end{Bmatrix}_{aero} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{Bmatrix} c_d \\ c_l \end{Bmatrix} \quad (3.18)$$

To obtain the values of the forces per unit length at the position of a node from a bar element, one starts by doing a linear interpolation over the non-dimensional distributions of coefficients of force in the X and Z directions, using the spanwise position as reference, to get the values for the coefficients of force ($\{\bar{C}_x, \bar{C}_z\}_{aero}$). Then, these coefficients are multiplied by the dynamic pressure and the local chord (Equation (3.19)). In this step, ρ_{air} is the air density, V_{ac} is the aircraft speed, and $c_{loc.}$ the local chord.

$$\begin{Bmatrix} f_X \\ f_Z \end{Bmatrix}_{aero} = \frac{1}{2}\rho_{air} \cdot V_{ac}^2 \cdot c_{loc.} \begin{Bmatrix} \bar{C}_X \\ \bar{C}_Z \end{Bmatrix}_{aero} \quad (3.19)$$

The total force supported by two bars, upper and lower caps, is computed using the trapezoidal rule for integration. For each direction, one uses the average value of force per unit length ($\bar{f}_X, \bar{f}_Z$) multiplied by the element's length ($L_{elem.}$). This value is then divided by four, considering that all four nodes (two from the upper element and two from the lower bar) support equal value of force (first term of Equation (3.20)). The second term of Equation (3.20) corresponds to the definition of the load due to the aerodynamic pitching moment coefficient. The aerodynamic pitching moment is converted into a pair of forces that act in opposite directions along the X-axis. The value of this force depends on the vertical distance that separates both caps (d_{sc}), on the dynamic pressure ($\frac{1}{2}\rho_{air}V_{ac}^2$), on the local chord value ($c_{loc.}$), and on half of the element's length ($\frac{1}{2}L_{elem.}$). It is considered that, for a positive moment coefficient, the node located in the upper cap takes a positive contribution in the X-direction, while the opposite occurs for the lower node. One should note that the total force applied at a node, includes the contributions from all the elements that a node belongs to.

$$\begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{node,aero} = \frac{1}{4} \cdot L_{elem.} \begin{Bmatrix} \bar{f}_X \\ 0 \\ \bar{f}_Z \end{Bmatrix}_{aero} \pm \frac{L_{elem.}\rho_{air}V_{ac}^2 c_{loc.}^2}{4d_{sc}} \begin{Bmatrix} \bar{c}_m \\ 0 \\ 0 \end{Bmatrix} \quad (3.20)$$

The input of the static loading should consider that these loads are transferred just to the closer vertical pair of nodes. This way, the total load due to forces, first term of Equation (3.21), is divided by two, and the second term does not include the integration over the element's length. The consideration about the sign of the term related with moments is still valid for this case. Equation (3.21) corresponds to the definiton of the nodal forces due to the static loading.

$$\begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{node,static} = \frac{1}{2} \begin{Bmatrix} F_X \\ 0 \\ F_Z \end{Bmatrix}_{static} \pm \frac{1}{d_{sc}} \begin{Bmatrix} M_Y \\ M_X \\ 0 \end{Bmatrix}_{static} \tag{3.21}$$

In this thesis, the static loading is useful to transfer the aerodynamic loads of the outer panels to the central one. As a result of this procedure, all forces due to lift, drag, and moments are concentrated at the pair of nodes located in the tip section. This approach may lead to some local effects regarding stress anaylsis, that are adressed in the contraints section. M_Y corresponds to the result of integrating the aerodynamic moment over the two outer panels, while M_X is the bending moment due to the vertical forces in these panels.

The load at each node is obtained via superposition of both loadings: aerodynamic and static (Equation (3.22)).

$$\begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{node} = \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{node,aero} + \begin{Bmatrix} F_X \\ F_Y \\ F_Z \end{Bmatrix}_{node,static} \tag{3.22}$$

The boundary conditions are applied by restricting the node's dregrees of freedom (DoF) at its definiton. Each degree of freedom is represented by a number from 1 to 6, as can be seen in Table 3.4. A restriction is applied when a number or a group of numbers are associated to a node. For example, a fixed condition is represented by the sequence '123456', where all DoF are restriced. For this problem, the boundary conditions are applied at the root section, considering a symmetry condition for the shell, and webs (24), and a fixed support for the caps (123456).

Table 3.4: Definition of degrees of freedom

Number	Degree of freedom
1	Linear displacement in X direction
2	Linear displacement in Y direction
3	Linear displacement in Z direction
4	Rotation over X axis
5	Rotation over Y axis
6	Rotation over Z axis

3.2 Optimization problem

This section describes the optimization component of the computation tool by presenting the implemented constraints and the objective function. A total of four constraints are presented: two are geometric and two are failure-related. The mass of the wing is computed using information about the selected stacking sequences as well as the number of layers in each spar cap, and makes part of the objective function together with the penalties.

3.2.1 Constraints

Constraints are an important part of the optimization problem, since they allow to exclude non-viable solutions through the penalization of the objective function. In this case, it is important to check the deformation of the wing and structural failure, therefore four constraints are implemented: tip deflection and twist, and composite and caps failure.

3.2.1.1 Deflection (C1)

To quantify the vertical displacement of the wing tip (δ_{tip}), the difference of the z-coordinate of point E is used. In Figure 3.13, points A and B correspond to the intersections of the main spar with the shell of the tip section from the original wing shape while A' and B' are the intersections for the deformed shape.

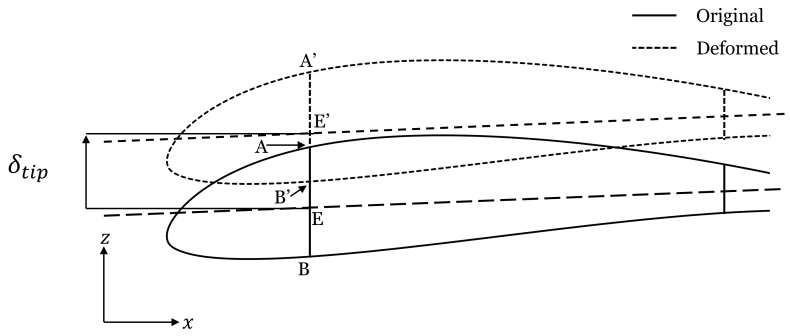


Figure 3.13: Scheme of tip deflection constraint

Being z_E the value of the vertical coordinate of point E, midpoint between A and B, for the unloaded wing and $z_{E'}$ the value in the deformed one, the difference between $z_{E'}$ and z_E corresponds to the value of δ_{tip} (Equation (3.23)).

$$\delta_{tip} = z_{E'} - z_E \quad (3.23)$$

The value of the deflection constraint (C1) is then assessed considering the minimum and maximum values for tip deflection, $\delta_{tip,min}$ and $\delta_{tip,max}$. If δ_{tip} is inside of the interval $[\delta_{tip,min}, \delta_{tip,max}]$ no penalty is applied (C1 = 0), otherwise C1 is defined by the ratio between δ_{tip} and the limit that is closer to it: $\delta_{tip,min}$ or $\delta_{tip,max}$ (Equation (3.24)).

$$C1 = \begin{cases} 0, & \delta_{tip,min} \geq \delta_{tip} \leq \delta_{tip,max} \\ \frac{\delta_{tip}}{\delta_{tip,min}}, & \delta_{tip} < \delta_{tip,min} \\ \frac{\delta_{tip}}{\delta_{tip,max}}, & \delta_{tip} > \delta_{tip,max} \end{cases} \tag{3.24}$$

To define the bounds of the deflection constraint, $\delta_{tip,min}$ and/or $\delta_{tip,max}$, the deflection of a cantilever beam under a constant distributed load of magnitude ω is used as reference (Figure 3.14). Equation (3.25) [32] defines the value of the vertical displacement (δ) as a function of the position along the beam (y). In this equation, *EI* is the flexural stiffness, *L* is the beam's length, and the sign depends on whether the load pushes the beam upward (+) or downward (-).

$$\delta(y) = \pm\frac{\omega}{24EI}(6L^2y^2 - 4Ly^3 + y^4) \tag{3.25}$$

The value of the displacement along the beam can also be computed using Equation (3.26), in which $\delta_{tip,beam}$ is the result of applying Equation (3.25) to $y = L$, η is the ratio between the position value (y) and the beam's length (L), and $\delta^*_{tip,beam}$ is an assumed value for the beam's deflection.

$$\delta(y) = \frac{\delta(y)}{\delta_{tip,beam}}\delta^*_{tip,beam} \Leftrightarrow \delta(\eta) = \pm\frac{1}{3}(6\eta^2 - 4\eta^3 + \eta^4)\delta^*_{tip,beam} \tag{3.26}$$

This approach reveals useful in this work, since it helps setting the bounds for the vertical displacement of the structure ($\delta_{tip,min}, \delta_{tip,max}$) considering that only one panel will be studied. The latter presented expression allows to compute the vertical displacement expected at a given position, provided the ratio parameter, and the deflection at the wing tip. In early design phases, it is common to assume that the vertical displacement of the wing tip should not exceed ten percent of the wing's semi-span value.

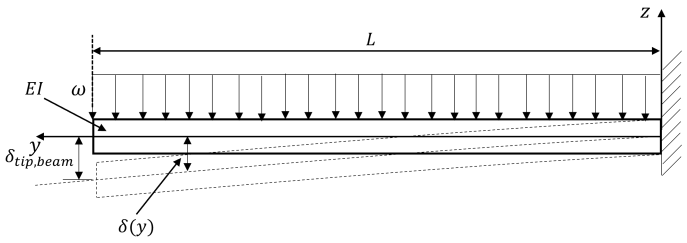


Figure 3.14: Cantilever beam under constant distributed load

3.2.1.2 Twist (C2)

During flight, the aerodynamic loads applied on the wing lead to changes in its shape, affecting the performance of the aircraft. In order to avoid stall or considerable aerodynamic losses in the outer sections of the wing, a limitation to the tip twist is applied to prevent an excessive rotation.

The tip twist angle (θ_{tip}) is then obtained through the dot product between the vectors $\overrightarrow{EF}$ and $\overrightarrow{E'F'}$ (Equation (3.27)). To compute the referred vectors one uses the difference between the *x* and *z*-coordinates of the points E and F in the deformed and original structures. Point F is the midpoint between C and D, that are the intersections of the rear web with the shell.

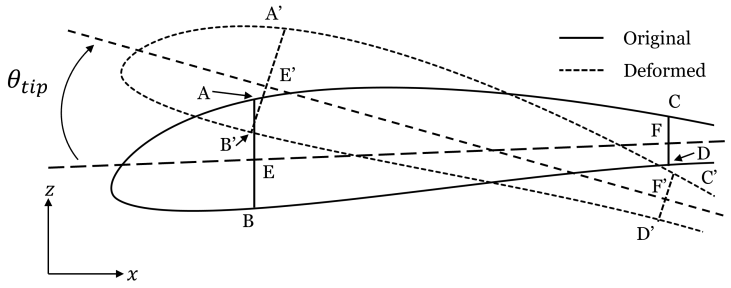


Figure 3.15: Scheme of tip twist constraint

$$\theta_{tip} = \arccos\left(\frac{\overrightarrow{EF}\cdot\overrightarrow{E'F'}}{\|\overrightarrow{EF}\|\|\overrightarrow{E'F'}\|}\right)\frac{180^\circ}{\pi} \quad (3.27)$$

The C2 constraint value is calculated in similar way to the deflection constraint (Equation (3.28)). If θ_{tip} is inside of the desired range, $[\theta_{tip,min}, \theta_{tip,max}]$, then C2 = 0, on the other hand C2 is equal to the ratio between θ_{tip} and the closer limit, either $\theta_{tip,min}$ or $\theta_{tip,max}$.

$$C2 = \begin{cases} 0, & \theta_{tip_{min}} \geq \theta_{tip} \leq \theta_{tip_{max}} \\ \frac{\theta_{tip}}{\theta_{tip_{min}}}, & \theta_{tip} < \theta_{tip_{min}} \\ \frac{\theta_{tip}}{\theta_{tip_{max}}}, & \theta_{tip} > \theta_{tip_{max}} \end{cases} \quad (3.28)$$

The calculation of the bounds values used in Equation (3.28), $\theta_{tip,min}, \theta_{tip,max}$, follows a similar approach to the one presented for the deflection constraint. According to the analysis of single cell beam structures subjected to a torsion moment (T), the twist varies linearly along the beam's length (dy) (Equation (3.29) [32]).

$$\frac{d\theta}{dy} = \frac{T}{4A^2} \oint \frac{ds}{Gt} \quad (3.29)$$

Therefore, if one assumes a value for the twist at the wing tip (θ^*_{tip}), the expected value of twist for any position in the semi span ($\theta\eta$) is given by Equation (3.30), where η corresponds to the ratio between the spanwise position and the semi-span value.

$$\theta(\eta) = \eta\theta^*_{tip} \quad (3.30)$$

3.2.1.3 Failure assessment of composites (C3)

The failure analysis of composite structures uses the output values of the failure theory from MYSTRAN, each composite structure is considered separately and the analysis is performed layer-by-layer. Equation (3.31) is the definition of the C3 constraint. Failure occurs if the overall value of the failure theory of the wing, $FT_{overall}$, equals or exceeds the reference value for the failure theory selected, $FT_{failure}$, meaning C3 = 1.

$$C3 = \begin{cases} 0, & FT_{overall} < FT_{failure} \\ 1, & FT_{overall} \geq FT_{failure} \end{cases} \quad (3.31)$$

Due to the way the loads are transferred to the structure and the elements used, it was clear, during the development of the tool that local failure was often found in a small region close to the caps. Bearing this in mind, the evaluation of the overall value for the failure theory, $FT_{overall}$, uses a smoothing technique to avoid sizing the structure based only on the failure of one element. Considering a layer of a structure, Figure 3.16, in which at least one triangle presents a failure index greater or equal to the failure theory reference value, the smoothing process centers a sphere, of radius *r*, in the one with higher value (dark grey triangle) and identifies the triangles which centers are in the vicinity region defined by the sphere (grey triangles). The value of the radius is defined by the user and is mostly based on an empiric analysis of the problem.

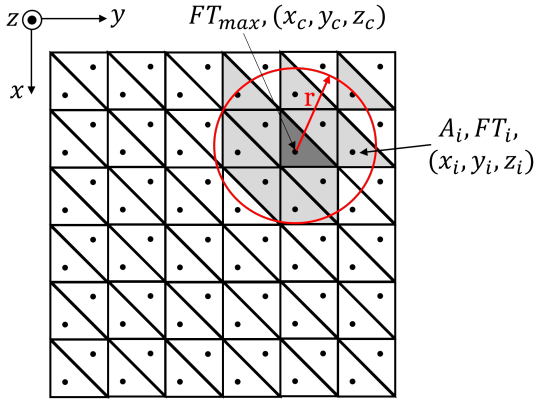


Figure 3.16: Example of composite failure analysis

Using Equation (3.32) it is possible to assess if there is failure in the structure and return this value to the constraint assessment (Equation (3.31)). This analysis considers that if one layer fails, the structure also fails, hence the solution is penalized. In Equation (3.32), A_i is the area of each triangle inside the sphere and FT_i is the respective failure theory index.

$$FT_{overall} = \frac{\sum_{i=1}^{n_{tri}} A_i FT_i}{\sum_{i=1}^{n_{tri}} A_i} \quad (3.32)$$

In this thesis, the Tsai-Hill failure criteria is used to assess the occurrence of failure in composite structures, therefore the reference value, $FT_{failure}$, is considered to be 1.0 (Equation (2.5)).

3.2.1.4 Failure assessment of caps (C4)

To check if there is failure in the caps structures, the values of the safety margin are used. According to the definition of safety margin, the failure of a structure occurs when a value of zero or lower is reached, thus the C4 constraint assumes value 1 when the minimum safety margin in the caps are less than or equal to zero. When there is no failure, $SM_{min} > 0$, no penalty is applied by considering C4 equal to zero. Equation (3.33) is the mathematical definition of the C4 constraint.

$$C4 = \begin{cases} 0, & SM_{min} > 0 \\ 1, & SM_{min} \leq 0 \end{cases} \quad (3.33)$$

The safety margin is defined according to Equation (3.14).

3.2.2 Objective function

The objective function is the formula that represents the fitness of a certain individual in the population. In this case, the objective is divided into two main terms: the mass and the penalties. For the general case, the wing mass (m) is given by the sum of the n_{wp} panels masses, and each panel is composed of a shell, n_{web} webs and n_{cap} caps (Equation (3.34)).

$$m = \sum_{j=1}^{n_{wp}} \left(m_{shell_j} + \sum_{k=1}^{n_{web}} m_{web_{jk}} + \sum_{k=1}^{n_{cap}} m_{cap_{jk}} \right) \quad (3.34)$$

The shell's mass of the *j*-th panel is defined by Equation (3.35) and depends on its stacking sequence mass (γ_{shell_j}) and on the surface area of the panel ($\bar{c}_{panel_j} \cdot b_{panel} \cdot p^*_{airfoil_j}$). The surface area is computed using the panel's mean chord ($\bar{c}_{panel_j}$), the panel length (b_{panel_j}) and the average of the non-dimensional airfoils' perimeters ($p^*_{airfoil_j}$).

$$m_{shell_j} = \bar{c}_{panel_j} \cdot b_{panel_j} \cdot p^*_{airfoil_j} \cdot \gamma_{shell_j} \quad (3.35)$$

Similarly to shells, the mass of the webs are computed using the mass per unit area of the correspondent stacking sequence (γ_{web}). In Equation (3.36), $\bar{h}_{web,jk}$ refers to the web's height in milimeters.

$$m_{web_{jk}} = \bar{h}_{web,jk} \cdot b_{panel_j} \cdot \gamma_{web,jk} \quad (3.36)$$

The mass of the *k*-th cap of panel *j* is the result of the product between the volume of the cap and the density of the material used ($\rho_{cap,jk}$) (Equation (3.37)). The volume of the cap depends on the cap's width ($w_{cap,jk}$) and thickness ($t_{cap,jk}$), and on the panel length ($b_{panel,j}$).

$$m_{cap,jk} = w_{cap,jk} \cdot t_{cap,jk} \cdot b_{panel,j} \cdot \rho_{cap,jk} \quad (3.37)$$

According to the definition presented by Smith and Coit [17] in the previous chapter, an exterior static penalty method is composed of a penalty constant, a distance metric, and a penalty exponent to reduce the fitness of non-feasible individuals. Both geometric constraints are continuous since there is a relation between the bound and the actual value of the parameter. The failure constraints are classified as discrete since they can only assume the value 0 or 1. Regarding the continuous constraints, one should note that the solution is more penalized as the distance between the value of the parameter and the bound increases. The penalty term of the objective function (Equation (3.38)) is the result of the product between the penalty constant (PC) and the sum of the constraints (C_l) powered to the penalty exponent (PE).

$$p = PC \sum_{l=1}^{4} (C_l)^{PE} \quad (3.38)$$

Equation (3.39) is the objective function (*of*) for this problem and its value results of the sum of the wing mass and the penalty term described above.

$$of = m + p = \sum_{j=1}^{n_{wp}} \left(m_{shell_j} + \sum_{k=1}^{n_{web}} m_{web_{jk}} + \sum_{k=1}^{n_{cap}} m_{cap_{jk}} \right) + PC \sum_{l=1}^{4} (C_l)^{PE} \quad (3.39)$$

The design variables (DV) defined for the problem are related with the stacking sequence of each laminated structure (shell or web) and with the number of layers on each cap. For one panel, one can have a maximum of seven design variables that correspond to: stacking sequence of shell, main spar web, rear spar web, and number of layers in each cap, if four caps are activated. For the laminated structures, the design variable value corresponds to the ID of the SST line that must be used, and its range, by default, starts at one and goes up to the number of lines of the SST. As already referred, for the caps the design variable corresponds to the number of layers that should be applied, and a maximum value must be set. In both cases, all design variables are positive integers. When a design variable assumes a value that is outside the interval, ($DV_i > DV_{i,max}$) the optimizer automatically considers that the objective function is infinite. This issue is motivated by the use of a binary encoded system, and appears when the number of bits needed to represent the upper bound of the design variable range allows the representation of numbers above the limit.

The aim of this study is to minimize the wing mass ensuring that the solutions comply with deformation and failure constraints, resulting in the following formulation of the optimization problem:

$$\begin{aligned}
&\text{minimize} && \sum_{j=1}^{n_{wp}} \left(m_{shell_j} + \sum_{k=1}^{n_{web}} m_{web_{jk}} + \sum_{k=1}^{n_{cap}} m_{cap_{jk}} \right) + PC \sum_{l=1}^{4} (C_l)^{PE} \\
&\text{with respect to} && DV_i,\ i = 1, \ldots, ndv; \\
&\text{subject to} && DV_i \leq DV_{i,max}
\end{aligned}$$

In the fomulation of the optimization problem, *ndv* corresponds to “number of design variables”.

3.3 Implementation

3.3.1 Structural optimization tool layout

As observed in Chapter 2, coupling an evolutionary algorithm with a finite element analysis software can be used to perform the optimization of composite structures. Figure 3.17 presents the structure of the program considering that it was launched with *n* processes in parallel, named workers, for a problem with *m* design variables. The tool is split into two major components: "StructOpt - StandAlone" (Subsection 3.3.2) and the optimizer, that includes the genetic algorithm driver, and the objective function (Subsection 3.2).

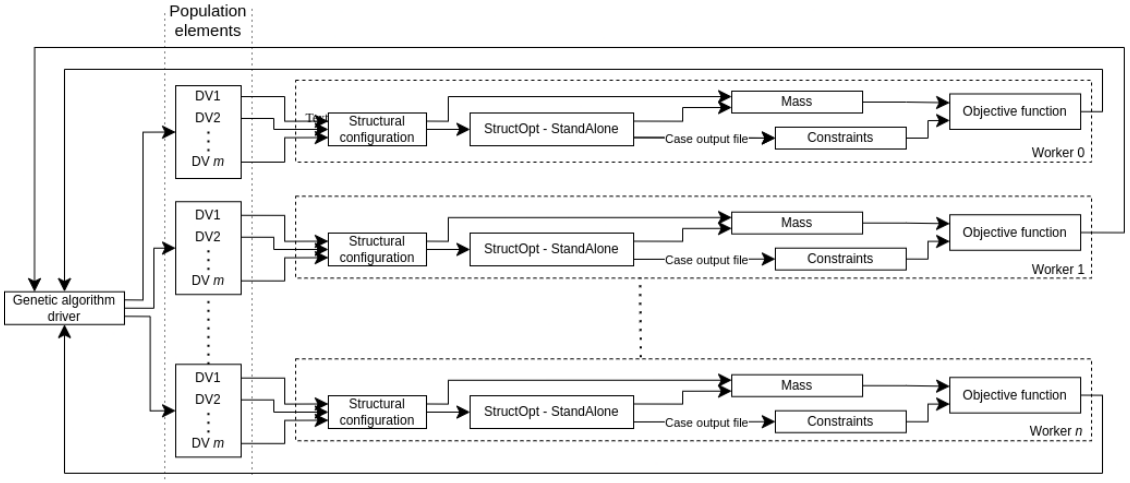


Figure 3.17: Optimization tool layout

Each worker includes all the tools necessary to analyze a structural configuration and fitness of a given wing, acting like a blackbox for the optimization driver. The communication with the genetic algorithm is done at two points: at the conversion of the design variables of a population element into a standard file ("Structural configuration") and at the feedback of the fitness of an individual ("Objective function"). The "StructOpt - StandAlone" comprises the BDF generator, and the FEA solver. This block was designed to also be used outside of the optimization tool to perform structural analyses of wings. The constraints are evaluated using the results of the structural analysis, namely the failure theory indices, the safety margin values, and the deformed shape. The objective function is the result of the sum of the contraints with the wing mass value.

The optimization process can be performed either in series or parallel, however using each computer processor as a worker in parallel reduces the execution time since several elements in the population are evaluated concurrently. More details about the implementation and execution of this tool can be found in Appendix A.

3.3.2 StructOpt - StandAlone

The “StructOpt - StandAlone” is a sub-component of each “worker” and is divided into two main parts: the BDF generator and the FEA solver (MYSTRAN). In Figure 3.18 it is possible to observe how the sub-component is internally divided. This program can be used without the optimization driver, providing an easy way to perform static structural analysis of different wings (“StructOpt - StandAlone_Solo”). The version used as part of the optimization is named “StructOpt - StandAlone_Opt”.

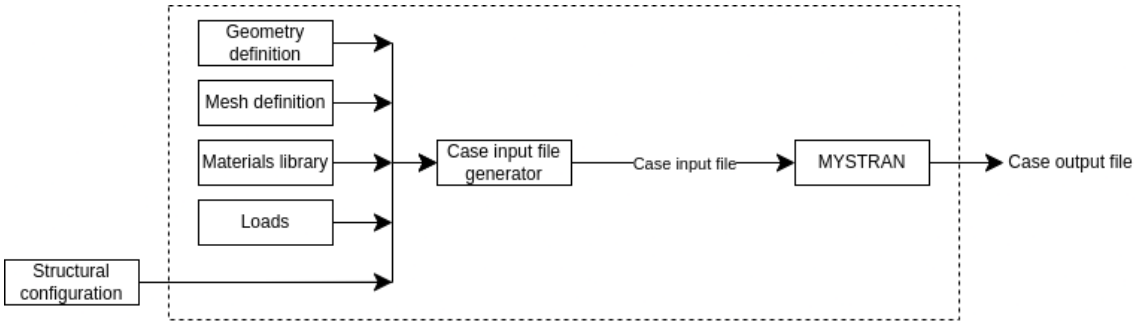


Figure 3.18: Block diagram of “StructOpt - StandAlone”

Since the input for the structural solver is a BDF (Figure 3.19), developing a program able of generating this type of document is a need. A bulk data file is divided into three mandatory sections: the executive section, the case-control, and the bulk data part. The analysis’ type and settings of each subcase are defined in the first two sections and the information about geometry, materials and loads composes the third one.

According to the MYSTRAN users’ reference manual [30], a static structural analysis is defined using “SOL 1” in the executive section, followed by “CEND” to close this part and proceed to the definition of the subcases in the case control section. Each subcase represents a different structural analysis, which is very useful if one wants to analyze a given structure under different operating conditions. An operating condition can be represented by one or more “LOAD” entry/entries, and the outputs in terms of stresses and displacements are also defined at this stage. The third section definition starts with the “BEGIN BULK” statement and contains the entries related to grid points, elements, materials, and loads. Some parameters related to the solver execution can also be defined in this section before the “ENDDATA” statement.

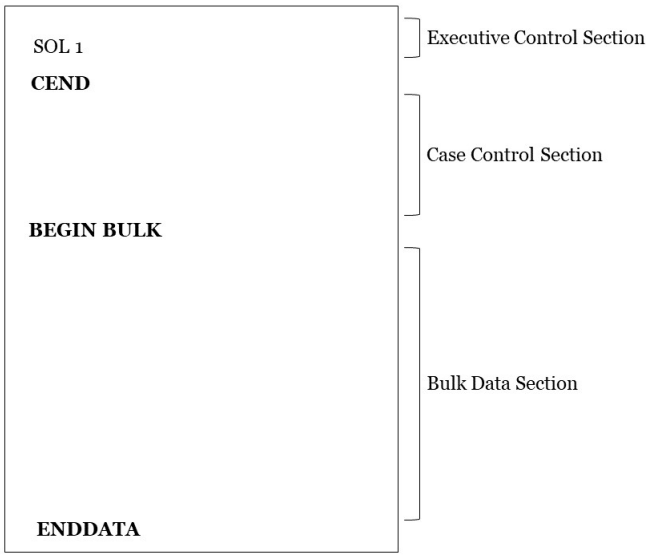


Figure 3.19: Structure of a bulk data file

To define a node, a “GRID” entry must be used (Table 3.5). In this entry, “X1”, “X2”, and “X3” correspond to the X, Y, and Z coordinates of the node, respectively, “GID” is the node ID determined when numbering the nodes, and “PSPC” is the permanent single point constraint. This last setting is where the boundary conditions of the problem are defined, being a sequence of integers corresponding to the degrees of freedom that are restricted. The definiton of each node is performed in the global reference frame of the problem, which is translated by assigning the value zero in the third column.

Table 3.5: Example of entry for grid points (GRID) (adapted from [30])

GRID	GID	0	X1	X2	X3	0	PSPC		

The triangular element definition, Table 3.6, requires the input of information related with the element’s ID (“EID”), material’s property (“PID”), and the sequence of grids that compose the element (“G1-G3”). In this work, the material’s property for these elements is a “PCOMP” entry (see Table 3.12), representing a layered structure. The order that the grids are declared in this entry must be in accordance with the local numbering presented in Figure 3.6.

Table 3.6: Example of entry for triangular elements (CTRIA3) (adapted from [30])

CTRIA3	EID	PID	G1	G2	G3	0	0		

Bar elements are used to define the spar caps. The defnition of these elements require the assignment of two nodes (“GA” and “GB”), the element’s ID (“EID”), as well as the declaration of a mechanical property, “PBARL” (PID, see Table 3.10), and the components of a vetor that define the vertical plane along the element (“V1-V3”).

Table 3.7: Example of entry for bar elements (CBAR) (adapted from [30])

CBAR	EID	PID	GA	GB	V1	V2	V3	

Equation (3.22) gives the final result for the forces applied in the structure. Each one of these forces is converted into a "FORCE" entry for the BDF file (Table 3.8. In this entry, "FID" is the force's identification number, "GID" is the node ID at which the force is applied, "F" the magnitude of the force in a specific direction, and "N1-N3" is the orientation of the direction (Equation (3.40)).

$$[N] = \begin{cases} [1,0,0], \text{for forces in X direction} \\ [0,1,0], \text{for forces in Y direction} \\ [0,0,1], \text{for forces in Z direction} \end{cases} \tag{3.40}$$

Since all the forces are calculated in the global frame of reference, the value of column four is zero.

Table 3.8: Example of entry for forces (FORCE) (adapted from [30])

FORCE	FID	GID	0	F	N1	N2	N3		

For the structural solver, a "LOAD" entry represents a group of forces. This feature allows the definition of different operating points from the structural solver point of view, since all the forces can be condensed in one entry. In Table 3.9, "LID" is the load identification number, and "S" is a global scaling factor. For each force that is part of the loading under definition, an individual scaling factor ("Si") and the force ID ("Fi") should be declared. Since the magnitude of the computed forces is absolute, all scaling factors, either global or individual, are one.

Table 3.9: Example of entry for loads (LOAD) (adapted from [30])

LOAD	LID	S	S1	F1	S2	F2	S3	F3	+CONT
+CONT	S4	F4	(etc)						

As previously referred, each cap is a series of bar elements and a property that defines its mechanical behaviour, "PBARL", must be assigned to it. As one observes in Figure 3.9, a cap has a rectangular cross-section (BAR) with thickness t_{cap} and width w_{cap}. The typical bar element property entry is shown in Table 3.10. For this definition, a unique identification (PID) must be provided to the attribute, as well as a linear isotropic material (MID). The remaining items are concerned with the cross-section shape and size. Considering that the caps are intended to carry the bending loads and a high ratio between the total spar cap length and its cross-section area is expected, linear isotropic materials (MAT1) are adopted to define the mechanical properties of the unidirectional fabrics that compose the caps.

Table 3.10: Example of entry for bar element property (PBARL) (adapted from [30])

PBARL	PID	MID		BAR					+CONT1
+CONT1	w_{cap}	t_{cap}							

According to Table 3.11, one must provide a unique material identifier (MID) and at least two of the following three parameters: Young's modulus (E), shear modulus (G) and Poisson's ratio (ν). The material's stresses limits, that are used to compute the safety margins, must be given in the second line of the entry, with "TA" representing the tensile limit, and "CA" and "SA" defining the compression and shear limits respectively.

Table 3.11: Example of entry for linear isotropic materials (MAT1) (adapted from [30])

MAT1	MID	E	G	ν	0	0	0	0	+CONT
+CONT	TA	CA	SA						

Figure 3.10 is an example of a general layered composite structure that composes the shell and the spar webs. Table 3.12 represents the structure of a composite material entry, "PCOMP". Each property is identified through a unique ID (PID) and this value is used in the definition of the plate elements. "Zo" corresponds to the distance between the grid points surface and the midplane of the composite structure. Since an offset is applied to get the grid points of the shell and the mid-planes of the webs are used, this value is zero. "FT" corresponds to the definition of the failure theory used to assess the structure and four options are available in MYSTRAN: Tsai-Hill (HILL), Hoffman Criterion (HOFF), Tsai-Wu (TSAI) and strain based (STRN). The lamination scheme type is defined through the "LAM" parameter. In the case of symmetrical structures, this setting must be defined as "SYM" and just half of the sequence needs to be detailed. For non-symmetric laminates, the full sequence must be specified and the "LAM" parameter should be "NONSYM". After the continuation entries, begins the definition of the stacking sequence in terms of layers starting from the top one.

Table 3.12: Example of entry for PCOMP property (adapted from [30])

PCOMP	PID	Zo	0	0	FT	0	0	LAM	+CONT1
+CONT1	MID1	t_{l1}	θ_{l1}	NO	MID2	t_{l2}	θ_{l2}	NO	+CONT2
+CONT2	MID3	(etc)							

A layer is defined by its material (MID), thickness (t_l) and orientation relative to the element material axes (θ_l). Only linear orthotropic materials can be used as MID and its definition is shown in Table 3.13. Similarly to the definition of linear isotropic materials, in the first line of the "MAT8" entry one must input the data relative to the elastic properties (E1, E2, $\nu_1 2$, G12) and the values associated to the materials' strengths (Xt, Xc, Yt, Yc) must be defined in the second line. The "F12" and "STRN" parameters are related to the failure analysis when using the Tsai-Wu and strain methods, respectively.

Table 3.13: Example of entry for linear orthotropic materials (MAT8) (adapted from [30])

MAT8	MID	E1	E2	ν_{12}	G12	0	0	0	+CONT1
+CONT1	0	0	0	Xt	Xc	Yt	Yc	S	+CONT2
+CONT2	0	F12	STRN						

The user must classify each material either as core or reinforcement fiber, specify the layer's thickness and orientations that it can assume and assign it to a structure or group of structures in accordance with the definition presented in Table 3.14.

Table 3.14: Assignments definition

Value	Assignment
1	Shell
2	Main spar web
3	Rear spar web
4	Shell & main spar web
5	Shell & rear spar web
6	Main & rear spar web
7	All

The program has seven text input files where the definition of the geometry, mesh, materials, loads and structural configuration is contained. Figure 3.20 is a block diagram of the implemented bulk data file generator, where one can observe the flow of information accross the program. The main processing stations of the program are explained in the previous sections, providing more details on the methods used to convert the inputs into the bulk data file.

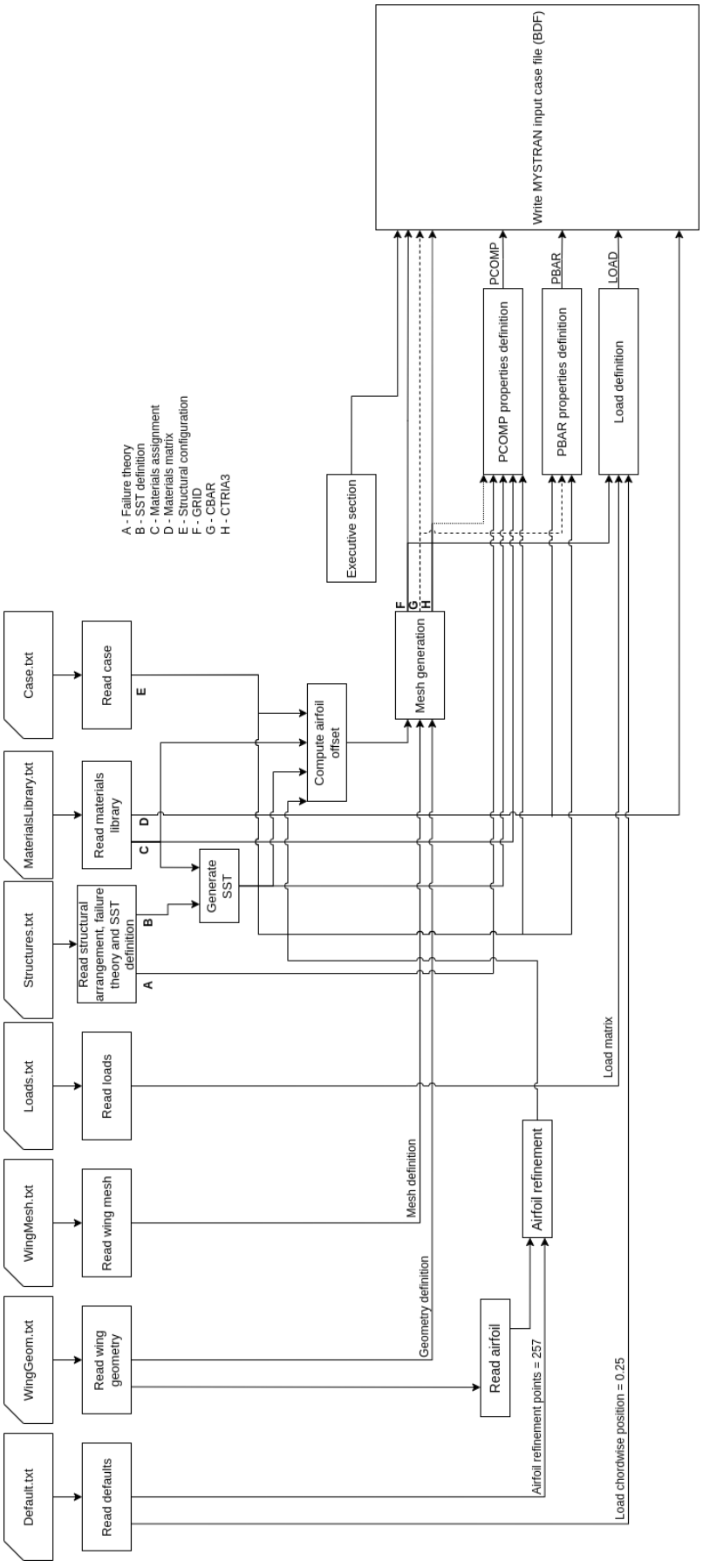


Figure 3.20: Block diagram of bulk data file generator

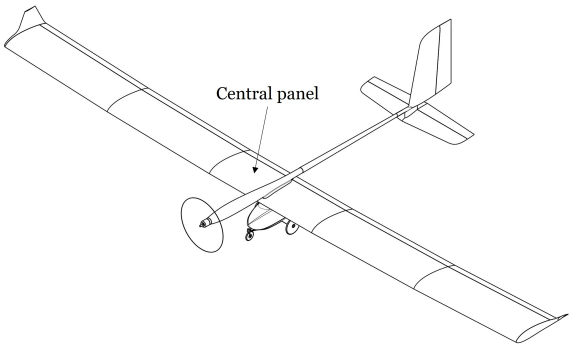
Chapter 4

Results

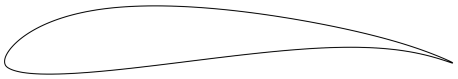
In this chapter, the ACC2019 aircraft central wing panel is optimized, showing an application of the optimization tool. Initially, the wing geometry, loading conditions and materials selected are presented, followed by the mesh convergence study. A brief analysis of the population size, number of generations and mutation rate is performed to assess which set of parameters better suit the optimization problem. At the end of the chapter, the optimized configuration is discussed.

4.1 Wing geometry and baseline design

For the Air Cargo Challenge 2019, the AERO@UBI team designed an aircraft that uses a five panel rectangular wing. In order to ease the manufacturing process and to fit all parts inside the transportation box, all panels have the same length, 745 mm, and a 300 mm chord [33]. The wing also includes a flaperon across the whole span with a chord of 17%. Figure 4.1a is an isometric view of the aircraft and Figure 4.1b is the selected airfoil, NewACC2017, which presents a maximum thickness-to-chord ratio of 12.95% around the quarter chord. The cross-section is a two-cell beam structure with sandwich skin, spar webs, and laminated spar caps similar to the one shown in the previous chapter. The maximum thickness-to-chord ratio location of the airfoil combined with the desired center-of-gravity position defined the location of the main spar at twenty-five percent (25%), and the flaperon chord positions the rear spar web position at eighty percent (80%) of the chord.



(a) Isometric view of the ACC 2019 aircraft



(b) NewACC2017 airfoil

Figure 4.1: Isometric view and wing airfoil of the ACC 2019 aircraft

The sizing procedure took an analytical approach, with bending loads being supported only by the spar caps and shear and torsion loads carried by the shell and webs.

Considering that some unpredictable loads can occur during the flights' stages, both caps were designed with an equal number of layers. For the shell and webs, the core material was selected to ensure the stability of skins and enough stiffness during handling and manufacturing, with its contribution to carrying part of the stresses being disregarded. This process also did not take into account any information on tip twist and deflection, focusing only on avoiding structural failure.

The central portion of the wing is the one that is loaded the most, which makes the central panel the heaviest of the five. From the sizing process of this part, it was found that each cap should have an area of 30 mm^2, being chosen a 20 mm width by 1.5 mm thick cap. An 80 g/m^2 unidirectional carbon fiber fabric was chosen to laminate the caps, and since each layer is 0.15 mm, 10 layers are required to achieve the referred area. To withstand the shear forces at the main spar web, two layers of bidirectional 30 g/m^2 carbon fiber should be added to each side of the core, a 10 mm thick AIREX C70.55. To close the second cell, a 5 mm thick AIREX C70.75 foam is used as rear web. The shell is a sandwich structure of 1.5 mm balsa core with two layers of bidirectional 30 g/m^2 carbon fiber added as reinforcement on each side, with a total thickness of 1.82 mm.

Table 4.1 presents the values for the masses of each part of the central panel. The shell and spar caps together represent more than 90% of the panel's mass, and given the importance of these parts in the structure's behavior, using the shell's stacking sequence and the number of layers of each cap as design variables could lead to an improvement on the panel's mass. The case study considered for the optimization, uses the central panel of the wing, and due to the computational cost of the structural analysis, the leading edge part is removed. As a result, the structure under analysis is a composite wingbox composed of a shell, a main and rear spar webs, and two spar caps (Figure 4.2).

Table 4.1: Masses of central panel's parts

Part	Mass [g]	Percentage (%)
Shell	227.2	69.57
Spar caps	72.0	22.05
Main web	25.2	7.71
Rear web	2.2	0.67
Total	326.5	-

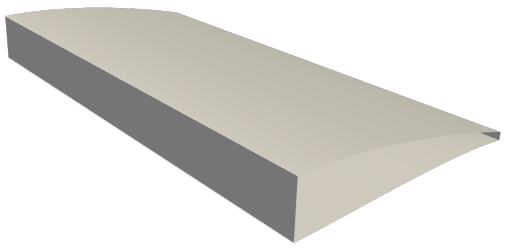


Figure 4.2: Wingbox structure used in case study

4.2 Materials and SSTs

Considering the baseline design presented in the previous section, it is necessary to detail the materials, their mechanical properties, and the stacking sequence tables that are generated in order to optimize the wing structure. As already referred, two technical foams (AIREX C70.55 and AIREX C70.75) and balsa wood were selected to compose

the spars' webs and the shell, respectively. The main spar web and the shell core were reinforced by applying two layers of a low aeral weight carbon fabric (CW 30) on each side. Since the structural analysis does not predict instability, and considering the satisfactory result obtained in the structures manufactured for the ACC 2019 aircraft, it was decided that the same core materials and assignments should be used in the optimization process. Regarding the orientations allowed for this materials, presented in Table 4.4, a zero degree angle with the longitudinal direction of panel was chosen because two of the materials are considered isotropic (AIREX foams), therefore varying the orientation should not change the results, and due to manufacturing details for the case of balsa wood. The commercially available balsa sheets, are usually 100 mm x 1000 mm, with the fibers aligned with the longer side therefore orienting the length of the balsa sheet with the longitudinal direction of the panel is interesting considering the ammount of material that needs to be bought to manufacture the structure. Regarding the reinforcement skins, four orientations are available for the bidirectional carbon fabric (0º, 15º, 30º and 45º), separated by fifteen degrees due to manufacturing constraints. The material considered for the spar caps is a 80g/m^2 unidirectional fabric of carbon fiber with a thickness of 0.15 mm per layer. Tables 4.2 and 4.3 summarize the mechanical properties of the fibers and cores after applying a quality factor of 1.4, and a safety factor of 1.5. Table 4.4 summarizes the assignment of fibers and cores to composite structures. For the shell and main spar web, four combinations of material-angle are considered and only one core, while for the rear spar web only one core is assigned. To decode the assignment of each material, one should refer to Table 3.14, presented in the Methodology chapter.

A stacking sequence table of a structure is generated using all the material-orientation combinations available for a structure, and its size also depends on the maximum number of layers allowed for a laminate, as presented in Equation (3.15). At this point it is important to set the number of layers that can be used to define the half sequence of each laminated structure, that later will make possible to limit the design variables. Bearing this in mind, and provided the definition of the materials and assignments presented above, a maximum of five layers per side are allowed for the shell and main spar web structures, while only one layer is considered for the rear spar web. This definition results on having two stacking sequence tables with 340 lines for the shell and main spar (SST_Shell and SST_MS) and a stacking sequence table with just one line for the rear spar (SST_RS). The stacking sequences used in the ACC 2019 aircraft are shown in Table 4.6, and the "SST line" values are the input for the definition of the wing structure in the mesh convergence study that follows. One should also refer that just half of the sequence is presented considering that it is a symmetrical laminate. Table 4.5 presents the correspondence between a material-orientation arrangement and the respective value that appears in a stacking sequence table.

Table 4.2: Spar caps material (MAT1)

Material	E [MPa]	G [MPa]	ν	T [MPa]	C [MPa]	S [MPa]	Thickness [mm]	$\rho_{mat.}$ [kg/m^3]	Ref.
Carbon fiber unidirectional	102900	4.9	0.27	1064	805	35.47	0.15	1600	[34]

Table 4.3: Cores and fibers materials (MAT8)

Material	Type	E1 [MPa]	E2 [MPa]	G_{12} [MPa]	ν_{12}	Xt [MPa]	Xc [MPa]	Yt [MPa]	Yc [MPa]	S [MPa]	Thickness [mm]	Mass	Ref.
Balsa wood	Core	2650.0	39.75	98.05	0.49	48.67	6.0	0.67	0.67	0.67	0.75	150 [kg/m^3]	[34]
AIREX C70.55 (yellow)	Core	45.0	45.0	22.0	0.02	0.87	0.60	0.87	0.60	0.57	5.00	60 [kg/m^3]	[35]
AIREX C70.75 (green)	Core	66.0	66.0	30.0	0.10	1.33	0.97	1.33	0.97	0.80	2.50	80 [kg/m^3]	[35]
Carbon fiber (bidirectional)	Fiber	53900	52500	4550	0.06	449.4	420	399.47	420	33.13	0.08	66 [g/m^2]	[34]

Table 4.4: Assignment table of fibers and cores for shell, main spar web and rear spar web

Material	Assignment	Orientation [°]
AIREX C70.75 (green)	3	0
AIREX C70.55 (yellow)	2	0
Balsa wood	1	0
Carbon fiber (bidirectional) - CW30	4	0, 15, 30, 45

Table 4.5: Correspondence between materials ID and stacking sequence table ID

SST ID	Material	Orientation [°]	SST
1	Balsa wood	0.0	Shell
1	AIREX C70.55	0.0	Main web (MW)
1	AIREX C70.75	0.0	Rear web (RW)
2	CW30	0.0	Shell, MW
3	CW30	15.0	Shell, MW
4	CW30	30.0	Shell, MW
5	CW30	45.0	Shell, MW

Table 4.6: Stacking sequences for the 2019 aircraft

	Layers					
SST line	L1	L2	L3	$\gamma_{structure}$ [g/m^2]	t [mm]	Structure
1	1	-	-	400.00	5.00	Rear web
20	1	5	5	864.00	10.32	Main web
20	1	5	5	489.00	1.82	Shell

4.3 Loads

To compare the solutions of the optimization with the baseline design, similar loading conditions are applied. The analytical approach considered a load factor of 3 for a fully loaded aircraft (m = 12 kg) flying at 28.5 m/s for a sea-level flight. To predict the distributions of lift (C_l), drag (C_d) and pitching moment (C_m) coefficients over half span of the wing, the lifting line theory was used, and the results can be observed in Figure 4.3. The airfoil data used as input for the lifting line theory software was calculated using XFLR5. The lifting line theory results are used as input in the loads file together with a static portion equivalent to the loads acting on the tip and intermediate panels, since the analysis focuses only the central panel. Equation (4.1) presents the values found for the equivalent loading to be applied at the tip of the central panel due to the intermediate and tip panels. One can compute these values by converting the lift and drag at each position into forces acting in the X and Z- direction, using the value of the local angle of attack and the rotation matrix presented in Equation (3.18), and integrating these forces and the moments produced, with respect to the central panel tip position, along the tip and intermediate panels. Figure 4.4 shows the final load applied in the wing central panel that is considered during the optimization.

$$\begin{cases} F_x = 2.7 & \text{[N]} \\ F_z = 140.4 & \text{[N]} \\ M_x = 97481.2 & \text{[Nmm]} \\ M_y = -13258.8 & \text{[Nmm]} \end{cases} \tag{4.1}$$

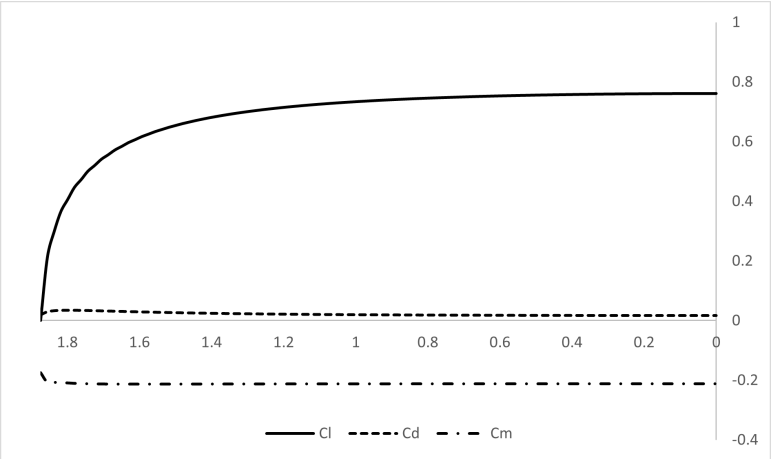


Figure 4.3: Distributions of C_l, C_d and C_m for the case study (n = 3, m = 12 kg, v = 28.5 m/s, sea-level flight)

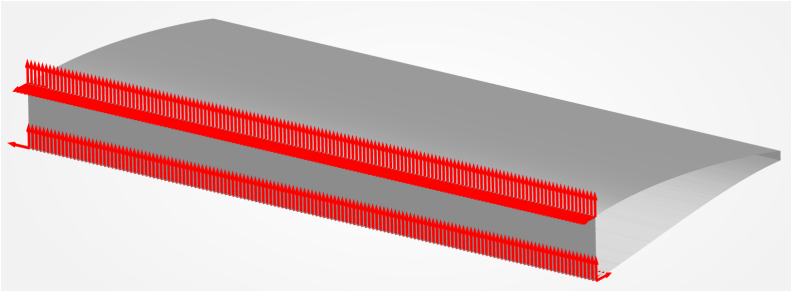


Figure 4.4: Loading conditions for the case study (obtained via FEX 1.0) (magnitude of forces is not to scale)

4.4 Mesh convergence

A mesh convergence study was performed using the baseline design (Section 4.1) under the loads presented at the end of Section 4.3. Selecting the mesh requires the analysis of some parameters such as the vertical displacement and twist angle of the tip, the aspect ratio of the triangular elements, the stress distributions, and the simulation time. Table 4.7 presents the definition of each tested mesh, the values obtained for the deformation parameters (δ_{tip} and θ_{tip}), and the time needed to solve each problem.

The number of divisions in each structure was chosen considering the aspect ratio of the triangular elements. An effort was made to keep the aspect ratio of the triangular elements between 1 and 2, even though it was not always possible, particularly for the webs given the differences between the webs' heights and the panel span. This is a direct consequence of the way the mesh is generated, since the number of divisions along the span is the same for all the structures. Regarding the wing's deformation values, δ_{tip} and θ_{tip}, one can affirm that no considerable changes occur with the increase of the refinement level of the mesh. The time required to solve the problem is important since it sets the global computational cost of the optimization. Selecting the mesh must be a compromise between the number of elements needed to have an acceptable stress resolution and the amount of time that the optimization will take. If on one hand, tip's displacement and twist are not significantly affected by the refinement level, on the other hand stresses in layers heavily depend on the number of triangular elements used to represent the structure. Figure 4.5 shows the results obtained for the stress in direction 1 for the top layer of the main spar web by meshes 0, 4, 8, and 9. It is clear that the definition of "Mesh 0" is not good enough to capture a realistic stress distribution. In Figure 4.5a it is observable that adjacent triangles do present completely different stress levels, and in the majority of the cases a change in sign occurs. These results are a consequence of using constant strain triangular (CST) elements, meaning that to improve in terms of stress distribution, a higher number of elements is required. In fact, this problem appears to start vanishing as the refinement level of the mesh increases, although it is not

completely solved (Figures 4.5a, 4.5b, 4.5c). The distributions of stress in direction 1 for the top layer of the main spar is used as an example, however this issue generally appears in structures represented by CTRIA3 elements. Comparing Figures 4.5c and 4.5d, one observes that no significant improvements are achieved in terms of stress distribution from "Mesh 8" to "Mesh 9". This conclusion can be withdrawn as well from the analysis of the shell structure, for the upper and lower surfaces. In terms of computation time, the difference between these two meshes is around twelve minutes. Observing the minimum and maximum values of the safety margin for the bar elements (Figure 4.6), it is possible to conclude that the stresses in the spar caps do not vary significantly with the number of divisions in the longitudinal direction. The safety margin values at the spar caps vary between around 5.7 and 10.5. Lower values of this parameter are located in the upper cap close to the root, where the compressive stress is more intense, whereas the higher values come from the lower cap, since the direct stresses in these elements are compared with the tensile strength properties.

Considering the results of this study, the mesh selected to be used in the optimization process is mesh number 8, with a total of 34800 nodes, 69900 elements (300 CBAR, and 69600 CTRIA3), and computation time around twenty one minutes. According to the information provided by the pre-processor "fex-1.0" about the elements' quality, the aspect ratio of the triangular elements from "Mesh 8" are considered to be in an acceptable range.

Table 4.7: Mesh convergence for the case study

	Number of divisions						Number of elements					
Mesh	Longitudinal	MS	RS	LS	US	Number of nodes	CBAR	CTRIA3	Total	δ_{tip} [mm]	θ_{tip} [°]	Time [s]
0	30	5	1	20	20	1380	60	2760	2820	3.86	0.78	8
1	45	5	1	30	30	2970	90	5940	6030	3.87	0.78	19
2	60	6	2	40	40	5280	120	10560	10680	3.88	0.78	46
3	75	13	3	50	50	8700	150	17400	17550	3.89	0.78	104
4	90	16	3	60	60	12510	180	25020	25200	3.89	0.78	194
5	105	19	3	70	70	17010	210	34020	34230	3.89	0.77	335
6	120	21	4	80	80	22200	240	44400	44640	3.90	0.77	542
7	135	24	4	90	90	28080	270	56160	56430	3.90	0.77	844
8	150	27	5	100	100	34800	300	69600	69900	3.90	0.77	1250
9	165	30	6	110	110	42240	330	84480	84810	3.90	0.77	2010

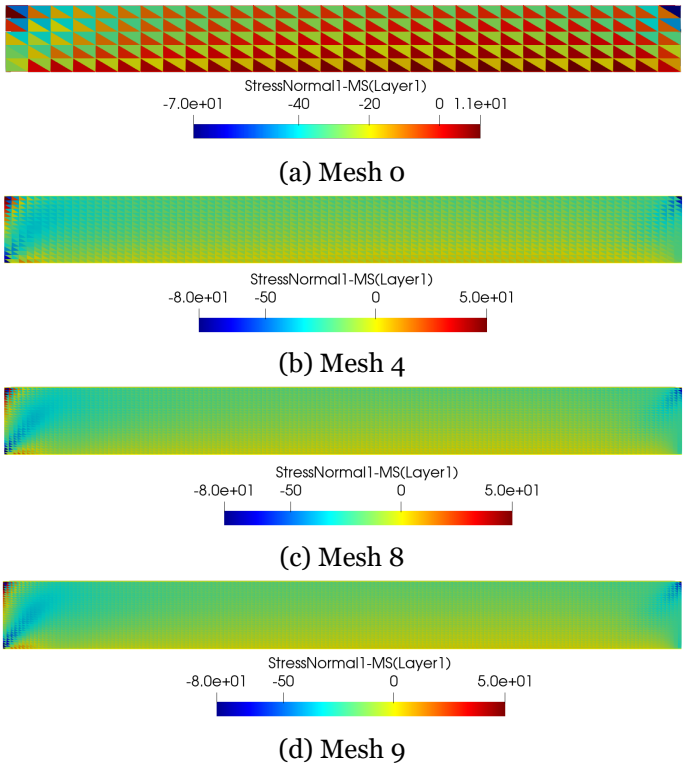


Figure 4.5: Stress [MPa] in direction 1 (Layer 1 - Main spar web)

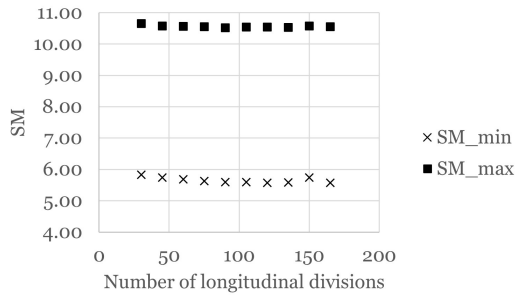


Figure 4.6: Safety margin values versus number of longitudinal divisions

4.5 Optimization

4.5.1 Optimization setup

To setup the optimization, the user must define the values associated with the design variables bounds, the evaluation of penalties, as well as the mutation rate (p_{mut}), population size (pop_{size}), and number of generations ($n_{gen.}$). Regarding the penalty parameters, one should not only set the values of the penalty constant (PC) and penalty exponent (PE), but also the references used to compute the different constraints.

As referred in Section 4.1, two design variables are selected for the case study. The

first variable, $DV1$, is the stacking sequence of the shell and the second one, $DV2$, corresponds to the number of layers of the main spar caps. Considering the materials and the maximum number of layers allowed for the shell, presented in Section 4.2, it is possible to establish the bounds for the first design variable. The SST for the shell has 340 lines, therefore the design variable associated with this structure should assume an integer value between 1 and 340 ($1 \leq DV1 \leq 340$). Regarding the design variable that controls the number of layers in each spar cap of the main spar, an upper limit of 16 layers was defined. Due to the fact that the structure under study is from a model aircraft and some unpredictable loads may occur during operation, it was agreed that both caps should have the same number of layers. The choice of the $DV2$'s upper limit was ruled by two factors: the number of bits needed to represent the design variable and the exploration of the design space. According to the procedure applied by the optimizer to compute the number of bits needed to represent a given integer, this value (16) can be represented using precisely 3 bits which ensures that this design variable will not assume values that are out of bounds, i.e. bigger than 16. From the design space exploration point of view, sixteen layers as the maximum number of plies to be laminated in each cap is considered to provide enough margin in the search for lighter and feasible solutions for the problem. From the analysis of the optimization results (Section 4.5.2), it is clear that the chosen settings are adequate.

In which concerns the definition of bounds for the C1 constraint, a value of ten percent of the semi-span length is admissible for the wing tip deflection. Since the central panel tip is located at one fifth of the semi-span value ($\eta = 0.20$), by applying Equation (3.26), the resultant deflection varies from 0.0 and 13.1 mm, for the minimum ($\delta_{tip_{min}}$) and maximum values respectively ($\delta_{tip_{max}}$). The values for the central's panel tip twist are deduced in a similar fashion. Considering that a two degree change in the tip of the wing is acceptable for the ACC 2019 aircraft, the twist variation at the central panel's tip is limited to 0.4º, setting the bounds used to compute the C2 constraint as -0.4º and 0.4º. The failure theory selected to assess the presence of failure in the composite structures (shell and spar webs) is the Tsai-Hill criterion, which sets the reference value for the C3 constraint ($FT_{failure}$) as 1.0. Additionally, for this constraint it is also necessary to define the radius of the sphere used to assess the existence of failure. The radius *r* of the sphere used in the calculation of this constraint was set as 30 mm since it is considered a reasonable value based on empirical experience. The process of checking if failure occurs in the spar caps is achieved using the safety margin values of the bar elements. Since the definition of safety margin states that failure occurs if this parameter assumes values lower than zero, this is the reference value applied to compute the C4 constraint. Table 4.8 is a summary of the parameters' bounds used to compute the constraints during the optimization through application of Equations (3.24), (3.28), (3.31), and (3.33) presented in the Methodology chapter.

To fully define the optimization problem, one should assign a value for the penalty constant (PC) and penalty exponent (PE). For this case study, the values of 100

and 2 were selected as PC and PE, respectively. The penalty exponent value selected corresponds to a typical value while the penalty constant was set considering the mass of the central panel for the baseline, corresponding to around 30% of the panel's mass. To summarize all the details presented for the case study regarding design variables, constraints, and penalty parameters, one should refer to Table 4.8 and Equation (4.2). The latter corresponds to the optimization problem statement.

Table 4.8: Constraints' bounds for the case study

Parameter	Value	Constraint
$\delta_{tip_{min}}$	0.00 mm	C1
$\delta_{tip_{max}}$	13.1 mm	C1
$\theta_{tip_{min}}$	-0.400°	C2
$\theta_{tip_{max}}$	0.400°	C2
$FT_{failure}$	1.0	C3
SM_{min}	0.0	C4

$$\begin{aligned} \min_{DV1,DV2} \quad & \sum_{j=1}^{1}\left(m_{shell_j}+\sum_{k=1}^{2}m_{web_{jk}}+\sum_{k=1}^{2}m_{cap_{jk}}\right)+100\sum_{l=1}^{4}(Cl)^2 \\ \text{s.t.} \quad & 1 \leq DV1 \leq 340 \\ & 1 \leq DV2 \leq 16 \end{aligned} \tag{4.2}$$

Considering that one of the goals of this thesis is to understand the set of optimization parameters (p_{mut}, pop_{size}, $n_{gen.}$) that better suit this problem, the exploration of the design space is performed using four mutation rates in combination with two population sizes, and two number of generations. Regarding the crossover operation, it was considered as being uniform for all the performed simulations. The decision to not studying the effect of the crossover operation is based on the conclusions of Williams and Crossley [21]. The selection is performed using the tournament criteria since it is the strategy implemented in the selected genetic algorithm driver.

$$\begin{cases} p_{mut} = \Big\{0.03; 0.05; 0.10; 0.15\Big\} \\ pop_{size} = \Big\{20; 30\Big\} \\ n_{gen.} = \Big\{10; 20\Big\} \end{cases} \tag{4.3}$$

Equation (4.3) presents the values selected for the mutation rates, population size, and number of generations for this study. The mutation rates are selected in order to ensure that the random effect of the search is considered with different magnitudes, being 0.05 a common mutation rate [16]. The population sizes are based on a rule of thumb that indicates that the number of individuals in a certain population should be, at least, an order of magnitude higher than the number of design variables. The total number of

individuals of each simulation can be computed using Equation (2.4), and the results using the selected parameters are presented in Table 4.9.

Table 4.9: Total number of individuals of each simulation based on the population size and number of generations, and expected time of analysis (ETOA)

pop_{size}	n_{gen}	$n_{individuals}$	ETOA [h]
20	10	220	7.6
20	20	420	14.6
30	10	330	11.5
30	20	630	21.9

The optimizations processes are performed in the main computer from the Structures and Vibrations laboratory which has 12 cores, Intel ®Xeon(R) CPU E5-2620 0 @ 2.00GHz, and 64 GB of RAM. To prevent CPU overloading, and ensure that operative system tasks are mantained, the optimization processes are launched using 10 cores in parallel ($n_{proc.parallel}$). According to Equation (4.4), the time spent during an optimization (ETOA) depends on the number of processes in parallel and in the time required to perform a structural analysis (SAT). Considering the results of the mesh convergence study presented in the previous section, the expected number of hours necessary to complete each optimization process are presented in the fourth column of Table 4.9. For each arrangement of optimization parameters, two runs are performed, resulting in a total of 32 simulations, and an expected total time of 445 hours. At this point, before starting the discussion of the obtained results, it is important to refer that any of the performed optimizations are independent from the others since the initial population is randomly generated.

$$ETOA = \frac{pop_{size}}{n_{proc.parallel}} \cdot (n_{gen.} + 1) \cdot SAT \quad (4.4)$$

4.5.2 Optimization results

The discussion of the results starts by analyzing the solutions found during the optimization phase of this study, followed by some considerations regarding the optimization parameters. Before closing this section and proceeding to the conclusions and possible future work, a presentation of the recommended optimization parameters and the Annex containing the results of the optimization processes are given.

From the performed optimizations, seven designs were found for the structure of the wing's central panel. The results are presented in Table 4.10 together with the baseline design (Design 0) for comparison purposes. This table contains the values of the design variables ($DV1$ and $DV2$) and objective function (of) of the different designs, as well as the values of the panel's tip deflection and twist. All designs found are penalty free, meaning that the presented structural configurations are compliant with the deformation

and failure criteria constraints thus the simplest form of comparing these solutions with the baseline design is to quantify the difference in the mass value. The improvement column of Table 4.10 is computed using Equation (4.5), in which m_0 is the baseline design mass and $m_{Design,i}$ corresponds to the mass of each design, and represents the percentual mass reduced with respect to the original configuration. With the help of this column it is easily understandable that not all the designs correspond to a reduction in the panel's mass. Design 1 is the structural configuration that represents the highest mass saving, around 16.5%, followed by designs 2 (12%) and 4 (12%), 3 (10%), and 5 (7.8%). Designs 6 and 7 are heavier than the baseline structure, 4.4% and 11% respectively. The heavier designs use a five layer shell while the lighter ones have three layers in this structure.

$$Improvement = \frac{m_0 - m_{Design,i}}{m_0}, \quad \text{i} = 1, 7 \quad (4.5)$$

By directly comparing Designs 1, 2, and 3, one can affirm that, for the same shell stacking sequence, the addition of layers in the spar caps leads to a reduction in the tip displacement value as well as in the tip twist value. This can also be observed for designs 4 and 5. For the case where the number of layers in a spar cap is kept constant and the carbon fiber layer of the shell changes its direction from 0 degrees to 15 degrees with the longitudinal direction of the panel (Design 2 vs Design 4), an increase of the displacement value occurs. This result is explained by the fact that changing the orientation by 15º, a decrease in the flexural stiffness of the wing over the X-axis occurs, leading to a higher deflection. This change in the orientation of a layer also has implications in the twist value although at a smaller scale. Increasing the ply angle by fifteen degrees results in increasing the torsional stifness of the shell, which makes it twist more than the less stiffer configuration (Design 2). None of this behaviours are just dependent on one structure (spar caps or shell), and as presented, it is possible to capture the contributions of each structure to the displacement and twist values of the panel. In any case, the spar caps appear to have a bigger influence which might be explained by the larger contribution to the second moment of area of the cross-section as well as by the alignment and application of the forces.

Table 4.10: Baseline and designs found during optimization

Design	DV1	DV2	of	δ_{tip} [mm]	θ_{tip} [º]	Improvement (%)
0 (baseline)	20	10	326.50	3.90	0.77	-
1	1	11	272.39	3.61	0.35	16.57
2	1	13	286.79	3.15	0.25	12.16
3	1	14	293.99	2.97	0.21	9.96
4	2	13	286.79	3.28	0.38	12.16
5	2	15	301.19	2.91	0.30	7.75
6	13	12	340.91	3.09	0.39	-4.41
7	14	15	362.51	2.71	0.38	-11.03

One should note that all the solutions found have more layers ($DV2$) in the spar caps than the value of the baseline design, which could be related with the appearence of structural failure, with the need to comply with the twist constraint, or both. From observation of the data presented in the fifth and sixth columns of Table 4.10, it seems that the twist constraint plays a more active role limiting the solution than the displacement constraint. Recalling the bounds established for this problem, Table 4.8, it is noticeable that the values for the tip displacement (δ_{tip}) have a good margin (between 9.5 mm and 10.4 mm) for the upper bound (13.1 mm) used to calculate the C1 constraint while, on the other hand, the values obtained for the tip twist (δ_{tip}), that are between 0.21º and 0.38º, are much closer to the upper limit (0.4º).

To better understand why more layers in the spar caps are required, an additional structural analysis was performed where the effect of removing one layer was assessed. From this study one concluded that by simply removing one layer of each cap, the resultant value for the tip twist would be over the established limit (0.4º) but no failure would occur. Hence, the results obtained during the optimization are not limited by the occurence of failure in the spar caps, but by the constraint imposed for the twist. The more layers exist in the spar caps, the higher the bending stiffnesses of these structures are over the vertical axis (Z-axis) and the horizontal axis (X-axis). Since the wingtip twist computation depends on the spar caps nodes positions, the increase in both bending stiffnesses leads to a smaller displacement along the X and Z-direction for the same loading conditions, reducing the wingtip twist value.

Table 4.11: Stacking sequences codified by design variable 1

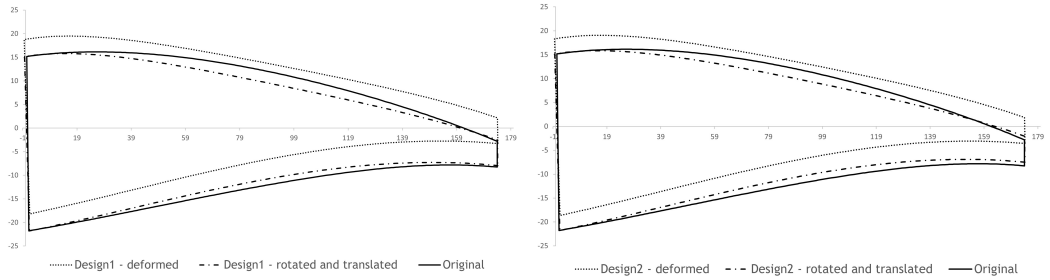
SST line/DV1	L1	L2	L3	L4	L5	γ_{shell} [g/m^2]	t [mm]
1	1	2	-	-	-	357.00	1.66
2	1	3	-	-	-	357.00	1.66
13	1	4	2	-	-	489.00	1.82
14	1	4	3	-	-	489.00	1.82
20	1	5	5	-	-	489.00	1.82

Table 4.11 presents the stacking sequences codified by design variable 1 which, with help of the information from Table 4.5, can be translated into a group of plies, each ply being defined by a material and orientation. Designs 1 to 3 have a shell composed by a balsa core of 1.5 mm reinforced with a carbon fiber layer of 30 g/m^2 on each side, which direction 1 of the fabric is aligned with the longitudinal axis of the panel. For designs 4 and 5, the stacking sequence just differs from the previous one on the angle of the reinforcement layer. In these sequences, the carbon fabric is placed in a way that an angle of 15º exists between the direction 1 of the fabric and the longitudinal axis of the panel. The other two designs found present a shell that uses two reinforcement layers per side, instead of just one, which makes these solutions heavier and thicker than the previous ones. In these cases, the first layer laminated on balsa core is oriented in a direction that makes an angle

of 30º with the longitudinal direction of the panel, while the direction of the second layer is 0º for Design 6 and 15º for Design 7.

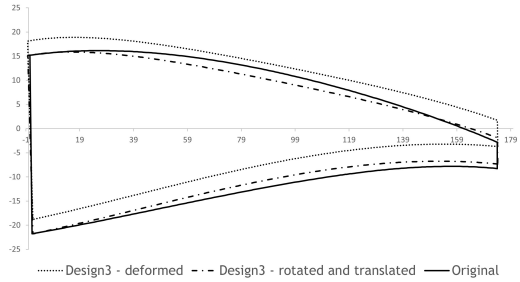
The reduction of the panel's mass occurs due to a selection of a lighter shell configuration. Besides the differences already presented related with the tip displacement and twist, the changes in the shell's structural arrangement appears to have consequences regarding the aerodynamic performance since the deformed shape does not maintain the airfoil's original geometry. Figure 4.7 is a cross-section view of the central's panel tip where it is possible to compare the deformed and the original shapes for each of the designs. In order to obtain the "rotated and translated" curves, one uses the deformed shape and rotates it back using the tip twist angle (θ_{tip}) and translates it -δ_{tip} mm, in such a way that the planes of the main spar web of the deformed and original stages coincide. From observation of Figure 4.7 it is possible to check that the deformed shape ("rotated and translated") presents some deviations from the original airfoil countour. This changes are more pronounced for the outer surface, since the gap between curves is bigger, however some changes are also noted for the inner surface. From a general point of view, one can state that the designs that use more layers in the shell structure (Design 6 and 7) tend to maintain their shapes closer to the original contour when compared to the lighter ones. This result is particularly evident for the lower surface and is expected to be related with the shell's torsional stiffness.

Even though these differences have not been quantified, some points can be laid down regarding the shell's deformation. Starting with the lighter solutions (Design 1 vs Design 2 vs Design 3), that present the same shell stacking sequence and different values for the number of layers of the spar caps, it is noticeable that as the caps get thicker the deviation of the lower surface gets more pronounced. This observation is in line with what was discussed regarding the influence of the spar caps in the torsional behaviour of the section. Since thicker caps increase the torsional constant, thus reducing the twist of the main spar, it is expected that for the same loading conditions the plate elements of the shell should be required to support higher in-plane loads. These conditions might lead to larger deformations of the outer surface of shell when compared to solutions where a less stiff spar is employed. For the case where the number of layers of the spar caps is kept constant and a rotation of 15º is applied to the carbon layers in the shell structure, Design 2 versus Design 4, it is possible to observe that the lower surface presents a smaller deviation for the case that uses the rotated layers. This result can be explained, again, by the increment in the torsional stiffness value of the shell due to the rotation. Regarding the upper surface, it seems that Design 4 presents a bigger and more extent gap when compared to Design 2. This is the result of a stiffer behaviour that transmits the loads around the structure with less deformation in the lower surface, since it is almost straight, until the shell has no ability to maintain its shape due to the curvature of the upper surface.

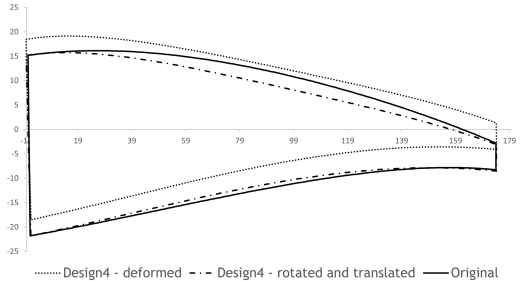


(a) Design 1

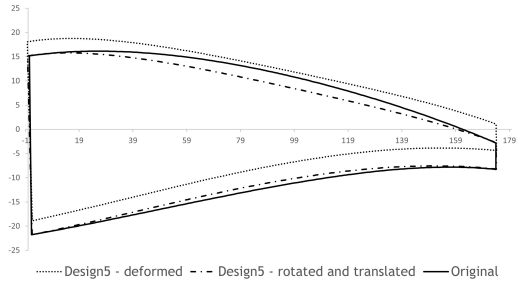
(b) Design 2



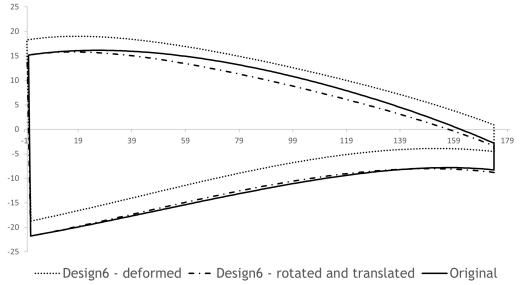
(c) Design 3



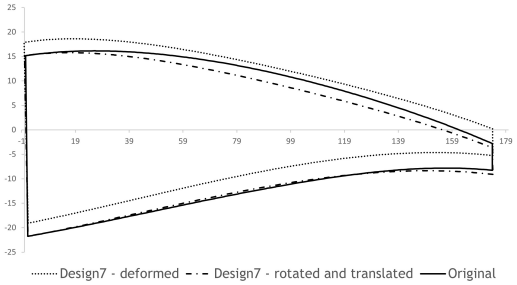
(d) Design 4



(e) Design 5



(f) Design 6



(g) Design 7

Figure 4.7: Comparison between deformed and original tips' cross-section

Considering that the loss of aerodynamic performance is highly prejudicial for the aircraft performance, a more detailed investigation of this issue is suggested to improve the results obtained when using the optimization tool. To address this problem in future versions of this tool, two approaches are suggested: addition of ribs or implementation of a new constraint. Adding ribs should not be a complex task since it would use features already implemented. However, it may require some flexibility to deal with the mesh part considering that it might require the computation of extra nodes/grids and interfacing with different structures (shell, webs, and caps). Adding a new constraint that quantifies the differences between the deformed and original shapes corresponds to another

approach that can help solving this issue. This constraint could use the information of the deformed cross-section and the data associated with the original one, and compute the average distance between points of both cases for the same position (x/c). Another possibility would be to compute and compare the slopes' values along the countours of the deformed and original shapes, imposing another constraint by setting a maximum value for the difference.

Before jumping into a more direct analysis of the optimization parameters, a comparison between the expected time to complete the simulations and the real time is held nextly. Table 4.12 presents the average time spent to complete a simulation with pop_{size} and n_{gen} parameters. These values were computed considering the times elapsed in the simulations of all mutation rates. Comparing the time column of Table 4.12 with the ETOA column of Table 4.9, it is clear that the predictions are quite away from the real elapsed time. These differences are based on the fact that the structural solver writes the solving status in the terminal during execution. Since this status often involves writing logs of operations as, for example, checking if all elements are correctly defined, and counting the total number of elements, the delay on the execution starts to appear due to the fact that all parallel processes are writing into the same terminal. Considering that the terminal does not support parallel writing, when one processor is logging the status, the others must wait until it is finished. One possible upgrade to this limitation might be removing the logging instructions or make possible to execute the tasks of each processor in a specific terminal. In any case, the time gains due to executing the optimization in parallel are superior to its downsides, since the other option would be a serial execution.

Table 4.12: Elapsed simulation time for each combination of pop_{size} and n_{gen} parameters

pop_{size}	n_{gen}	Time [h]
20	10	14.3
20	20	20.3
30	10	26.0
30	20	38.3

Table 4.13 presents the results obtained, per run, for the different combinations of optimizations parameters (p_{mut}, $n_{gen.}$, pop_{size}). The column "Pen." refers to the total penalty apllied to the solution. As referred, since no penalties are applied to the designs, all the solutions found are feasible. Regarding the number of generations ($n_{gen.}$), the best solution is found 10 times when using a total of 21 populations (20 generations) versus 8 times when 10 generations are selected. For the population sizes tested, the lightest solution was found an equal number of times, 9, suggesting that any of the options is suitable to study the problem.
From the data in the table, one observes that the lightest solution is found using any of the mutation rates tested. For a mutation rate of 0.05 (Table 4.13b), the optimal solution is found in 6 runs of the possible 8 whereas for the other three mutation rates, the best solution is found in 4 runs of the 8. The heavier designs are discovered only once when

using higher mutation rates, 0.10 and 0.15. In terms of design variable 1, the stacking sequence for the shell structure that is more frequently found is the value of 1 (25 times), followed by the SST line of 2 (5 times). For the second design variable, the most common value found is 11 (18 times).

Considering the results, the mutation rate that seems more adequate to the problem is 0.05. In fact, this mutation rate seems to provide consistent results when using 30 elements in the population since, in these cases the obtained solution is the lightest one for any value of number of generations.

Table 4.13: Results of the optimization runs for different optimization settings.

(a) Mutation rate: 0.03

		Run 1				Run 2			
Population size	Number of generations	DV1	DV2	of	Pen.	DV1	DV2	of	Pen.
20	10	1	13	286.79	0	1	11	272.39	0
	20	1	13	286.79	0	1	11	272.39	0
30	10	1	11	272.39	0	1	13	286.79	0
	20	1	11	272.39	0	2	13	286.79	0

(b) Mutation rate: 0.05

		Run 1				Run 2			
Population size	Number of generations	DV1	DV2	of	Pen.	DV1	DV2	of	Pen.
20	10	1	14	293.99	0	1	11	272.39	0
	20	1	13	286.79	0	1	11	272.39	0
30	10	1	11	272.39	0	1	11	272.39	0
	20	1	11	272.39	0	1	11	272.39	0

(c) Mutation rate: 0.1

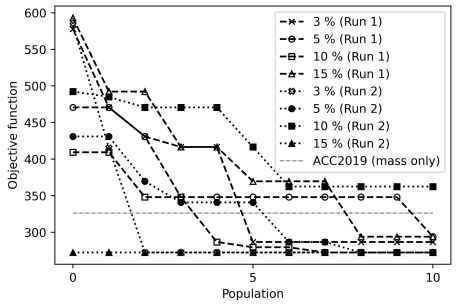
		Run 1				Run 2			
Population size	Number of generations	DV1	DV2	of	Pen.	DV1	DV2	of	Pen.
20	10	1	11	272.39	0	14	15	362.51	0
	20	2	13	286.79	0	1	11	272.39	0
30	10	1	11	272.39	0	1	14	293.99	0
	20	2	13	286.79	0	1	11	272.39	0

(d) Mutation rate: 0.15

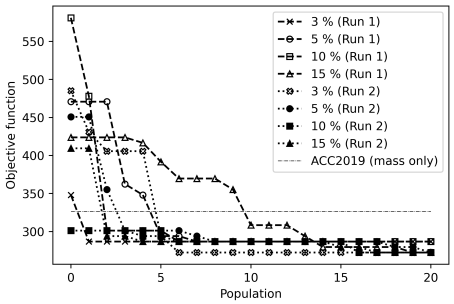
		Run 1				Run 2			
Population size	Number of generations	DV1	DV2	of	Pen.	DV1	DV2	of	Pen.
20	10	1	14	293.99	0	1	11	272.39	0
	20	1	11	272.39	0	1	11	272.39	0
30	10	13	12	340.91	0	2	15	301.19	0
	20	1	11	272.39	0	2	13	286.79	0

Figure 4.8 shows the objective function value of the fittest point during the optimization processes for different optimization parameters. Using this picture, it is possible to note some points regarding the optimization process. Since all runs are independent, due to the fact the initial populations are randomly generated, it is not possible to judge on how fast a better solution is found by the use of different mutation rates, however when

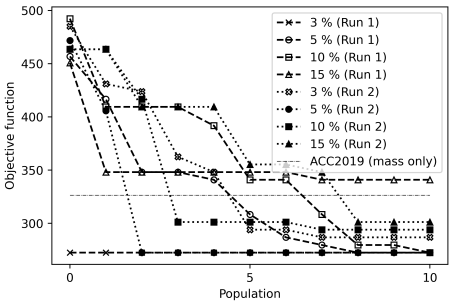
Design 1 is found there is no further improvement in the objective function value during the remaining of the run.
Through observation of the figures that result from the application of a higher number of generations (Figures 4.8b and 4.8d), it is noted that after the fifteenth generation there is the tendency to not improve the solution. There is no apparent reason that relates this tendency with the mutation rate, however with more simulations some connections might be identified.



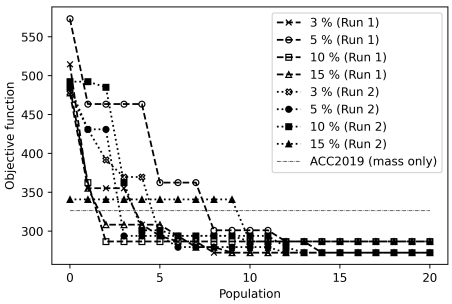
(a) 10 generations with 20 elements in the population



(b) 20 generations with 20 elements in the population



(c) 10 generations with 30 elements in the population

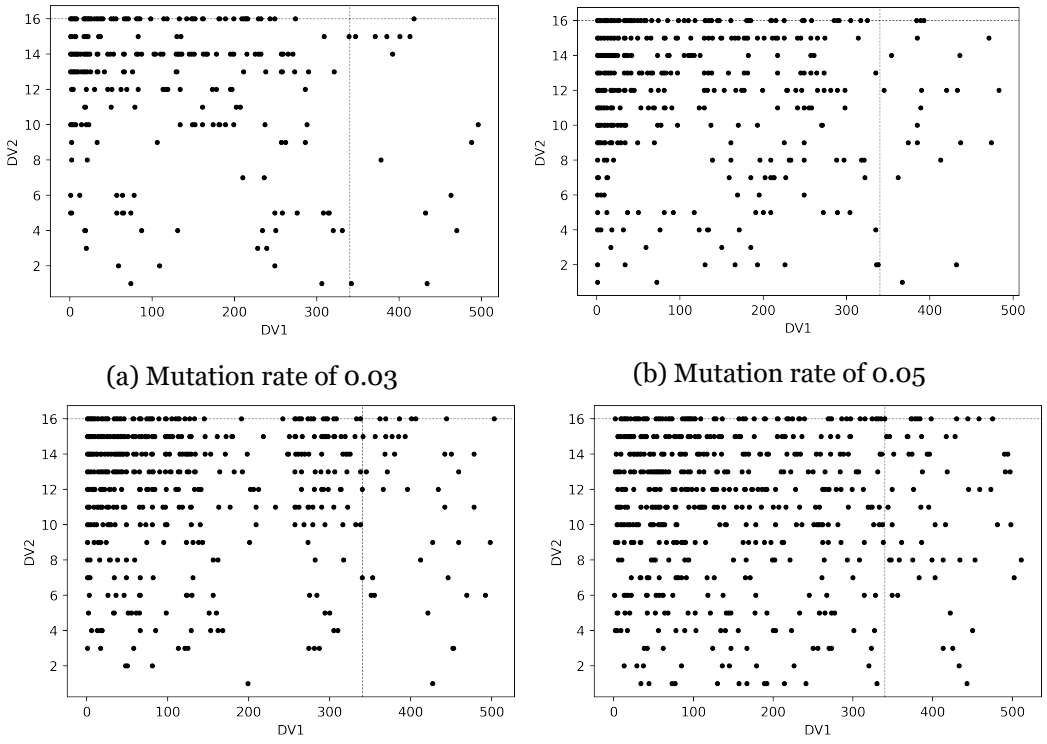


(d) 20 generations with 30 elements in the population

Figure 4.8: Evolution of the best fitness point during the optimization processes for different mutation rates

Figure 4.9 shows how the mutation rate value can affect the search of solutions. In these figures, the dashed lines correspond to the design variables bounds (340 for DV1 and 16 for DV2). Although these pictures are taken from four of the thirty two performed runs, they are considered as examples of the registered behaviours concerning the mutation rate setting, since searches using the same mutation rate tend to exhibit similar profiles. In these plots, one observes that the mutation rate value plays a significant role on how disperse the individuals are in the design space.
The clear trend that can be inferred from these results is as the mutation rate increases the more disperse the individuals are, hence a bigger coverage of the design space is performed. For lower mutation rates, the search tend to be more local since the evolution process falls on the crossover operator. As the mutation rate increases, not only the dispersion accross the design space is wider but also it appears to be more uniform. In

general, this capability is interesting, particularly, to explore different possibilities in early design stages, trying to identify possible areas to perform more local searches. From Figures 4.9a - 4.9d it is possible to identify a concentration of points in the area $\{0 \leq DV1 \leq 100$ and $10 \leq DV2 \leq 16\}$, suggesting that a higher number of feasible solutions might be located there.



(a) Mutation rate of 0.03

(b) Mutation rate of 0.05

(c) Mutation rate of 0.10

(d) Mutation rate of 0.15

Figure 4.9: Search across the design space using different mutation rates

The use of higher mutation rates comes with the drawback of potentially having more individuals with design variables out-of-bounds, conditioning the evolution process. As already explained, the out-of-bounds situation occurs when at least one of the design variables assumes a value that is beyond the defined bound. When using binary encoded genetic algorithms, this situation can appear when the number of bits used to represent a variable allows the representation of numbers bigger than the bound. In this case study, this issue can only occur with the design variable associated with the shell's stacking sequence. Since 9 bits are required to represent the integer number of 340, a gap of 172 integers exists between 340 and the maximum integer that can be represented with this number of bits, 512. During the optimization processes, some individuals had a design variable 1 with value in this interval, between 341 and 512, leading to out-of-bounds individuals.

Table 4.14 summarizes the total number of individuals which design variable is out-of-bounds versus the mutation rate value. The values presented in the second column are totals, this meaning that it corresponds to the sum of all the individuals that are not

analyzed in the 8 runs performed for each mutation rate. As observed in this table, as the mutation rate increases, the number of individuals out-of-bounds tend to increase. This is explained by the fact that as the mutation rate becomes higher, the possiblity of one bit to mutate, change from 0 to 1 or vice-versa, increases. This way, higher mutation rates tend to favor a more random search leading to the appearence of more out-of-bounds individuals. From the data presented in the table, the percentage of out-of-bounds individuals increases with the mutation rate. For the mutation rate of 0.15, this percentage value is four times higher (12%) than the value observed for the mutation rate of 0.03 (4%)

Table 4.14: Individuals with design variables out-of-bounds for different mutation rates

Mutation rate	Out-of-bounds (%)
0.03	128/3200 (4%)
0.05	158/3200 (5%)
0.10	255/3200 (8%)
0.15	368/3200 (12%)

Even tough an attempt was made to run as much optimization processes as possible, it is considered that performing more simulations will make possible to sustain a bit more the conclusions regarding the optimization parameters. In short, from the presented discussion, the optimization parameters that might lead to find the optimal solution in future applications of this tool in similar problems are:

- Elitism: 1 individual
- Selection: Tournament
- Crossover: Uniform
- Mutation rate: 0.05
- Population size: 30
- Number of generations: 15 to 20

A detailed view of each optimization process is presented in Appendix B. In this Appendix, the reader finds four plots for each combination of mutation rate (in ascending order), population size, number of generations and run:

- History of the fittest individual in the population: objective function, design variables, and penalty;
- Design variables' values per population;
- DV1 vs DV2;
- Number of individuals per population with out-of-bounds design variables;

Chapter 5

Conclusions and Future Work

5.1 Conclusions

From a global point of view, all the objectives proposed for this thesis were achieved and a structural optimization tool of composite wings implemented. The framework developed uses MYSTRAN as the structural solver and the genetic algorithm from OpenMDAO as optimization driver. Parallel evaluation of individuals plays an important role by significantly reducing, the time spent on the optimization process. The "StructOpt - StandAlone" can be used without the optimization driver, providing an easy way to perform static structural analysis of composite wings (e.g.: for the mesh convergence process). Single cell (wingbox) or double-cell structures can be studied using the developed computational tool. From the optimization point of view, four constraints were implemented (two geometric and two failure related), as well as the objective function, based on the wing's mass, to evaluate the fitness of an individual in the population.

Due to computational costs, the case study focused the optimization of the wingbox structure from UBI's ACC 2019 aircraft central panel. As a result of a mesh convergence study, the mesh selected has a total of 69900 elements, taking around 21 minutes to solve the structural analysis. From the mesh convergence process, it is observable that the value of the safety margin of the "BAR" elements does not vary significantly with the number of divisions along the panel.

To understand the best setting for this type of problems, four mutation rates were tested in combination with two different population sizes (20 or 30 individuals), and number of generations (10 or 20). Two runs were performed for each arrangement of mutation rate - population size - number of generations, resulting in a total of 32 optimization processes being accomplished. The penalty constant was set at 100 and the penalty exponent at 2. From these simulations, seven possible solutions for the central panel were found, all feasible considering the implemented constraints. The best result corresponds to a mass reduction of 16.5%. Considering the values of the geometrical constraints presented for each design, the panel's tip twist appears to present a big influence on the number of layers of each spar cap, in order to keep the design feasible. It was observed that higher mutation rates tend to present more individuals with design variables out of the bounds. The mutation rate that seems to favour the optimization process is 0.05, since the best result was found more consistently using this setting. Regarding the number of individuals in a population and the number of generations, it is hard to draw sustained conclusions. However, from the performed simulations, the searching process using 20 individuals and 30 generations tends to find the lightest solution more frequently. A higher number

of simulations are needed to better assess the optimization settings.

5.2 Future Work

Considering the results and conclusions, as well as the issues found during the development of the computational tool, there is some room for improvement. The following topics highlight changes that might be implemented to help with solving the problems and/or expanding the tool.

To address the shell's stiffness problem, one can implement ribs and/or a new constraint that measures the changes in the airfoil's shape. Adding ribs will ensure that the airfoil shape is maintained, enabling the use of the current constraint to obtain more realistic designs. Implementing a new constraint, that quantifies the change in the airfoil's shape, is expected to provide feasible designs, since the shell's stiffness issue can be fed back to the optimizer.

Another important needed improvement is to use a mesh based on quadrilateral elements instead of triangular ones. The quadrilaterals will provide a better stress resolution, because of a higher order of the approximation function commonly used in its formulation, thus reducing the computational cost of the analysis because smaller numbers of elements can be used. This will allow faster optimization processes and enable the study of a double-cell wing with the inclusion of the leading edge portion.

Regarding the tool's expansion topic, one must develop some upgrades for the bulk data file generator and for the optimization part to study a multi-panel wing considering several sub-cases. Even though the mesh section of the BDF generator allows the discretization of several panels, the connections between panels need to be implemented using, for example, rigid elements connecting the spar caps. To study several sub-cases, and assuming that different boundary conditions might be required, the application of constraints at the nodes (PSPC) must be moved to each subcase definition. To accommodate these changes in the optimization part of the code, some tweaks must be applied to the scripts to allow reading sub-cases from the F06 file.

To better understand the performance of the stacking sequence tables in the structural optimization of this type of problems, more data should be collected from the optimization process. Knowing which constraints are active for a structural configuration in the population, will provide useful information to rate this technique (SSTs) versus others that might be applied. This information, together with more runs, is important to conclude which set of optimization parameters (number of generations, population size, mutation rate) is best.

Bibliography

[1] AKAModell Stuttgart e.V., “Regulations for the Air Cargo Challenge 2019 in Stuttgart,” Stuttgart, 2019. 1

[2] J. M. da Silva, *Design and Optimization of a Wing Structure for a UAS Class I 145 kg, MSc Thesis, Academia da Força Aérea*, 2017. 5

[3] K. K. Rumayshah, A. Prayoga, and M. Agoes Moelyadi, “Design of High Altitude Long Endurance UAV: Structural Analysis of Composite Wing using Finite Element Method,” in *Journal of Physics: Conference Series*, vol. 1005, no. 1, 2018. doi: 10.1088/1742-6596/1005/1/012025. ISSN 17426596 5

[4] A. Gillet, P. Francescato, and P. Saffre, “Single- and multi-objective optimization of composite structures: The influence of design variables,” *Journal of Composite Materials*, vol. 44, no. 4, pp. 457–480, 2010. doi: 10.1177/0021998309344931 5, 12

[5] R. Sorenson and J. Kann, *Optimisation of composite structures using lamination parameters in a finite element application, MSc Thesis, Aalborg University*, 2011. 5

[6] M. Sprengholz, H. Traub, M. Sinapius, S. Dähne, and C. Hühne, “Rapid transformation of lamination parameters into stacking sequences,” *Composite Structures*, vol. 276, 2021. doi: 10.1016/j.compstruct.2021.114514 5

[7] A. Arias-Montaño, C. A. Coello, and E. Mezura-Montes, “Multiobjective evolutionary algorithms in aeronautical and aerospace engineering,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 662–694, 2012. doi: 10.1109/TEVC.2011.2169968 6, 9

[8] X. Zhong, P. Jin, and Q. Han, “A method for composite wing box optimization with manufacturing constraints,” in *Proceedings of the 2015 International Conference on Power Electronics and Energy Engineering*, vol. 20, 2015. doi: 10.2991/peee-15.2015.61 6

[9] H. Yin and X. Yu, “Integration of manufacturing cost into structural optimization of composite wings,” *Chinese Journal of Aeronautics*, vol. 23, no. 6, pp. 670–676, 2010. doi: 10.1016/S1000-9361(09)60269-7 6

[10] A. Długosz and W. Klimek, “The optimal design of UAV wing structure,” in *AIP Conference Proceedings*, vol. 1922, 2018. doi: 10.1063/1.5019124. ISBN 9780735416147. ISSN 15517616 6

[11] H. Ghiasi, D. Pasini, and L. Lessard, “Optimum stacking sequence design of composite materials Part I: Constant stiffness design,” *Composite Structures*, vol. 90, no. 1, pp. 1–11, 2009. doi: 10.1016/j.compstruct.2009.01.006 7

[12] M. T. McMahon, L. T. Watson, G. A. Soremekun, Z. Gürdal, and R. T. Haftka, "A Fortran 90 Genetic Algorithm Module for Composite Laminate Structure Design," *Engineering with Computers*, vol. 14, no. 3, pp. 260–273, 1998. doi: 10.1007/BF01215979 7

[13] F. X. Irisarri, A. Lasseigne, F. H. Leroy, and R. Le Riche, "Optimal design of laminated composite structures with ply drops using stacking sequence tables," *Composite Structures*, vol. 107, pp. 559–569, 2014. doi: 10.1016/j.compstruct.2013.08.030 7

[14] F. Farzan Nasab, H. J. M. Geijselaers, I. Baran, and A. de Boer, "Generating the Best Stacking Sequence Table for the Design of Blended Composite Structures," *Advances in Structural and Multidisciplinary Optimization*, pp. 779–788, 2018. doi: 10.1007/978-3-319-67988-4-59 7

[15] G. Venter, "Review of Optimization Techniques," *Encyclopedia of Aerospace Engineering*, 2010. doi: 10.1002/9780470686652.eae495 7

[16] J. R. R. A. Martins and A. Ning, *Engineering Design Optimization*. Cambridge University Press, 2021. ISBN 9781108833417 7, 8, 52

[17] D. W. Coit and A. E. Smith, "Constraint-Handling Techniques - Penalty Functions," in *Handbook of Evolutionary Computation*. Bristol, U.K.: Institute of Physics Publishing and Oxford University Press, 1997, ch. C5.2. 8, 33

[18] T. Bhoskar, O. K. Kulkarni, N. K. Kulkarni, S. L. Patekar, G. M. Kakandikar, and V. M. Nandedkar, "Genetic Algorithm and its Applications to Mechanical Engineering: A Review," *Materials Today: Proceedings*, vol. 2, no. 4-5, pp. 2624–2630, 2015. doi: 10.1016/j.matpr.2015.07.219 9

[19] Q. Wang, P. Spronck, and R. Tracht, "An overview of genetic algorithms applied to control engineering problems," *International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1651–1656, 2003. doi: 10.1109/icmlc.2003.1259761 9

[20] R. T. Haftka, "Genetic Algorithms for Optimization of Composite Laminates," *Mechanics of Composite Materials and Structures*, pp. 431–442, 1999. doi: 10.1007/978-94-011-4489-6-28 9

[21] E. A. Williams and W. A. Crossley, "Empirically-derived population size and mutation rate guidelines for a genetic algorithm with uniform crossover," *Soft Computing in Engineering Design and Manufacturing*, pp. 163–172, 1998. doi: 10.1007/978-1-4471-0427-8-18 10, 52

[22] J. S. Gray, J. T. Hwang, J. R. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, vol. 59, no. 4, pp. 1075–1104, 2019. doi: 10.1007/s00158-019-02211-z 11

[23] J. P. Jasa, J. T. Hwang, and J. R. Martins, "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, vol. 57, no. 4, pp. 1815–1827, 2018. doi: 10.1007/s00158-018-1912-8 11

[24] OpenMDAO, "SimpleGADriver." [Online]. Available: http://openmdao.org/twodocs/versions/latest/features/building_blocks/drivers/genetic_algorithm.html 11

[25] J. N. Reddy, *Mechanics of Laminated Composite Plates and Shells*, 2nd ed. CRC Press LLC, 2003. ISBN 0849315921 12

[26] Z. Gürdal, S. IJsselmuiden, and J. van Campen, "Composite Laminate Optimization with Discrete Variables," *Encyclopedia of Aerospace Engineering*, 2010. doi: 10.1002/9780470686652.eae499 12

[27] R. T. Haftka and J. L. Walsh, "Stacking-sequence optimization for buckling of laminated plates by integer programming," *AIAA Journal*, vol. 30, no. 3, pp. 814–819, 1992. doi: 10.2514/3.10989 12

[28] C. H. Wang and C. N. Duong, "Failure criteria," in *Bonded Joints and Repairs to Composite Airframe Structures*, 2016, pp. 21–45. 12

[29] MYSTRAN, "MYSTRAN." [Online]. Available: https://www.mystran.com/ 13

[30] B. Case, *Users Reference Manual for the MYSTRAN General Purpose Finite Element Structural Analysis Computer Program*, 2020. 13, 36

[31] D. Sousa, P. Gamboa, and D. Melo, "Aerodynamic performance of aerofoils obtained from a geometric offset applied to a given initial aerofoil," *Open Engineering*, vol. 6, no. 1, pp. 711–723, 2016. doi: 10.1515/eng-2016-0098 17

[32] T. H. Megson, *Aircraft structures for engineering students: Sixth edition*, 2016. ISBN 9780081009147 29, 30

[33] Silva, F., Palmeira, R., Ferreira, M., Lousada, M., Domingues, R. and Morão, T., "AERO@UBI – Air Cargo Challenge 2019, Design Report," Universidade da Beira Interior, Tech. Rep., 2019. 43

[34] I. M. Daniel and O. Ishai, "Appendix A: Material Properties," in *Engineering Mechanics of Composite Materials*. New York: Oxford University Press, 2006. 46

[35] A. C. Materials, "Airex - Universal Structural Foam Data Sheet," p. 3, 2011. [Online]. Available: https://www.3accorematerials.com/uploads/documents/TDS-AIREX-C70-E_1106.pdf 46

Appendix A

Implementation and execution notes

This appendix presents some notes on the implementation and use of the optimization tool. The first section, “Implementation”, focus some details related with the computational tool such as the software versions, the files of each component, and instructions to compile the StructOpt-StandAlone. In the “Run/Execution” section it is described the steps needed to run an optimization process in parallel. The last section of this appendix, “Input files”, presents the structures of the input files used in this tool.

A.0.1 Implementation

As previously referred, this tool is composed of two major components: the bulk data file generator and FEA solver, and the optimization script. The following list presents the versions of the softwares used in this tool as well as some notes regarding the implementation.

- To take advantage of the parallel procesing, the tool was developed in the Ubuntu system (Ubuntu 21.04, 64-bit);
- Parallel runs are only possible using a terminal;
- OpenMPI is used as message passing interface;
- The Python version is 3.9.5;
- OpenMDAO version is 3.9.0;
- MYSTRAN version is 12.01;
- LLT2021 is the lifting line theory software;
- The StructOpt - StandAlone Fortran code was developed using Intel’s Fortran compiler;
- To collect some data related with the optimization process, small changes to the original GA code were applied;
- The python script used to generate the Paraview files needs to be adjusted for each case.

A list of the Fortran files and description of its content is presented in Table A.1.

ID	File	Description
1	ModuleGeometry.f90	Declaration of variables: geometry
2	ModuleMesh.f90	Declaration of variables: mesh
3	ModuleVariables.f90	Declaration of variables: general variables used across the code, such as iteration variables, counters
4	ModuleBDF.f90	Declaration of variables: BDF file structure related variables
5	ModuleFEA.f90	Declaration of variables: Finite Element Analysis
6	ModuleFiles.f90	Declaration of variables: names and locations of folders and files
7	ModuleLoads.f90	Declaration of variables: loading conditions
8	ModuleMaterials.f90	Declaration of variables: materials
9	ModuleOptimization.f90	Declaration of variables: structural configuration (number of caps, caps width, failure theory)
10	ModuleSST.f90	Declaration of variables: stacking sequence tables
11	Geometry.f90	Subroutines to read, process and compute the wing geometry
12	GeometryUtilitaries.f90	Additional subroutines used in the generation of the geometry
13	Mesh.f90	Subroutines related with mesh generation
14	MeshUtilitaries.f90	Additional subroutines used in the generation of the mesh
15	BDFinterface.f90	Subroutines used to write the bulk data file
16	Loads.f90	Subroutines to read, process, and transfer loads to the structure
17	Material.f90	Subroutines to read, and process materials, including generation of SSTs
18	MathUtilitaries.f90	Additional subroutines that define mathematical operations
19	offset.f90	Subroutines to offset airfoils
20	OptimizationInterface.f90	Subroutines that deals with the interface for the optimizer
21	output.f90	Subroutines to interface with Tecplot
22	StructOpt.f90	Main Fortran script
23	lapack	Linear algebra package subroutines

Table A.1: Fortran files list

A list of the Python scripts and description of its content is presented in Table A.2.

ID	Script name	Description
1	workers_module.py	Functions to allocate and deallocate workers
2	loads_module.py	Count the number of loads/subcases
3	sst_module.py	Functions to read, store and access stacking sequence tables
4	constraints_module.py	Functions to compute geometric contraints values (tip deflection and twist)
5	directories_module.py	Declaration of the directories needed for execution
6	materials_module.py	Functions to read materials' definition
7	geometry_module.py	Functions to read geomtry's definition
8	mesh_module.py	Functions to read mesh definition, compute number of elements and mesh quality
9	mass_module.py	Functions to compute the wing mass
10	optimization_module.py	Functions to execute the correct worker
11	ConstraintsAssess.py	Functions to assess failure contraints (failure theory and safety margin)
12	MainOpt.py	Main script of the optimization problem definition
13	PlotOptimisationHistory.py	Generate plots of the optimization results
14	SSTBits.py	Functions to study the size of an SST versus the number of bits of the design variables
15	graphics.py	Functions to convert F06 to VTK file

Table A.2: Python scripts list

Table A.3 is a summary of the files that are part of the computational tool. There are three types of files: IN, OUT, and OPT. The "IN" type are input files in which the user must define parameters or load information such as geometry, mesh, and materials. The "OUT" files are result of the execution of the bulk data file generator and/or the structural solver. "OPT" files are related with the optimization part and the data is loaded during the evolutionary process.

ID	File	Description	Type
1	WingGeom.txt	Input: Definition of wing geometry	IN
2	WingMesh.txt	Input: Definition of wing mesh	IN
3	Structures.txt	Input: Definition of SSTs' number of layers, and spar caps width	IN
4	MaterialsLibrary.txt	Input: Definition of materials	IN
5	Default.txt	Input: Definition of default parameters	IN
6	CriticalPoints.txt	Loads: Definition of the air density, and speed and load factor of each case	IN
7	Load1.txt	Loads: Values of the loads along the span for critical point 1	IN
8	Case.dat*	Results: BDF file for analysis	OUT
9	Case.F06*	Results: Solver output file containing the analysis' results (F06)	OUT
10	Afile1.dat	Project: Airfoil 1 coordinates	IN
11	solverpath.txt	Project: File that contains the FEA solver path	IN
12	Directories.txt	Project: File that contains the directories' path	IN
13	Case.txt	Project: Definition of the stacking sequences and caps' number of layers	IN
14	sstms.txt	Project: Main spar web stacking sequence table	OUT
15	sstrs.txt	Project: Rear spar web stacking sequence table	OUT
16	sstshell.txt	Project: Shell stacking sequence table	OUT
17	GRID.txt	Project: Nodes ID and coordinates	OUT
18	CTRIA.txt	Project: CTRIA elements ID and grids	OUT
19	CBAR.txt	Project: CBAR elements ID and grids	OUT
20	MSGrid.txt	Project: Main spar web nodes and coordinates	OUT
21	MaterialsMatrixFile.txt	Project: Materials' global identification	OUT
22	CombShell.txt	Project: Conversion between identification used in sstshell.txt and the global ID	OUT
23	CombMS.txt	Project: Conversion between identification used in sstms.txt and the global ID	OUT
24	CombRs.txt	Project: Conversion between identification used in sstrs.txt and the global ID	OUT
25	objectivefile.dat	Objective function value of each individual	OPT
26	optfile.dat	Best point during evolutionary process, objective function and design variables values	OPT
27	popfile.dat	Values of the design variables for each element in the populations	OPT

Table A.3: Description of the files used in the optimization tool

For the version intended to be used in the optimization process, only the files marked with an asterisk are outputted to avoid an increase of the simulation time.

A.0.1.1 Compilation

The process presented in this section is applied to compile and produce de executable files for both versions of the StructOpt-StandAlone: "StructOpt-StandAlone_Solo" and "StructOpt-StandAlone_Opt". This solution was developed in the main computer from the Structures and Vibrations laboratory (12 cores, Intel ®Xeon(R) CPU E5-2620 0 @ 2.00GHz , 64 GB of RAM).

Step 1: Open a terminal.
Step 2: Set the Intel Fortran Compiler vars.

```
source /opt/intel/oneapi/compiler/2021.3.0/env/vars.sh intel64
```

Step 3: Navigate to the folder of the version that the user wants to compile

StructOpt-StandAlone_Solo : `cd Desktop/FilipeSilva/MSc/OptimisationStructOpt/StructOpt-StandAlone/StructOpt/StructOpt/FortranCodeMeshConv/`

StructOpt-StandAlone_Opt : `cd Desktop/FilipeSilva/MSc/OptimisationStructOpt//StructOpt-StandAlone/StructOpt/StructOpt/FortranCodeOpt/`

Step 4: Compile the solution. The executable file produced is named "structopt".

```
ifort -o structopt dgemm.f dgemv.f dger.f dgetf2.f dgetrf.f dgetri.f
dgetrs.f dlamch.f dlaswp.f dscal.f dswap.f dtrmm.f dtrmv.f dtrsm.f
dtrti2.f dtrtri.f idamax.f ieeeck.f ilaenv.f iparmq.f lsame.f xerbla.f
ModuleGeometry.f90 ModuleMesh.f90 ModuleVariables.f90 ModuleBDF.f90
ModuleFEA.f90 ModuleFiles.f90 ModuleLoads.f90 ModuleMaterials.f90
ModuleOptimization.f90 ModuleSST.f90 module.f90 output.f90
maths_derivatives.f90 maths_interpolation.f90 maths_miscellaneous.f90
geometry_airfoil.f90 GeometryUtilitaries.f90 Geometry.f90 Mesh.f90
MeshUtilitaries.f90 BDFinterface.f90 Loads.f90 Material.f90 maths.f90
MathUtilitaries.f90 offset.f90 OptimizationInterface.f90 StructOpt.f90
```

Step 5: To perform an analysis, the executable file can be activated using the following command:

StructOpt-StandAlone_Solo : `sh -c /home/ubiaeronautics/Desktop/FilipeSilva/MSc/OptimisationStructOpt/StructOpt-StandAlone/StructOpt/StructOpt/FortranCodeMeshConv/structopt`

StructOpt-StandAlone_Opt : `sh -c /home/ubiaeronautics/Desktop/FilipeSilva/MSc/OptimisationStructOpt/StructOpt-StandAlone/StructOpt/StructOpt/FortranCodeOpt/structopt`

A.0.2 Run/Execution

The following instructions should be used to start a parallel optimization process of a given wing. The instructions presented in the following commands allow the execution in laboratory's computer. Before running optimization process, the user must compile the "StructOpt-StandAlone_Opt" and "StructOpt-StandAlone_Solo" since the executable files are needed to accomplish the subsequent steps (see Compilation A.0.1.1).

Step 1: To start, the user must define the geometry, mesh, loads, materials and optimization parameters (Table A.3: Files 1-7; 10-13 and Table A.2: Script 12). These files are located in the project's directory.
Step 2: A terminal must be open to execute the solo version of the StructOpt and to run the parallel optimization later.
Step 3: One execution of the "StructOpt - StandAlone_Solo" version is needed to generate the stacking sequence related files (Table A.3: Files 14-16; 21-24) and the grids/elements files (Table A.3: Files 17-20). The following command should be entered:

```
sh -c /home/ubiaeronautics/Desktop/FilipeSilva/MSc/OptimisationStructOpt/
StructOpt-StandAlone/StructOpt/StructOpt/FortranCodeMeshConv/structopt
```

Step 4: The workers can be generated automatically by executing the functions inside of the "workers_module.py" (Table A.2: Script 1). Each worker is a copy of the

"StructOpt_Opt" directory, folders and files (Table A.3: 1-7; 10-12; 14-24). The directories indicated in the original directories' file (File 12) are corrected to match the new folder. The StructOpt executable is also copied for each worker.

Step 5: After creating the workers' folders, the user must redirect to the folder in which the optimization script is located, using the terminal:

```
cd Desktop/FilipeSilva/MSc/OptimisationStructOpt/OptimisationStructOpt/
```

Step 6: To execute the optimization tool in parallel it is mandatory to first activate the conda local environment in which all the dependencies and parallel tools are installed using the following instruction:

```
conda activate structoptenv
```

Step 7: The user must create three blank files, properly named, in the desktop that will store part of the data associated with the optimization process (Table A.3: 25-27).

Step 8: To start a parallel run using 10 cores, the following instruction should be entered in the terminal:

```
mpiexec -n 10 pyhton3 MainOpt.py
```

Step 9: In the end of the optimization process, the user can plot the optimization's history and design variables values using the "PlotOptimisationHistory.py" script (Table A.2: Script 13).

A.0.3 Input files

To perform the structural analysis, a set of six input files should be defined. This section presents some examples of the input files used in this work, providing an idea on the structure and data of each one.

```
Reference x/c (x_ref)
0.25
Number of points to airfoil refinement (nop) (odd number)
257
Number of trailing edge divisions
20
Type of spacing of trailing edge divisions
1
Mesh type: 1 - Quadrilaterals, 2 - Triangles
2
```

Figure A.1: Input file structure and data of "Default.txt"

```
StructOpt - WingMesh
Main and rear spars
Type of spacing         Number of divisions             Spar
1                       27                              1       (MS)
1                       5                               2       (RS)
Cross section
Type of spacing         Number of divisions             Section
1                       0                               1       (LE) [Not used for wingbox]
1                       100                             2       (LS)
1                       100                             3       (US)
Panel longitudinal direction
Type of spacing         Number of divisions             Panel ID
1                       150                             1
```

Figure A.2: Input file structure and data of "WingMesh.txt"

```
StructOpt - WingGeom
x (c/4) y (c/4) z (c/4) Chord   Incidence       Spar position (x/c)     Rear spar position (x/c)        Airfoil
[mm]    [mm]    [mm]    [mm]    [deg]
00.0    0       0       300.0   0               0.25                    0.83                            1.0
00.0    375.0   0       300.0   0               0.25                    0.83                            1.0
```

Figure A.3: Input file structure and data of "WingGeom.txt"

```
Max. Number of layers per panel
Shell   MainSpar        RearSpar
10      10              3
Failure Theory (FT): TSAI-HILL(1); TSAI-WU(2); HOFF(3)
1               1                       1
Spar caps setting: 1 - Main spar only, 2 - Both spars
1
Spar caps max number of layers per panel
MS-Upper        MS-Lower        RS-Upper        RS-Lower
3               3               3               3
Spar caps width per panel [mm]
MS-Upper        MS-Lower        RS-Upper        RS-Lower
20              20              10              10
```

Figure A.4: Input file structure and data of "Structure.txt"

```
StructOpt - Materials Library
nMAT1           nMAT8-Cores   nMAT8-Fibers
1               3             1
MAT1            -             LINEAR ISOTROPIC
E[N/mm^2]       G[N/mm^2]     NU                rho[kg/m^3] CTE    TREF         GE           TA[N^/mm^2] CA[N/mm^2]   SA[N/mm^2]  LayerThickness [mm]
102900          4.9           0.27              1600        0      0            0            1064        805          35.47       0.15
MAT8            -             LINEAR ORTHOTROPIC FOR CORES
Assignment      Orientations number E1[N/mm^2]        E2[N/mm^2]  NU12   G12[N/mm^2]  G1Z[N/mm^2]  G2Z[N/mm^2]  rho[kg/m^3] A1          A2                TREF  Xt[N/mm^2]  Xc[N/mm^2]  Yt[N/mm^2]  Yc[N/mm^2]  S[N/mm^2]  GE    F12   STRN  LayerThickness [mm]
1               1             2650              39.75       0.49   98.05        143.1        13.25        150          0           0                 0     48.67       6           0.67        0.67        0.67       0     0     0     0.75
0
2               1             45                45          0.02   22           22           22           60           0           0                 0     0.87        0.6         0.87        0.6         0.57       0     0     0     5
0
3               1             66                66          0.1    30           30           30           80           0           0                 0     1.33        0.97        1.33        0.97        0.8        0     0     0     2.5
0
MAT8            -             LINEAR ORTHOTROPIC FOR FIBERS
Assignment      Orientations number E1[N/mm^2]        E2[N/mm^2]  NU12   G12[N/mm^2]  G1Z[N/mm^2]  G2Z[N/mm^2]  rho[kg/m^3] A1          A2                TREF  Xt[N/mm^2]  Xc[N/mm^2]  Yt[N/mm^2]  Yc[N/mm^2]  S[N/mm^2]  GE    F12   STRN  LayerThickness [mm]
4               4             53900             52500       0.06   4550         3570         2870         66           0           0                 0     449.4       420         399.47      420         33.13      0     0     0     0.08
0               15            30                45
```

Figure A.5: Input file structure and data of “MaterialsLibrary.txt”

Appendix B

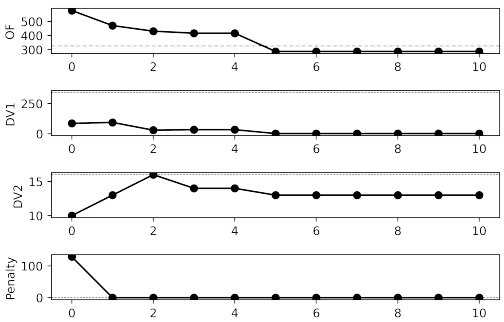
Optimization results

This appendix contains the results from all the optimization processes, and is organized by ascending order of mutation rate (from 0.03 to 0.15). For each combination of number of individuals and number of generations, the plots regarding the two optimization runs performed are presented.

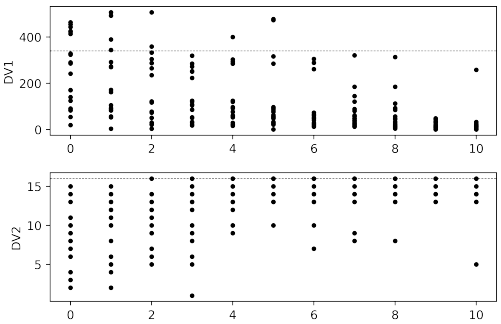
B.1 Mutation rate: 0.03

B.1.1 Population size: 20 elements

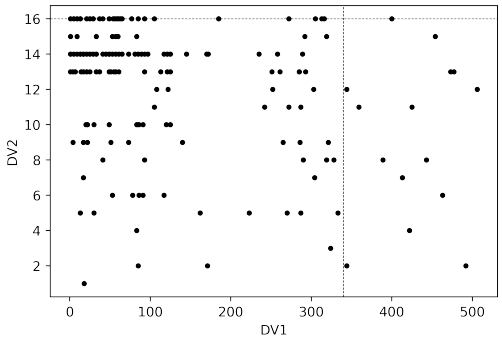
B.1.1.1 Number of generations: 10



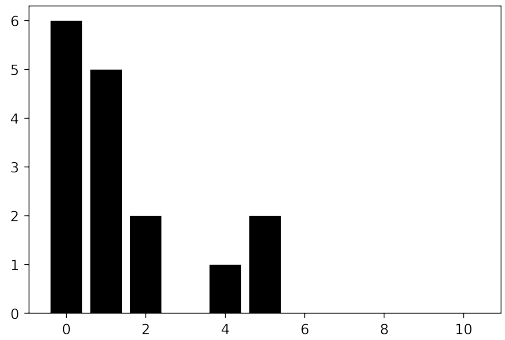
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

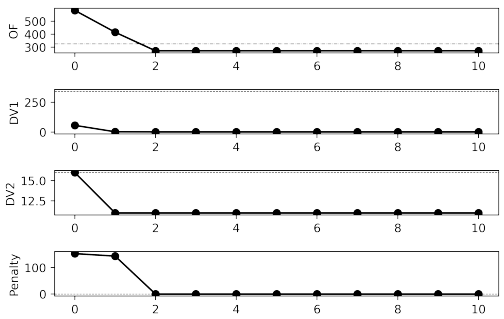


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

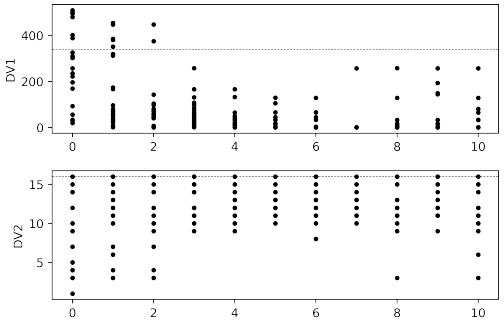


(d) Number of individuals per population with design variables out of bounds

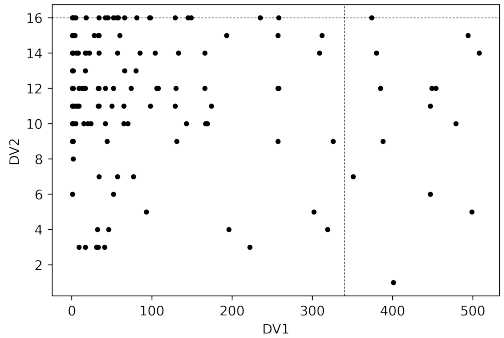
Figure B.1: Run 1 results: mutation rate of 0.03, population size of 20 elements, and number of generations of 10



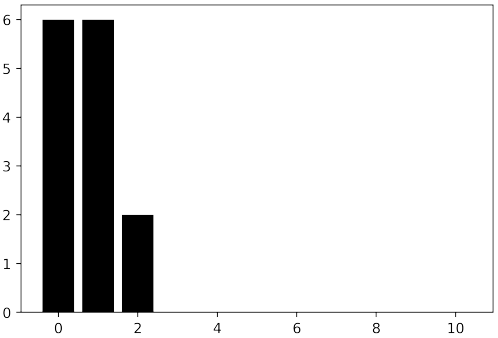
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



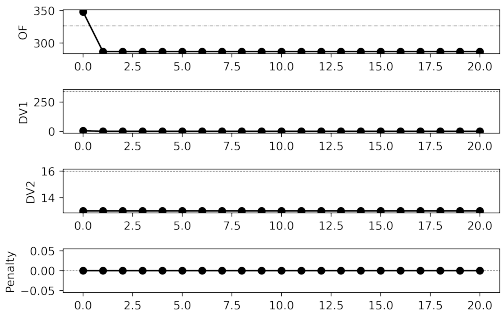
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



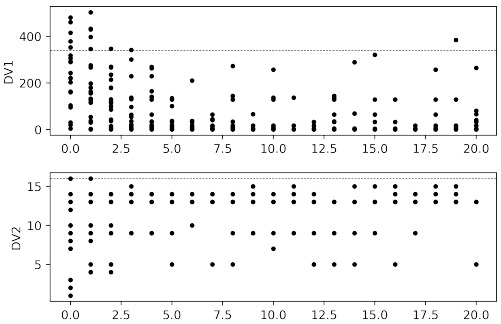
(d) Number of individuals per population with design variables out of bounds

Figure B.2: Run 2 results: mutation rate of 0.03, population size of 20 elements, and number of generations of 10

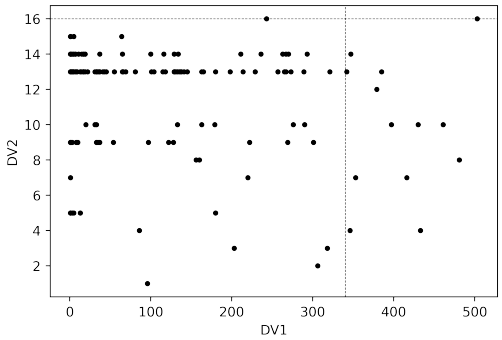
B.1.1.2 Number of generations: 20



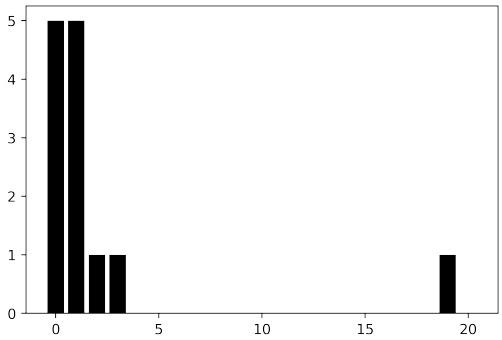
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

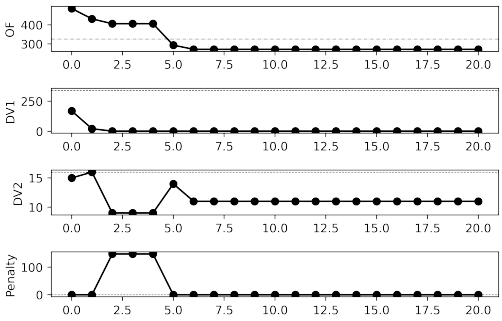


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

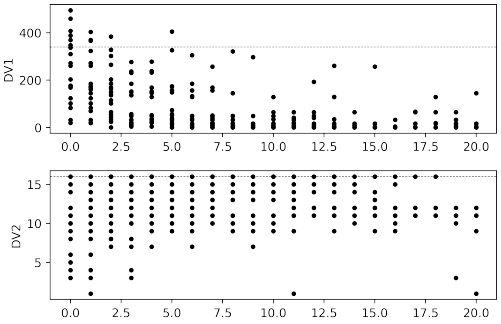


(d) Number of individuals per population with design variables out of bounds

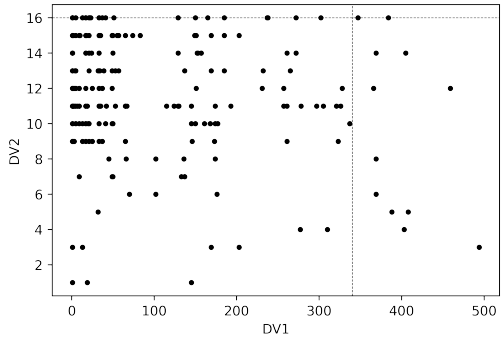
Figure B.3: Run 1 results: mutation rate of 0.03, population size of 20 elements, and number of generations of 20



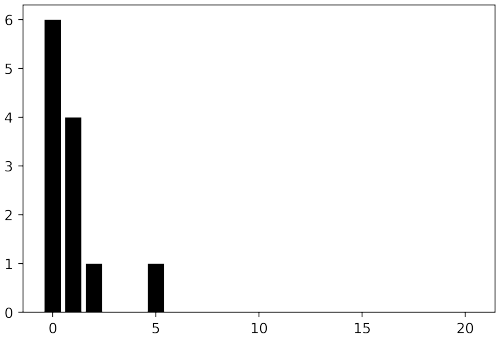
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits

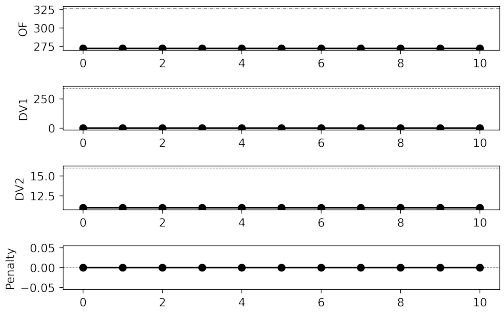


(d) Number of individuals per population with design variables out of bounds

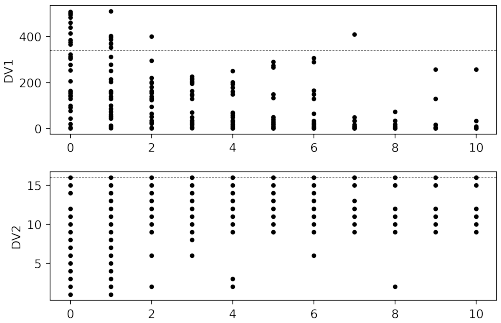
Figure B.4: Run 2 results: mutation rate of 0.03, population size of 20 elements, and number of generations of 20

B.1.2 Population size: 30 elements

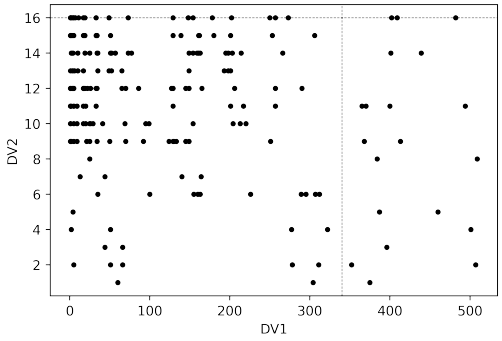
B.1.2.1 Number of generations: 10



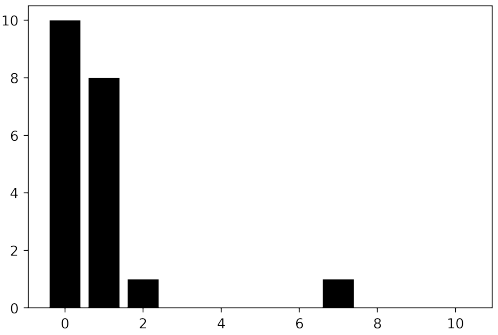
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

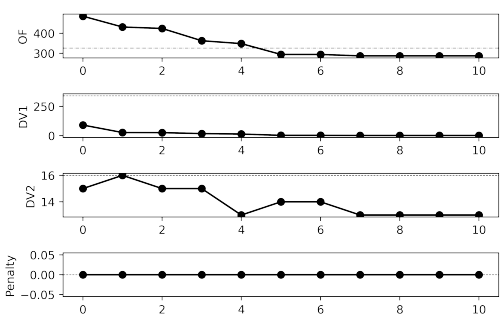


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

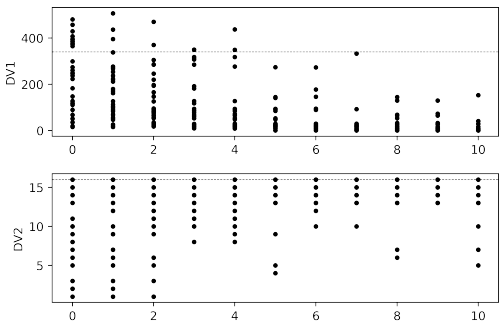


(d) Number of individuals per population with design variables out of bounds

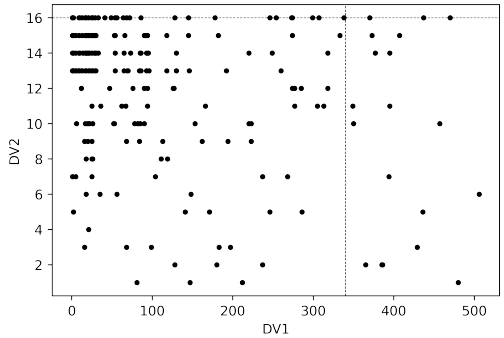
Figure B.5: Run 1 results: mutation rate of 0.03, population size of 30 elements, and number of generations of 10



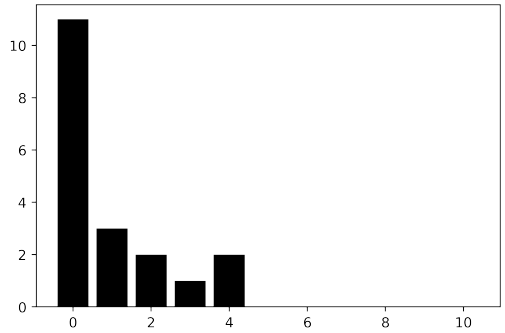
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



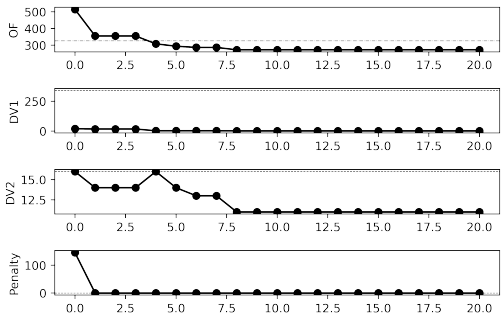
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



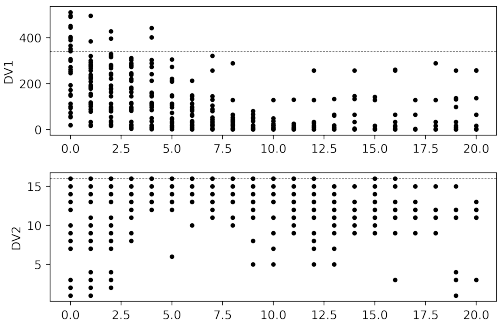
(d) Number of individuals per population with design variables out of bounds

Figure B.6: Run 2 results: mutation rate of 0.03, population size of 30 elements, and number of generations of 10

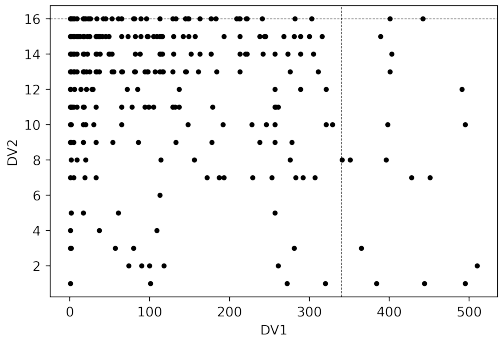
B.1.2.2 Number of generations: 20



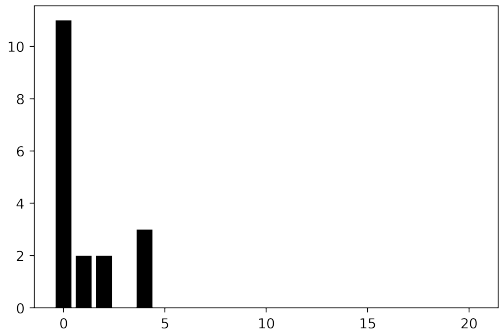
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

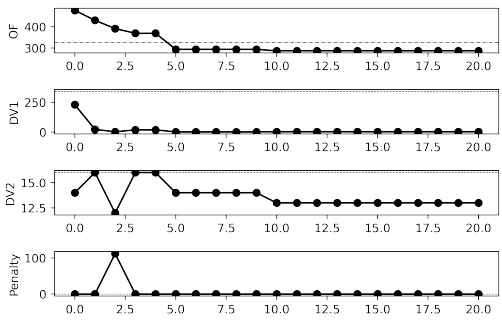


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

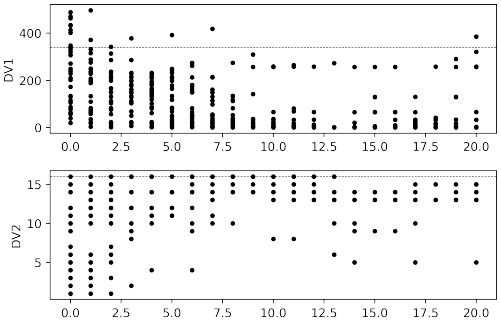


(d) Number of individuals per population with design variables out of bounds

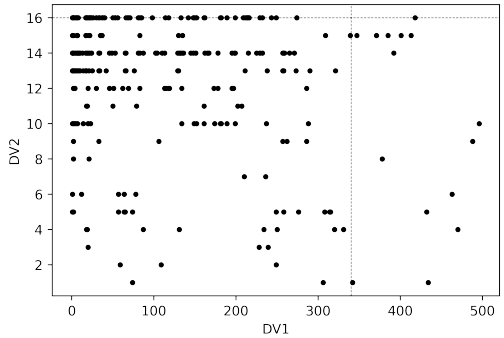
Figure B.7: Run 1 results: mutation rate of 0.03, population size of 30 elements, and number of generations of 20



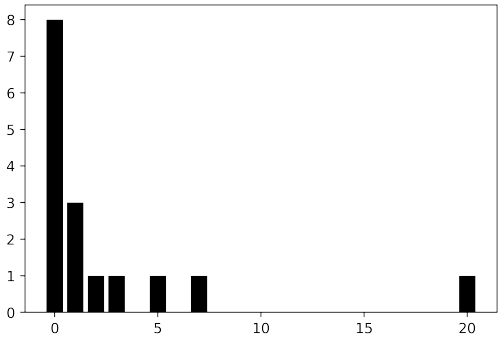
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits



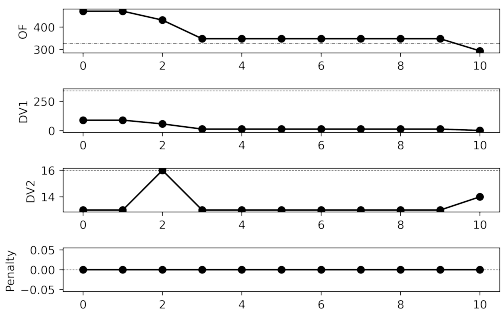
(d) Number of individuals per population with design variables out of bounds

Figure B.8: Run 2 results: mutation rate of 0.03, population size of 30 elements, and number of generations of 20

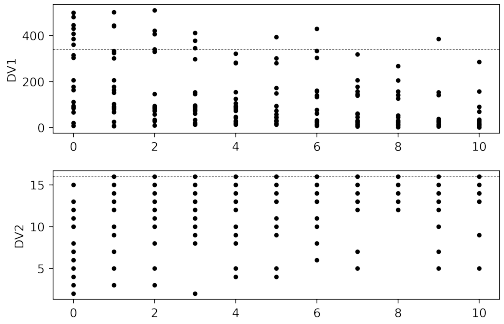
B.2 Mutation rate: 0.05

B.2.1 Population size: 20 elements

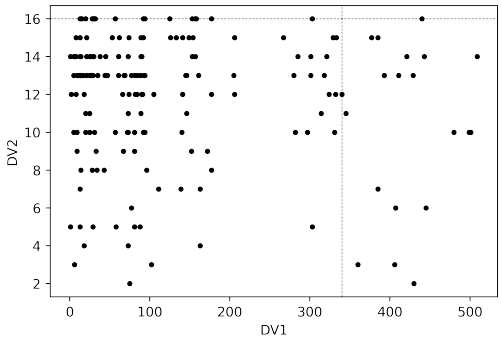
B.2.1.1 Number of generations: 10



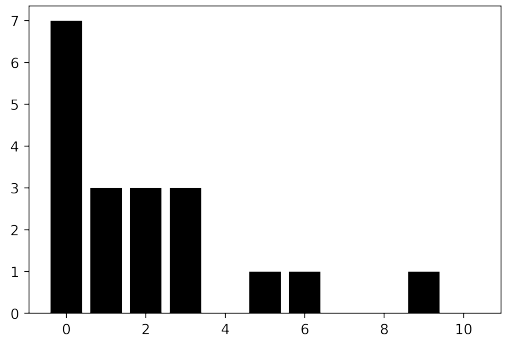
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

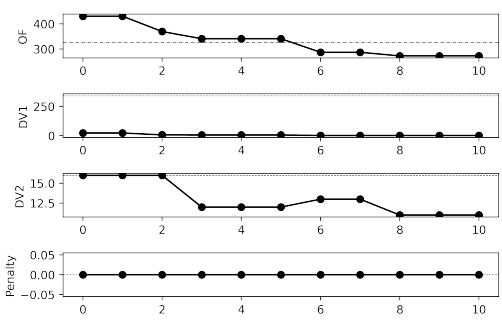


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

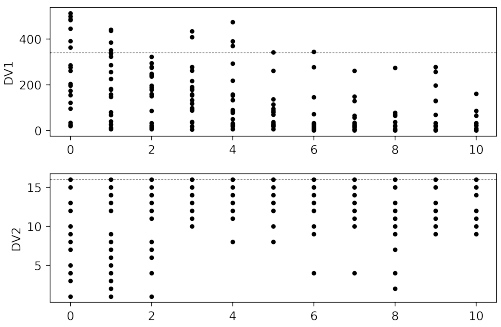


(d) Number of individuals per population with design variables out of bounds

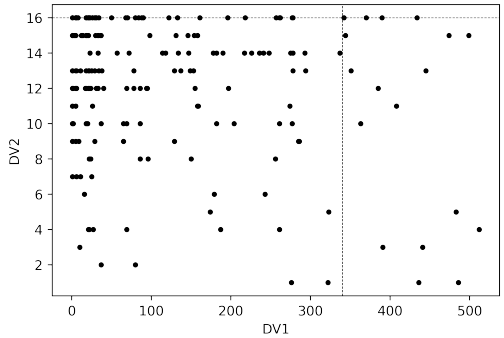
Figure B.9: Run 1 results: mutation rate of 0.05, population size of 20 elements, and number of generations of 10



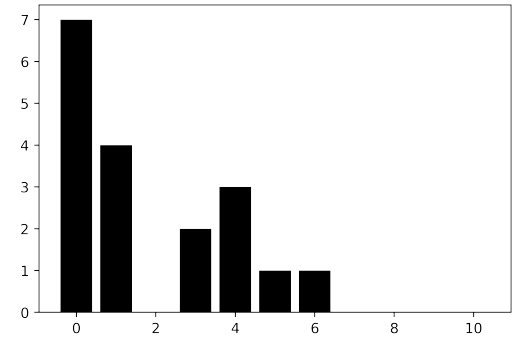
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



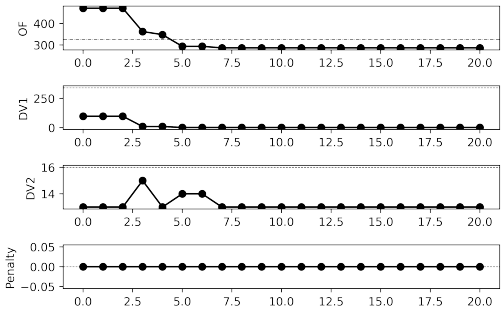
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



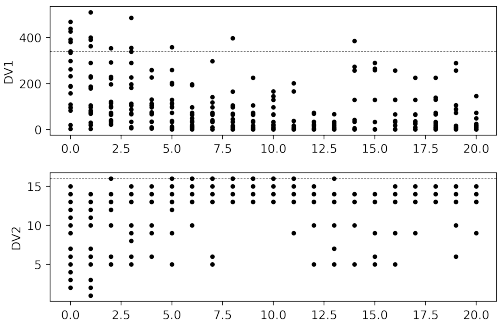
(d) Number of individuals per population with design variables out of bounds

Figure B.10: Run 2 results: mutation rate of 0.05, population size of 20 elements, and number of generations of 10

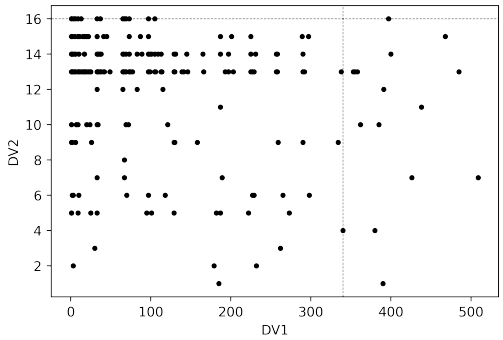
B.2.1.2 Number of generations: 20



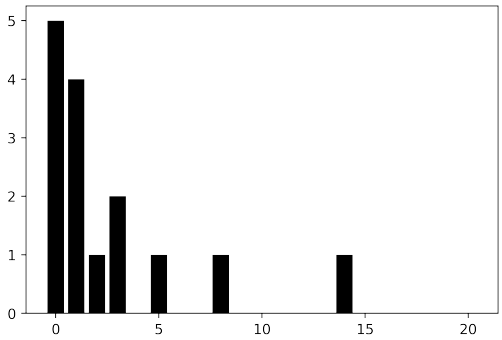
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

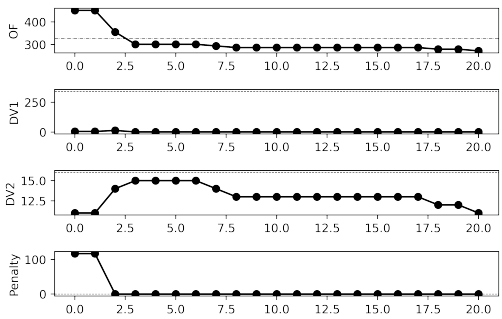


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

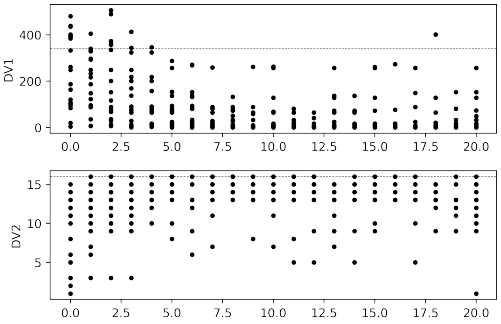


(d) Number of individuals per population with design variables out of bounds

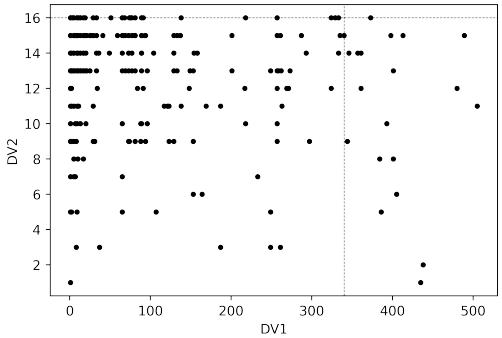
Figure B.11: Run 1 results: mutation rate of 0.05, population size of 20 elements, and number of generations of 20



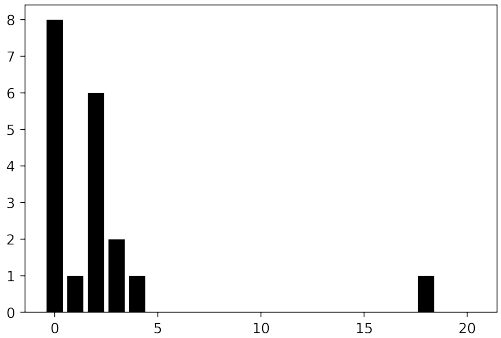
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits

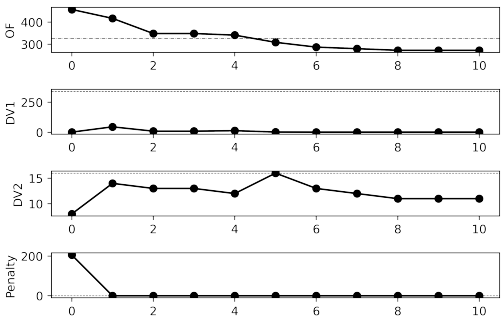


(d) Number of individuals per population with design variables out of bounds

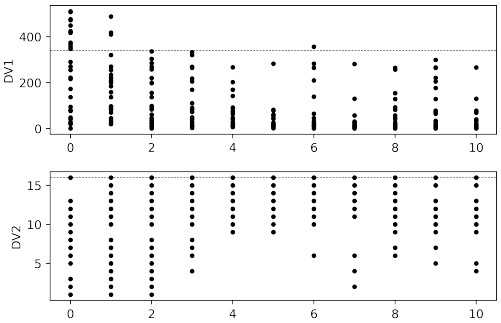
Figure B.12: Run 2 results: mutation rate of 0.05, population size of 20 elements, and number of generations of 20

B.2.2 Population size: 30 elements

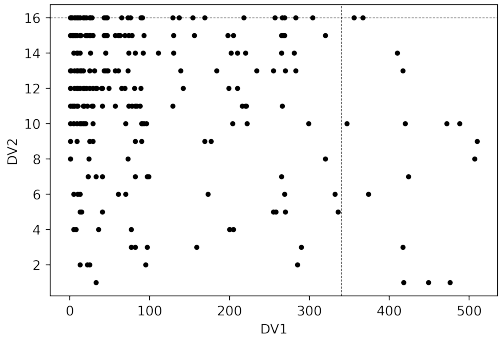
B.2.2.1 Number of generations: 10



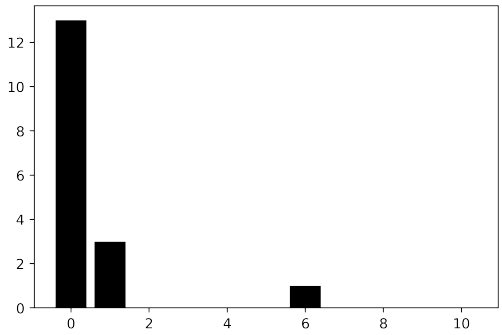
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

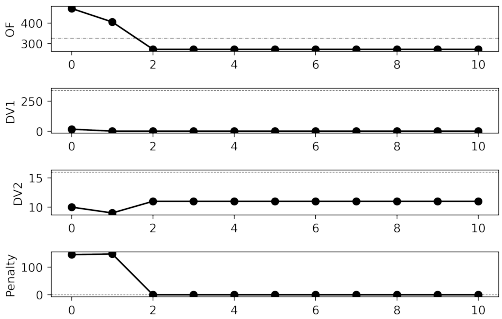


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

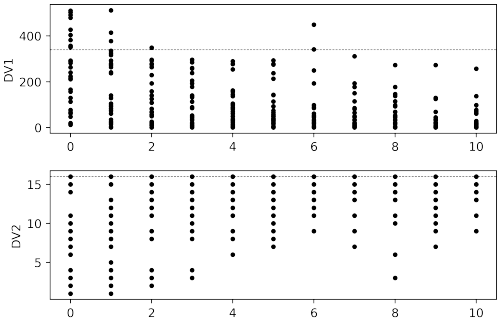


(d) Number of individuals per population with design variables out of bounds

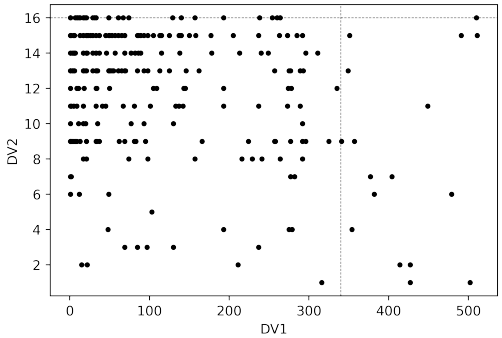
Figure B.13: Run 1 results: mutation rate of 0.05, population size of 30 elements, and number of generations of 10



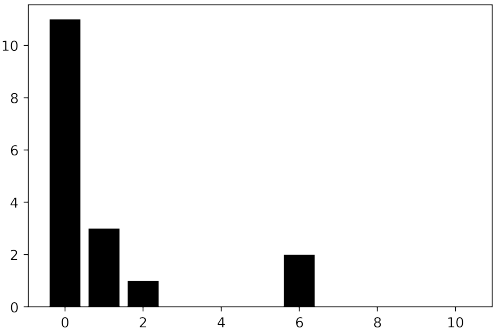
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



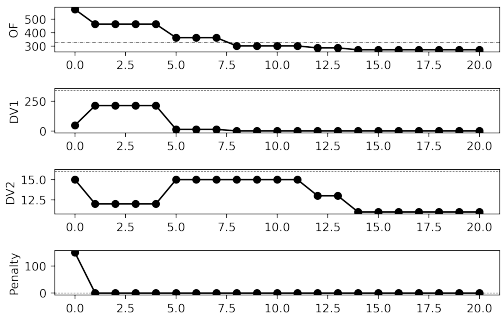
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



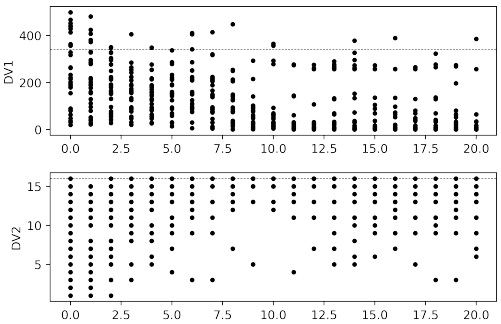
(d) Number of individuals per population with design variables out of bounds

Figure B.14: Run 2 results: mutation rate of 0.05, population size of 30 elements, and number of generations of 10

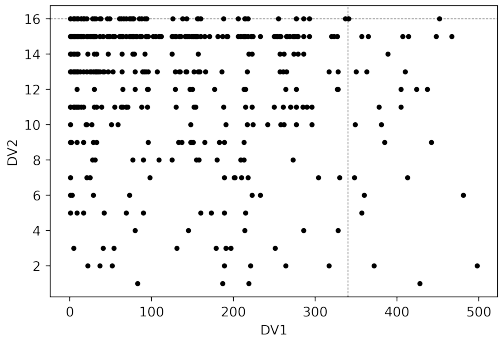
B.2.2.2 Number of generations: 20



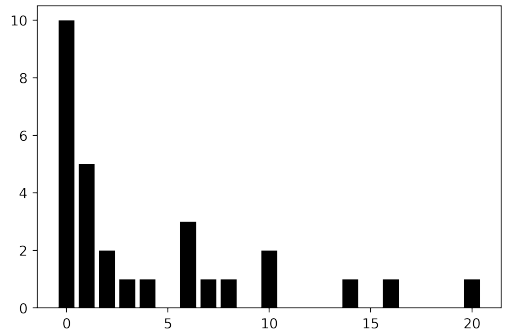
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

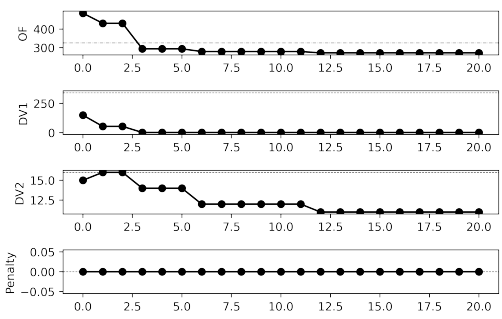


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

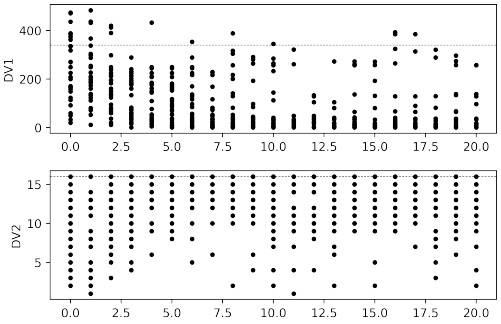


(d) Number of individuals per population with design variables out of bounds

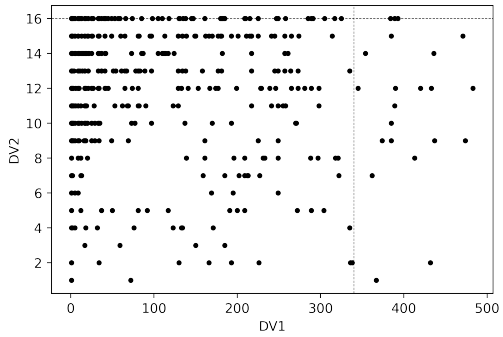
Figure B.15: Run 1 results: mutation rate of 0.05, population size of 30 elements, and number of generations of 20



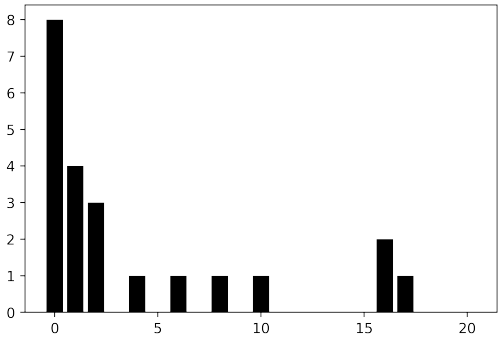
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits



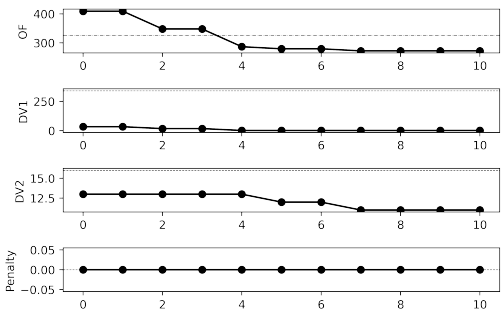
(d) Number of individuals per population with design variables out of bounds

Figure B.16: Run 2 results: mutation rate of 0.05, population size of 30 elements, and number of generations of 20

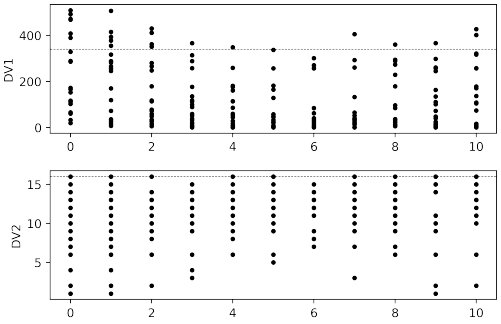
B.3 Mutation rate: 0.10

B.3.1 Population size: 20 elements

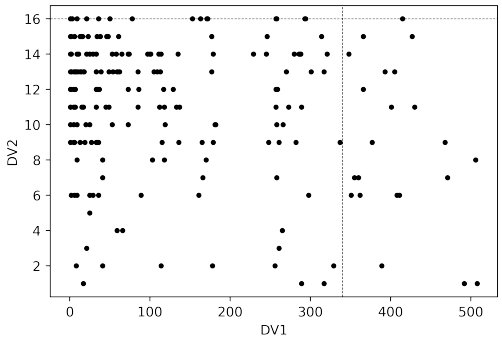
B.3.1.1 Number of generations: 10



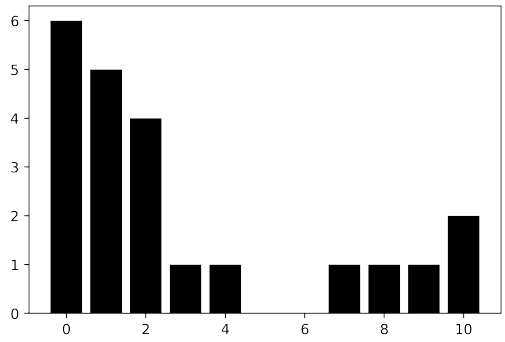
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

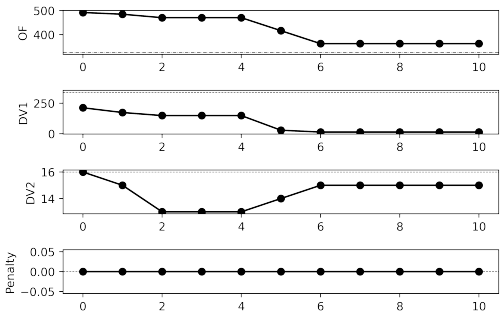


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

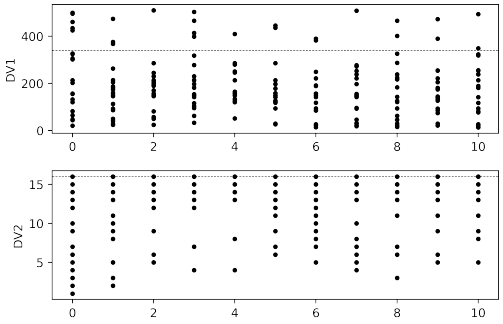


(d) Number of individuals per population with design variables out of bounds

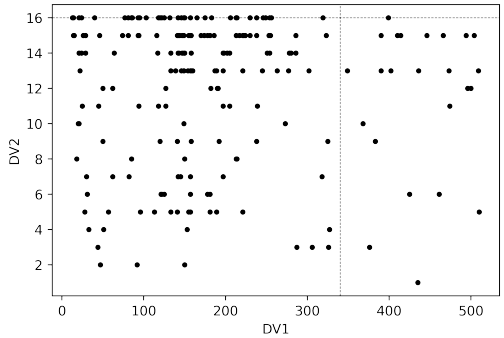
Figure B.17: Run 1 results: mutation rate of 0.10, population size of 20 elements, and number of generations of 10



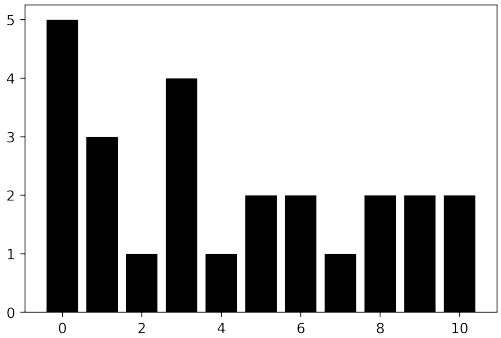
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



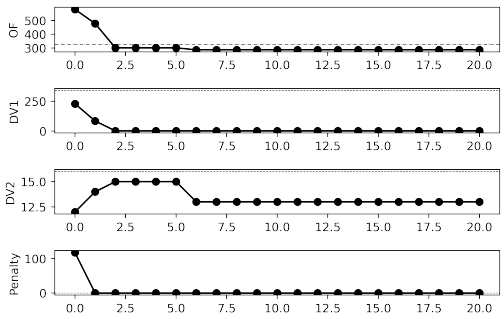
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



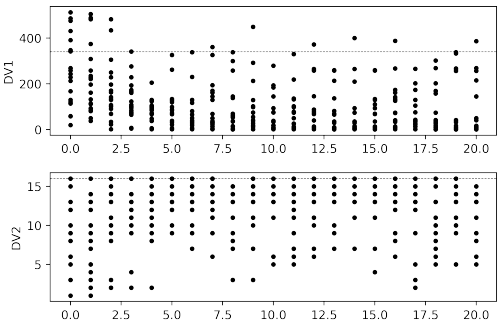
(d) Number of individuals per population with design variables out of bounds

Figure B.18: Run 2 results: mutation rate of 0.10, population size of 20 elements, and number of generations of 10

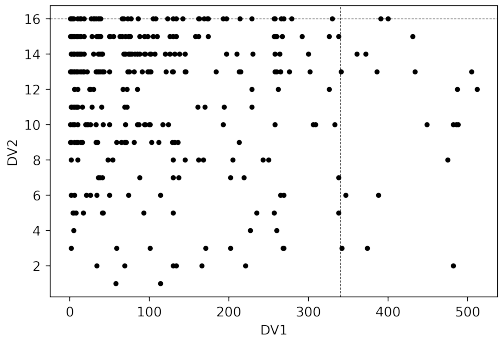
B.3.1.2 Number of generations: 20



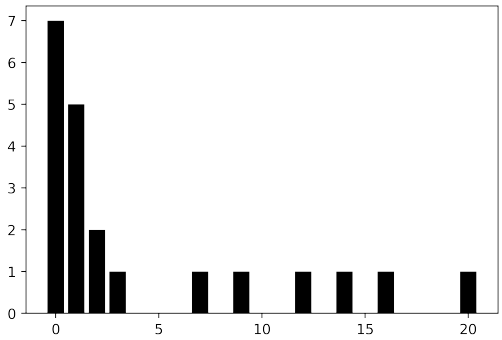
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

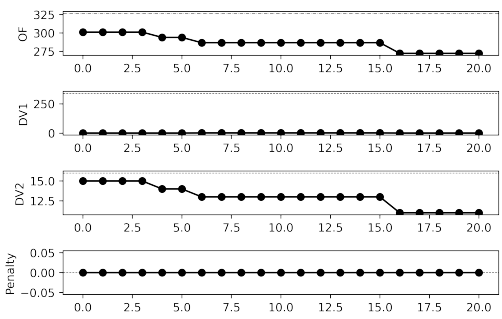


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

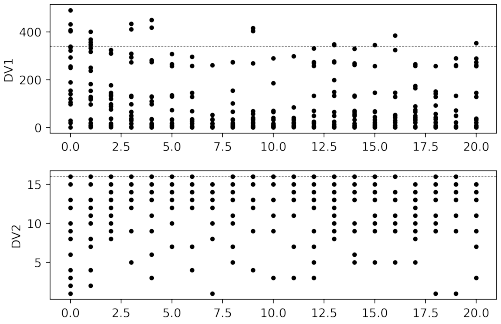


(d) Number of individuals per population with design variables out of bounds

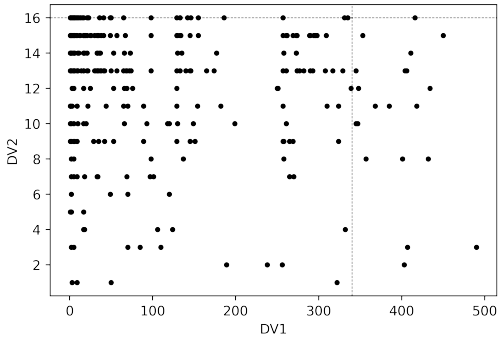
Figure B.19: Run 1 results: mutation rate of 0.10, population size of 20 elements, and number of generations of 20



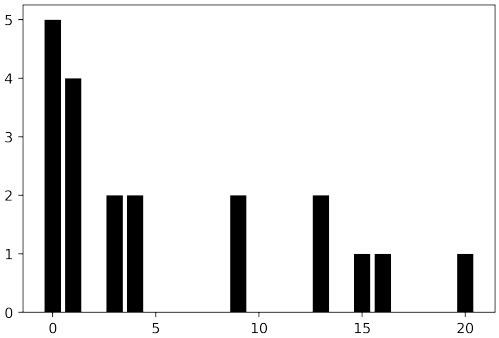
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits

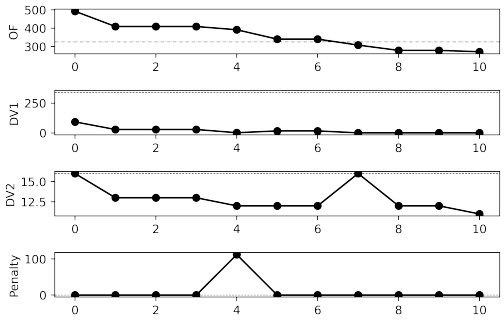


(d) Number of individuals per population with design variables out of bounds

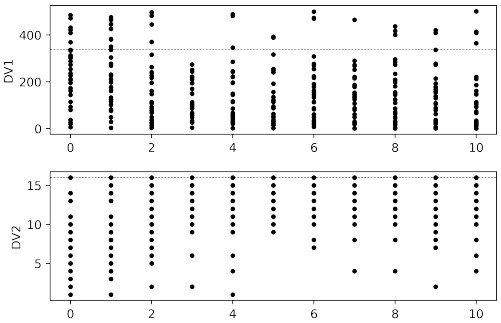
Figure B.20: Run 2 results: mutation rate of 0.10, population size of 20 elements, and number of generations of 20

B.3.2 Population size: 30 elements

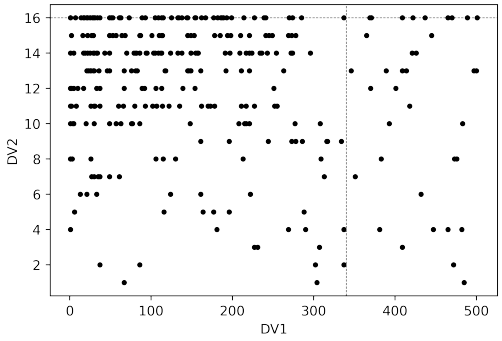
B.3.2.1 Number of generations: 10



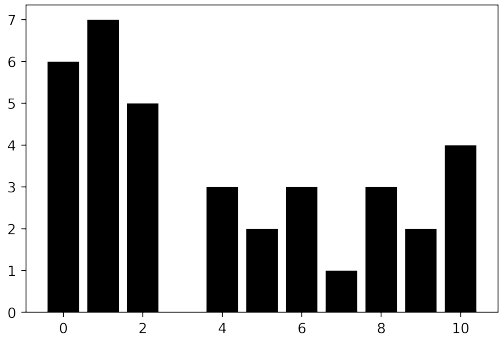
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

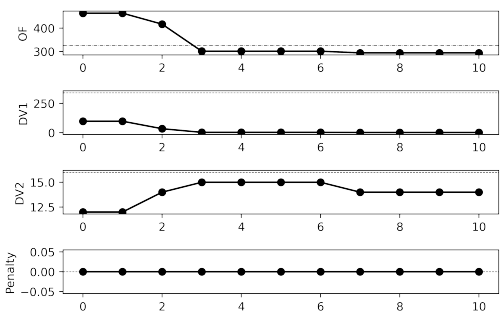


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

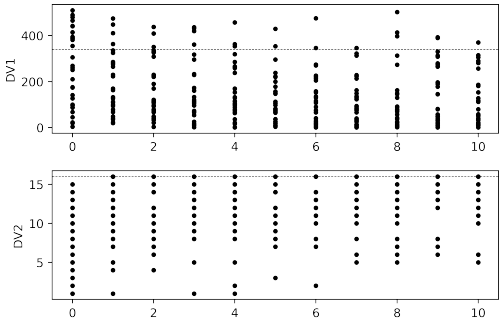


(d) Number of individuals per population with design variables out of bounds

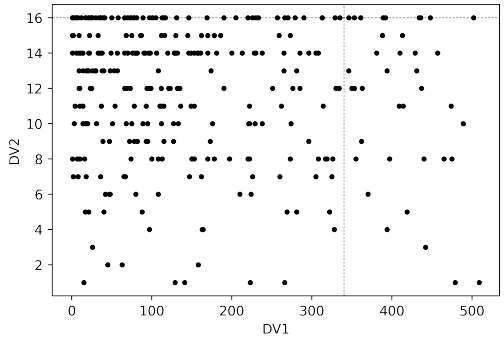
Figure B.21: Run 1 results: mutation rate of 0.10, population size of 30 elements, and number of generations of 10



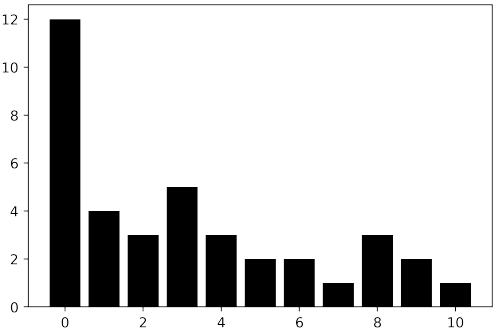
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



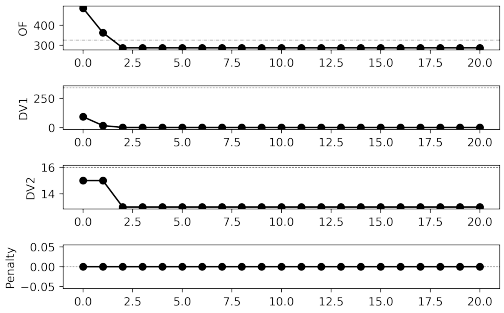
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



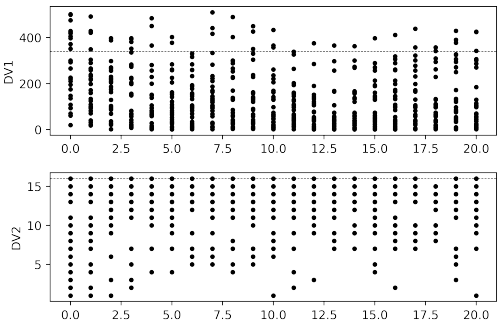
(d) Number of individuals per population with design variables out of bounds

Figure B.22: Run 2 results: mutation rate of 0.10, population size of 30 elements, and number of generations of 10

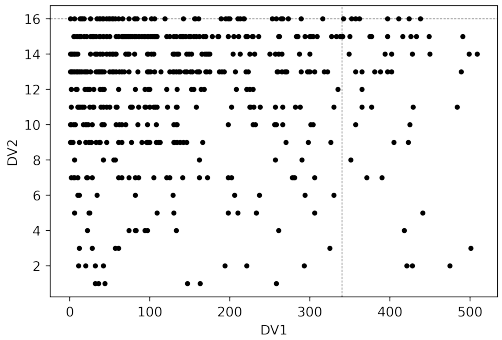
B.3.2.2 Number of generations: 20



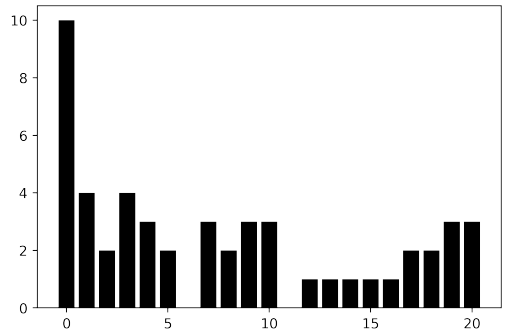
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

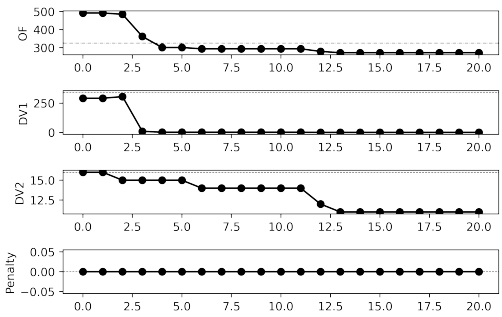


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

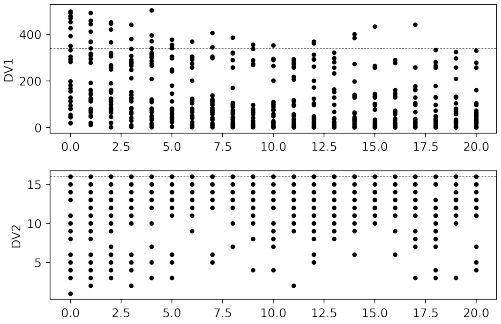


(d) Number of individuals per population with design variables out of bounds

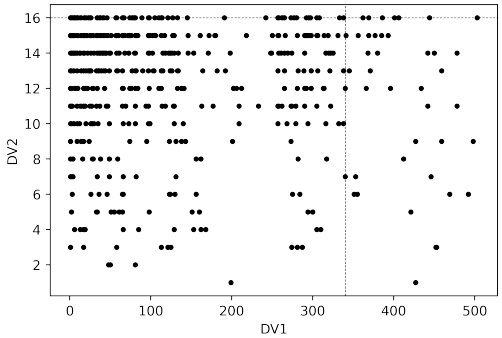
Figure B.23: Run 1 results: mutation rate of 0.10, population size of 30 elements, and number of generations of 20



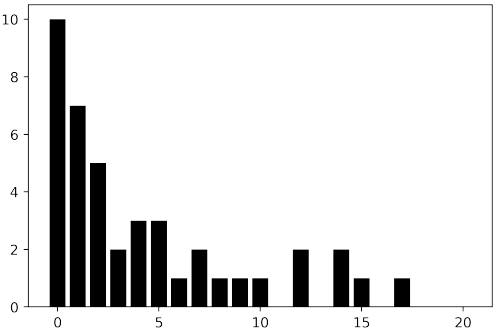
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits



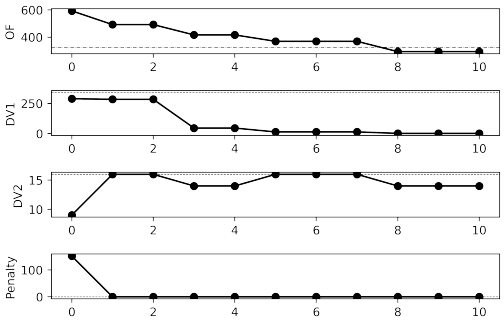
(d) Number of individuals per population with design variables out of bounds

Figure B.24: Run 2 results: mutation rate of 0.10, population size of 30 elements, and number of generations of 20

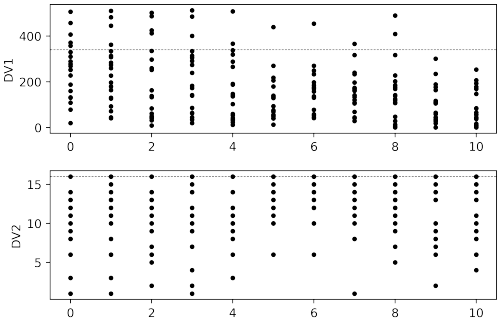
B.4 Mutation rate: 0.15

B.4.1 Population size: 20 elements

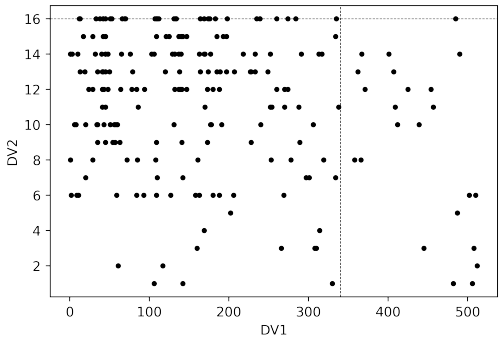
B.4.1.1 Number of generations: 10



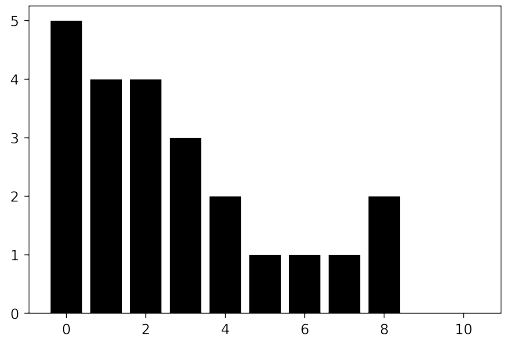
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

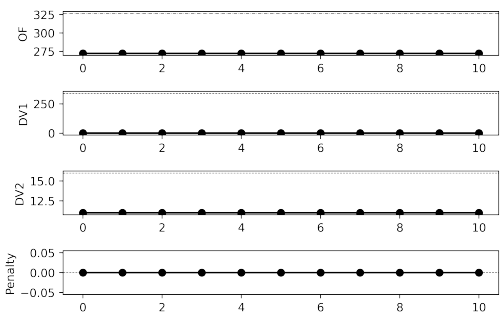


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

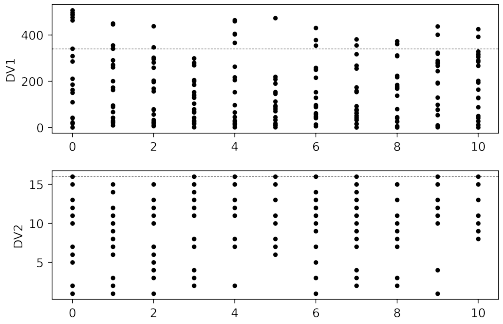


(d) Number of individuals per population with design variables out of bounds

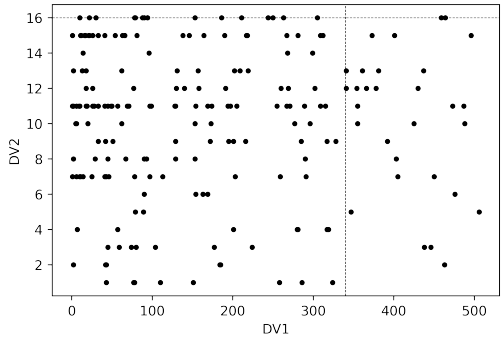
Figure B.25: Run 1 results: mutation rate of 0.15, population size of 20 elements, and number of generations of 10



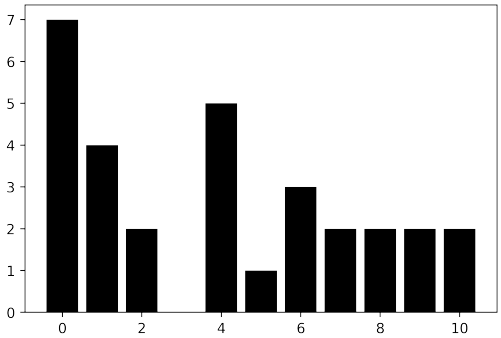
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



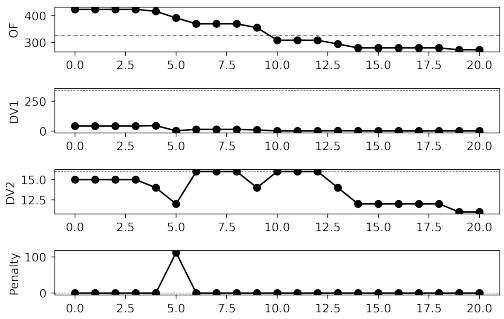
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



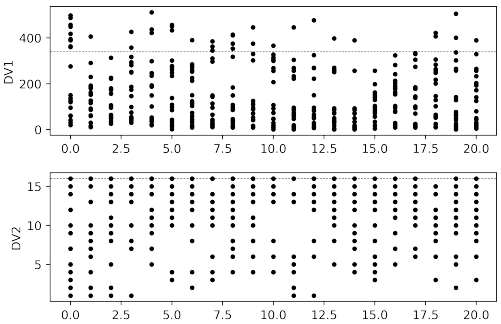
(d) Number of individuals per population with design variables out of bounds

Figure B.26: Run 2 results: mutation rate of 0.15, population size of 20 elements, and number of generations of 10

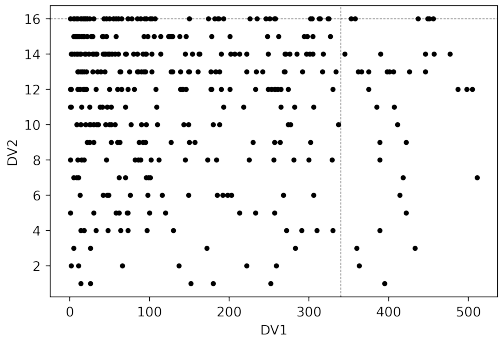
B.4.1.2 Number of generations: 20



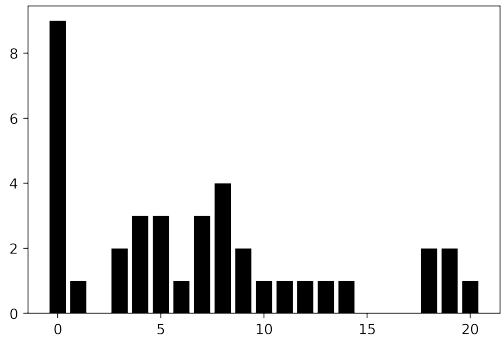
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

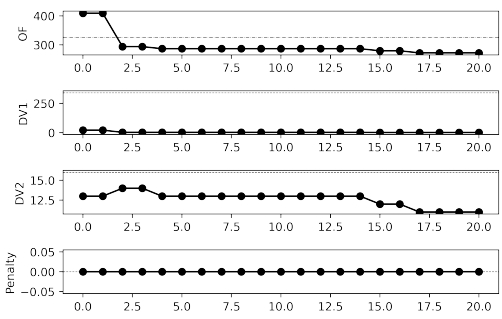


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

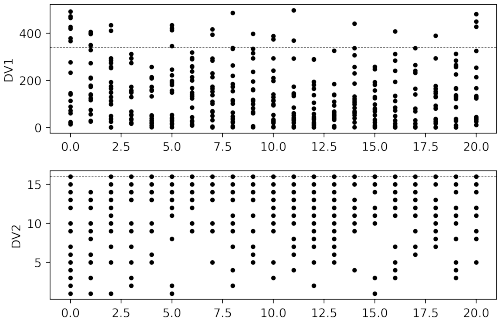


(d) Number of individuals per population with design variables out of bounds

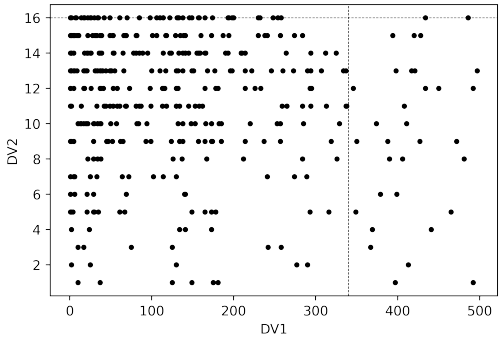
Figure B.27: Run 1 results: mutation rate of 0.15, population size of 20 elements, and number of generations of 20



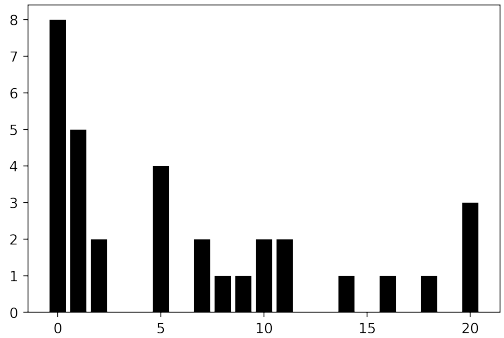
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits

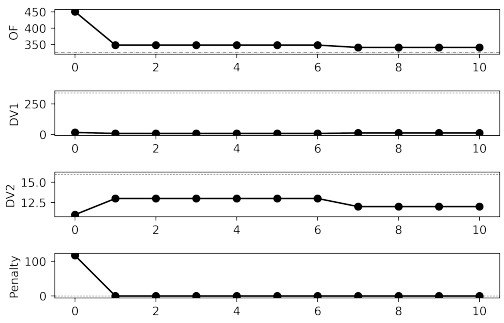


(d) Number of individuals per population with design variables out of bounds

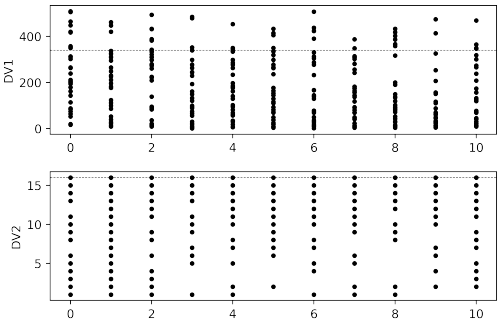
Figure B.28: Run 2 results: mutation rate of 0.15, population size of 20 elements, and number of generations of 20

B.4.2 Population size: 30 elements

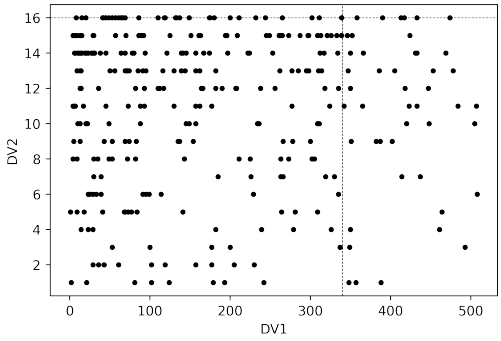
B.4.2.1 Number of generations: 10



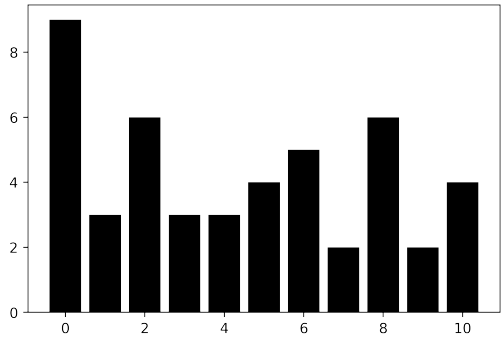
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

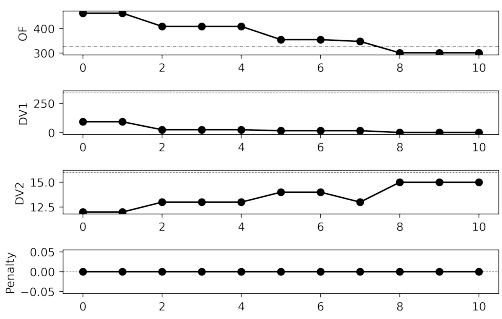


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

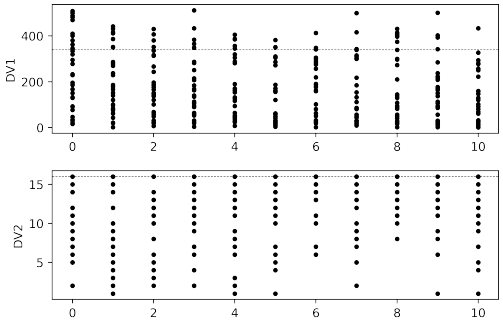


(d) Number of individuals per population with design variables out of bounds

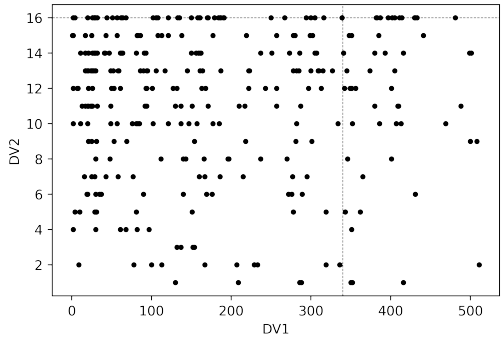
Figure B.29: Run 1 results: mutation rate of 0.15, population size of 30 elements, and number of generations of 10



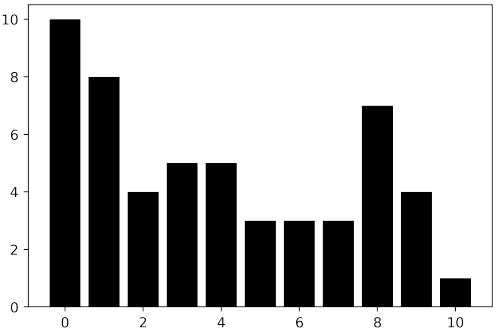
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



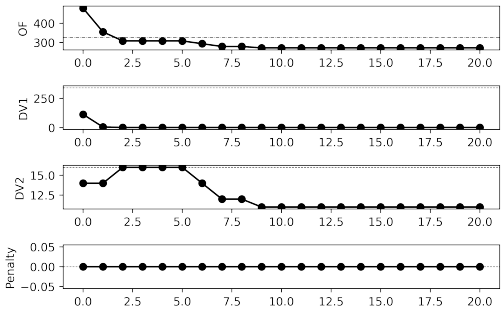
(c) DV1 vs DV2, dashed lines correspond to design variables' limits



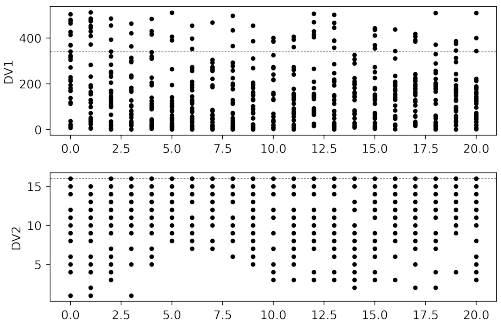
(d) Number of individuals per population with design variables out of bounds

Figure B.30: Run 2 results: mutation rate of 0.15, population size of 30 elements, and number of generations of 10

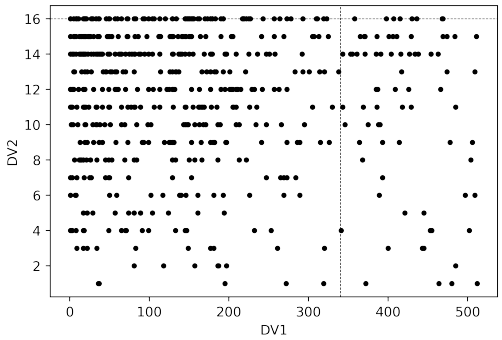
B.4.2.2 Number of generations: 20



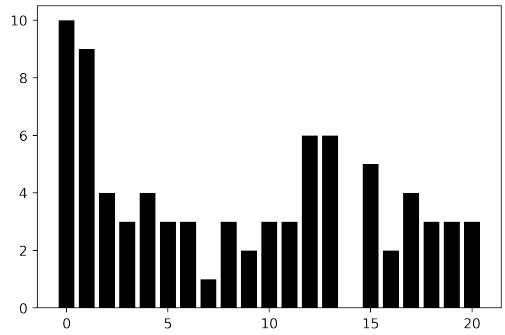
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits

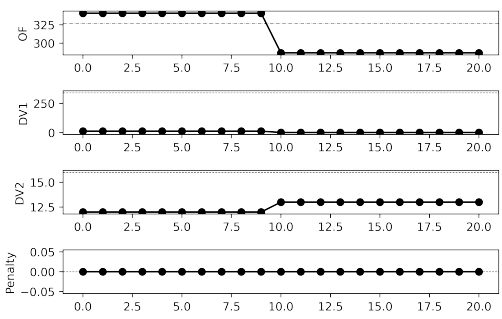


(c) DV1 vs DV2, dashed lines correspond to design variables' limits

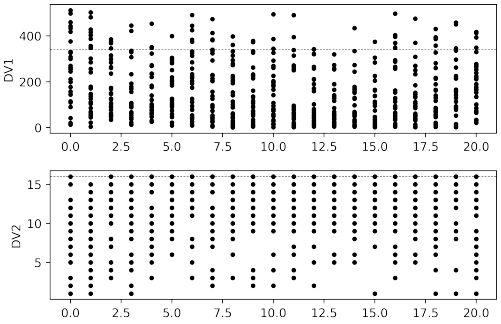


(d) Number of individuals per population with design variables out of bounds

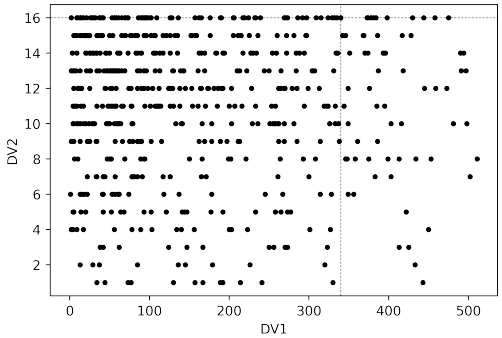
Figure B.31: Run 1 results: mutation rate of 0.15, population size of 30 elements, and number of generations of 20



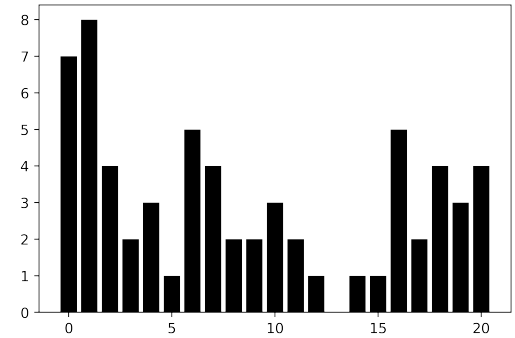
(a) History of fittest individual in the population: objective function, design variables, and penalty



(b) Design variables values per population, dashed lines correspond to design variables' limits



(c) DV1 vs DV2, dashed lines correspond to design variables' limits



(d) Number of individuals per population with design variables out of bounds

Figure B.32: Run 2 results: mutation rate of 0.15, population size of 30 elements, and number of generations of 20