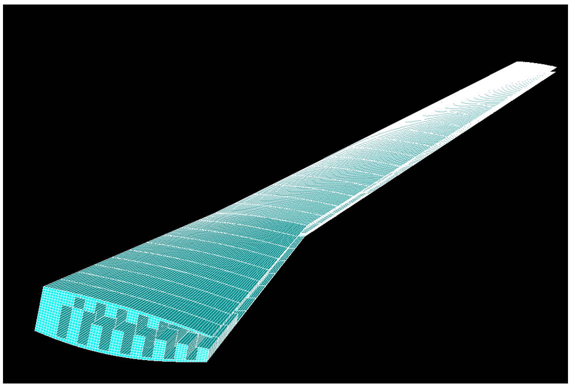


Design of a High Aspect Ratio Wing Box using an Ant Colony Optimization Algorithm

Ricardo João Araújo Diogo

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor(s): Prof. Afzal Suleman
Prof. André Calado Marta

Examination Committee

Chairperson: Prof. Fernando José Parracho Lau
Supervisor: Prof. Afzal Suleman
Member of the Committee: Doutor José Lobo do Vale

April 2016

To my parents, Tiago and brothers, for their unconditional support...

Acknowledgments

I would like to thank my thesis supervisors, professor Afzal Suleman and professor André Marta. The first one for giving me the opportunity of conducting this thesis in the University of Victoria, Canada, providing all the conditions necessary to my work. The second one for the guidance and support given during the last months.

I would also like to thank to Mário Brás for all the guidance given during the realization of my thesis and all the discussions that we had in order to find the solutions for the problems that I was encountering.

To Rasoul Sohouli, thank you for all the support and for all the suggestions regarding Topology Optimization.

To José Vale I would like to thank the help given during the CFD implementation.

The last but not the least, I would like to my family, especially my mother that helped getting through the most difficult parts of this academic journey, always believing in my work. Without my family nothing that I have achieved would have been possible.

Resumo

Todas as aeronaves que percorrem os céus actualmente usam combustíveis fósseis para gerar a propulsão necessária para voarem, o que resulta na produçao de dióxido de carbono (CO_2) e outros gases com efeito de estufa que prejudicam o ambiente. Tendo este problema em mente, esforços no desenvolvimento de novas aeronaves têm sidos feito de forma a aumentar a eficiência destas. Uma das soluções que se encontra a ser testada actualmente para aumentar a eficiência é a introdução de asas com uma razão de aspecto elevada. Asas com uma elevada razão de aspecto apresentam vários problemas estruturais, sendo necessário a introdução de uma *wing box* suficientemente rígida de forma a resolvê-los. Tal solução levará a um aumento do peso da asa, reduzindo as vantagens obtidas pela introdução desta, o que torna o projecto da *wing box* fundamental. De forma a obter a estrutura o mais rígida possível com o mínimo peso possível, uma optimização topológica da mesma deve ser feita. O principal objectivo desta tese é encontrar o projecto óptimo para a *wing box* fazendo uma optimização topológica utilizando o algoritmo *Ant Colony Optimization* (ACO) como optimizador. ACO é um algoritmo meta-heurístico com influências biológicas já utilizado com sucesso para resolver problemas de optimização combinatorial de uma forma simples. A sua aplicação em problemas de optimização topológica foi desenvolvida recentemente. Conclusões sobre as topologias óptimas do *wing box* para uma asa com razão de aspecto elevado são apresentadas. Nesta tese, um código de MATLAB® com uma interface com o ANSYS® foi desenvolvido, para resolver um problema de optimização topológica. Começou-se por resolver um caso de exemplo presente na literatura. Seguidamente, a *wing box* da asa do projecto NOVEMOR da EU foi optimizada.

Palavras-chave: Métodos heurísticos, Optimização topológica, Asas com elevada razão de aspecto

Abstract

Every aircraft currently crossing the sky are using fossil fuels in order to generate the propulsion required to fly, which leads to the production of carbon dioxide (CO_2) and other greenhouse gases that warm the environment. Having this in mind, the main drive for the next generation of aircraft is fuel efficiency. One of the solutions that is being tested to increase the fuel efficiency is the introduction of high aspect ratio (AR) wings. A high aspect ratio wing leads to many structural problems, and to solve them one needs to have a considerably stiff wing box. The given solution increases the weight of the wing and reduces the advantages brought by the introduction of a high AR wing, thus making the design of the wing box a fundamental aspect. In order to have the stiffest possible structure having the minimum possible weight, one can do a topology optimization to find the optimal design for the given constraints. The main goal of this thesis is to find the optimal design of the wing box by performing a topology optimization using Ant Colony Optimization (ACO) algorithm. ACO is a meta-heuristic biologically influenced algorithm that has been proven to be useful to solve NP-hard combinatorial optimization problems in an expedite way. Its application to solve topology optimization has been introduced recently. Conclusions on the close to optimal topology of a wing box from a high aspect ratio wing are presented. In this thesis, a MATLAB® code integrated with ANSYS® for the structural analysis, is developed, in order to do a topology optimization. One will start by solving a literature case. Afterwards, the wing box of the NOVEMOR project of the EU 7^{th} framework wing was optimized.

Keywords: Heuristic method, Topology optimization, High aspect ratio wings.

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

α	Pheromone influence parameter.
Γ	Dihedral angle.
ρ	Evaporation rate.
ν	Poisson's Ratio.
Λ	Sweep angle.
λ	Strain energy influence parameter.
β	Visibility information influence parameter.

Roman symbols

AoA	Angle of attack.
AR	Aspect Ratio.
c_b	Break chord.
D	Density.
P	Pressure.
c_r	Root chord.
a	Speed of sound.
b	Span.
c_t	Tip chord.
T	Temperature.
$\mathbf{u}$	Velocity vector.
u, v, w	Velocity Cartesian components.
V	Velocity module.

E	Young’s Modulus.

Subscripts

x, y, z	Cartesian components.
∞	Free-stream condition.
ref	Reference condition.

Glossary

ACO Ant Colony Optimization is a optimization algorithm used to solve NP-hard combinatorial problems.

CFD Computational Fluid Dynamics is a branch of fluid mechanics that uses numerical methods and algorithms to solve problems that involve fluid flows.

FCS Flight Control System consists of the devices and systems which govern the attitude of an aircraft.

FEM Finite element method is a general numerical technique which allows the determination of approximate solutions for the boundary value problems.

GB Gradient based algorithms are optimization algorithm which rely in the gradient of the objective function to find the optimal solution for the optimization process.

MAC Mean Aerodynamic Chord is a two dimensions representation of the whole wing, allowing to reduce the pressure distribution over the whole wing to a single lift force and a moment around the aerodynamic center of the mean aerodynamic chord.

MTOW Maximum Take-off Weight is the maximum weight at which the pilot is allowed to take off.

TO Topology Optimization is a mathematical approach that optimizes material layout within a given design space, for a given set of loads and boundary conditions such that the resulting layout meets a prescribed set of performance targets.

TSP Travelling Salesman Problem is a combinatorial problem where one must complete an Hamiltonian tour.

Chapter 1

Introduction

In every action that one takes, one tries to find the optimal way to do it, from the quickest route to work, in order to avoid traffic, to a given mathematical application in which one is trying to find an optimum of a function. This concept is extremely important in our life, mainly due to every resource being finite, limited in some sense. These resources can be time, money or a natural resource, to name a few. Bearing that in mind one needs to find the optimal way to use these finite resources, being this the main reason that one has to use optimization.

Optimization can be applied to many different areas from the most straightforward one as mathematics and economics to information technology. For instance, one may need to improve the performance of a given code, being the performance measured in running time or the quantity of resources that a given computer program requires to run properly. Across all of these areas, one has as the main goal to use a given resource/s in the most efficient way [Yang, 2014].

1.1 Historical Background

The beginning of modern aviation can be stated to occur on December 17 of 1903 [Jr., 1985]. This day has a special meaning for the modern aviation because it corresponds to the first flight of the Wright brothers, being the first time that a powered heavier-than-air aircraft was able to fly. From this point on, this historical mark lead to the construction of some aircraft by people that had some curiosity in flying and spare time to have a side project, being none of the developed aircraft built in order to fulfil a necessity.

In 1914, when the World War I started, the use of aircraft in Warfare was seen as a way of gaining superiority over the enemy, having an important tactical and strategical role. With the end of the World War I, there were more developments in aircraft technologies.

The first aircraft being powered by a turbojet engine was the Heinkel He 178, making its first flight on 27 August of 1939. From this point on, many developments were made and aircraft started to have a major importance in many aspects of ours life.

The increase in number of flights, resulting from an ever increasing demand, has led to an estimate of 2.3 billion passengers and 38 million tonnes of freight on scheduled services, making a total of 531 billion tonne kilometre [ICAO, 2010] crossed by aircraft in 2010. These numbers are also translated in the economic impact that aviation has, generating roughly €220 billion and providing 4.5 million jobs [EU, 2011].

It is expected that the number of passengers will increase at rate of 4.8% per year through the years until 2036 [ICAO, 2010]. However, due to the restrictions imposed in CO_2 emissions, and other greenhouse gases and knowing that liquid fuels availability is finite, the development of the new generation of aircraft has as the main drive fuel efficiency and how one can one decrease the harm done by aviation to the environment.

1.2 High Aspect Ratio Wings

One of the trends in aircraft development in order to increase fuel efficiency is the introduction of high aspect ratio wings.

The aspect ratio is defined as being the wing's span divided by the average or mean chord length [Anderson and Eberhardt, 2001]. Knowing that the area of the wing is equal to span times the mean chord the AR can be defined as

$$AR = \frac{b}{\bar{c}} = \frac{b^2}{S}. \tag{1.1}$$

The AR indicates the relation between the longitudinal dimension of the wing (b) and the width of the wing ($\bar{c}$). If an aircraft has an $AR = 8$, it means that the span is 8 times bigger than the mean aerodynamic chord. A comparison between the AR of different aircraft can be seen in Table 1.1.

Aircraft Type	Aspect Ratio
Personal	$5.0 - 8.0$
Commuter	$9.0 - 12.0$
Regional Turboprop	$11.0 - 12.8$
Business Jet	$5.0 - 8.8$
Jet Transport	$7.0 - 9.5$
Military Fighter/Attack	$2.4 - 5.0$

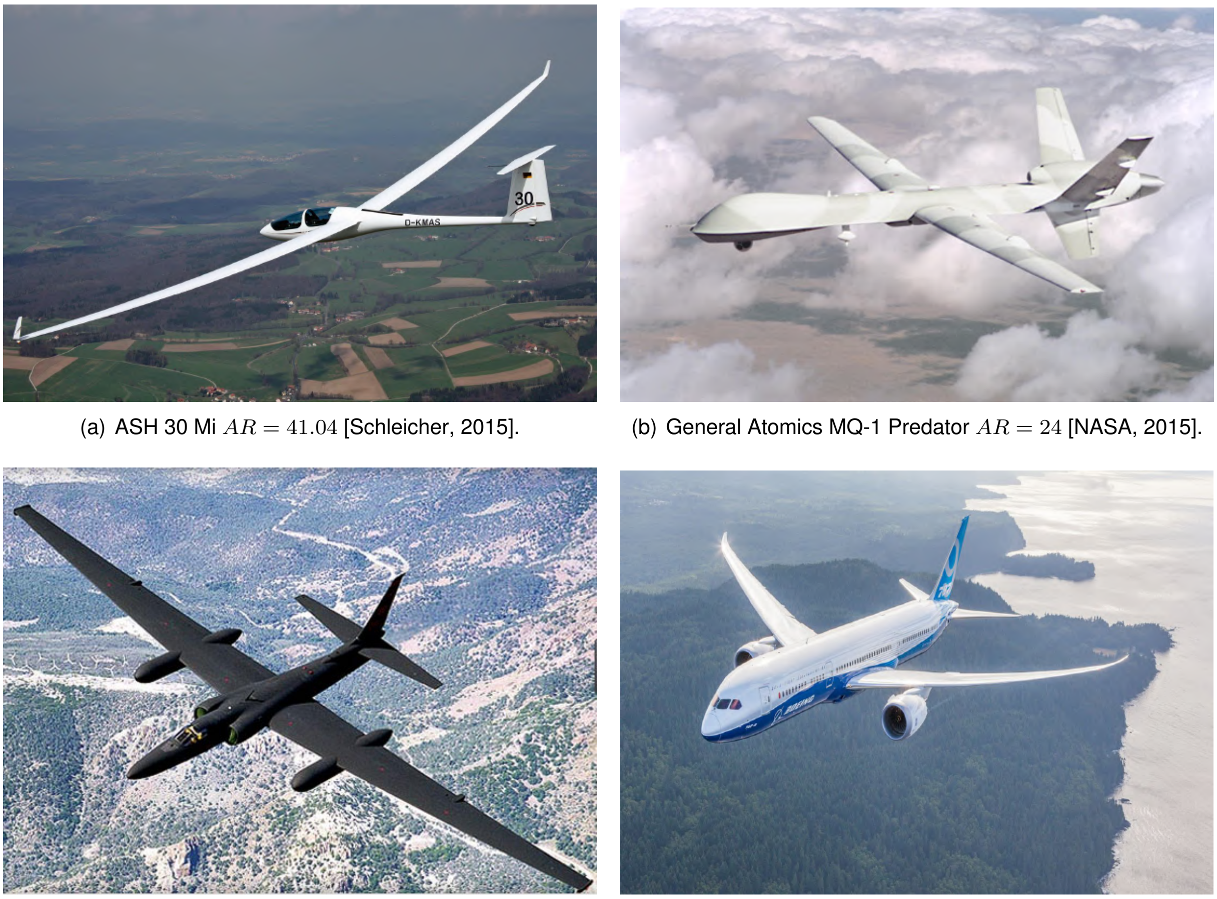
Table 1.1: Aspect ratio of different types of aircraft.[Corke, 2003]

The AR will have an influence in the performance of the aircraft, therefore, depending on the type of mission that the aircraft as to fulfil, a different AR will be selected.

Many aircraft have been developed using a high AR wing. Some examples of a successful implementation of high AR wings can be seen in Figure 1.1.

The first example, the ASH 30-Mi (Figure 1.1(a)) is a glider manufactured by Alexander Schleicher. A glider is defined by the Federal Aviation Administration as ”an heavier-than-air aircraft that is supported in flight by the dynamic reaction of the air against its lifting surface, and whose free flight does not depend principally on an engine” [FAA, 2013].

The ASH 30-Mi has span equal to 26.5 m, a fuselage height of 9.33 m and a glide ratio higher than 60 [Schleicher, 2015]. The glide ratio is defined as the ratio of distance that the glider travels horizontally



(a) ASH 30 Mi $AR = 41.04$ [Schleicher, 2015].

(b) General Atomics MQ-1 Predator $AR = 24$ [NASA, 2015].

(c) Lockheed U-2 $AR = 10.67$ [USAF, 1959].

(d) Boeing 787 Dreamliner $AR = 11$ [Boeing, 2015].

Figure 1.1: Examples of aircraft with high AR wing.

for every unit of altitude lost,

$$\text{Glide ratio} = \frac{\text{Lift}}{\text{Drag}}. \tag{1.2}$$

Since one wants to have the highest glide ratio as possible, one will want to maximize the lift to drag ratio.

The Drag coefficient of the wing ($C_{D_{wing}}$) is due to the contributions of the friction drag, profile or thickness drag, induced drag, interference drag and compressibility drag. One can separate the zero-lift drag coefficient of the wing ($C_{D_{0_w}}$), from the the drag coefficient due to the lift, also called induced drag coefficient ($C_{D_{L_w}}$) [Roskam and Lan, 1997],

$$C_{D_{wing}} = C_{D_{0_w}} + C_{D_{L_w}}, \tag{1.3}$$

being $C_{D_{L_w}}$ given by [Roskam and Lan, 1997],

$$C_{D_{L_w}} = \frac{C_{L_w}^2}{\pi \cdot e \cdot AR}. \tag{1.4}$$

As one can see the $C_{D_{L_w}}$ is dependent on the aspect ratio of the wing. Keeping the wetted area constant and increasing the AR, one can decrease the $C_{D_{L_w}}$, reducing the overall drag of the wing. The introduction of a wing with a higher AR is mainly based on this fact.

After the glider has been launched one wants to fly for the longest time possible. A glider will convert the potential energy accumulated during the launch into kinetic energy. In order for this trade to be as efficient as possible one needs to keep the resistance (drag) as low as possible, which explains the introduction of a high AR wing.

The second aircraft, the GA MQ-1 Predator (Figure 1.1(b)), is an Unmanned Aerial Vehicle (UAV) produced by General Atomics. It is a medium altitude, long endurance UAV to use in risky areas in order to minimize casualties. The MQ-1 Predator role is reconnaissance, surveillance and target acquisition, displaying a flight endurance up to 24 hours and a maximum range of 400 NM [NASA, 2015]. The MQ-1 Predator being a long endurance UAV, needs to sustain flight for a long time. Being the engine of the MQ-1 Predator a four-cylinder piston engine, for a reciprocating engine, one can use,

$$E = \frac{\eta}{C}\frac{L}{D}\frac{1}{V}\ln\left[\frac{W_i}{W_f}\right], \tag{1.5}$$

to explain the introduction of a high AR wing. For the same engine and weight fraction the endurance will be maximum for the maximum lift-to-drag ratio.

The third aircraft, the Lockheed U-2 (Figure 1.1 (c)) was manufactured by Lockheed Martin from 1959 to 1989, it is a single place, jet propelled aircraft designed to do long range operations at very high altitude (70000 ft)[USAF, 1959].

In Figure 1.2, the operating radius, which corresponds to half of the range of the aircraft, is given as a function of the wing's AR. The increasing AR will lead to an increasing Operating Radius.

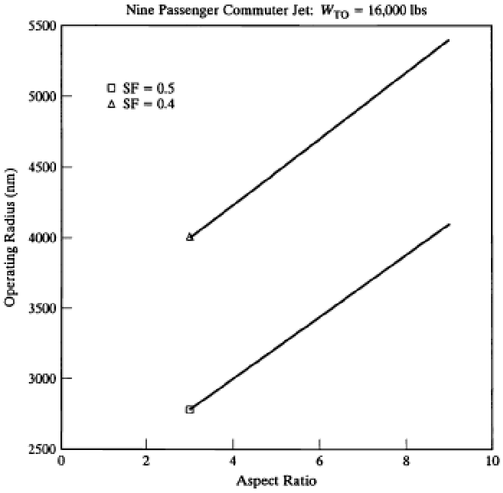


Figure 1.2: Operating radius as a function of the AR [Corke, 2003]

The U-2 is a reconnaissance aircraft, used to gather intelligence regarding the enemy, so it requires a long range in order to fulfil its mission, the range of the given aircraft is 5500 NM. In order to achieve this range the introduction of a high AR wing was fundamental.

The last example, is the Boeing 787-8 Dreamliner (Figure 1.1 (d)). It is produced by Boeing, its commercial service started on October 2011.

In Figure 1.3, the evolution of the wing AR of commercial aircraft throughout the years can be seen.

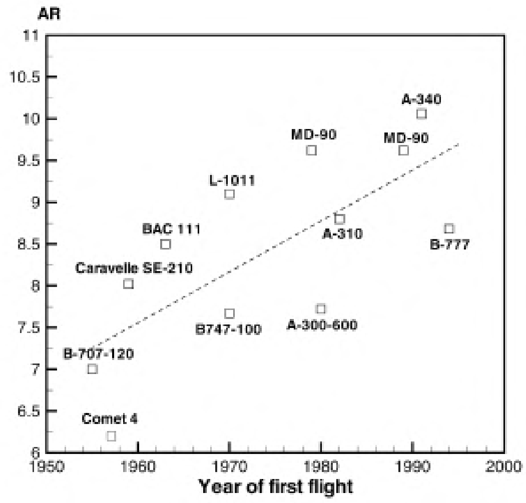


Figure 1.3: Evolution of the wing AR of the commercial aircraft [Filippone, 2000]

The trend displayed in figure 1.3 of the wing AR is clearly of increasing the AR. As stated before, a higher AR will lead to a lower induced drag, which will save fuel, increasing the fuel efficiency of the aircraft. Being a commercial aircraft, its main purpose is to transport passengers or cargo in the most efficient way, meaning that the fuel efficiency is really important during the design of the aircraft.

1.3 High AR Wing Drawbacks

The introduction of a high AR wing it is not always easy to accomplish, leading to some problems, such as, structural, aeroelastic and control related ones. Usually a high AR wing is really slender which will result in a increase of flexibility. Due to its flexibility, it will have large deflections when subjected to aerodynamic loads. The deflection in some aircraft can reach as much as 25% of the wing semi-span [Patil and Hodges, 2004].

An example of a high AR wing which presented aeroelastic problems is the NASA Helios prototype. The Helios prototype (Figure 1.4) is a remotely piloted flying wing aircraft developed under the ERAST project. This aircraft was built having in mind two goals, demonstrate sustained flight at an altitude near 100,000 ft and flying non-stop for at least 24 hours, including at least 14 hours above 50,000 ft. The aircraft had a wingspan of 75 m and a wing area of 183.6 m^2, which resulted in an AR equal to 30.9. On 13 August 2001, the beginning of the flight tests, Helios reached an altitude of 96,863 ft, breaking the world record for sustained horizontal flight by a winged aircraft.

On 26 of June of 2003, during a flight test the Helios began to experience airspeed excursions of around ± 2 ft/sec and showing a high dihedral for the speed the current flight speed, as seen in figure 1.4. The high dihedral persisted which led to a pitch oscillation built up [Noll et al., 2004]. In order to decrease the wing dihedral, the airspeed was increased, leading to a small decrease of the dihedral growing past 30 ft afterwards.

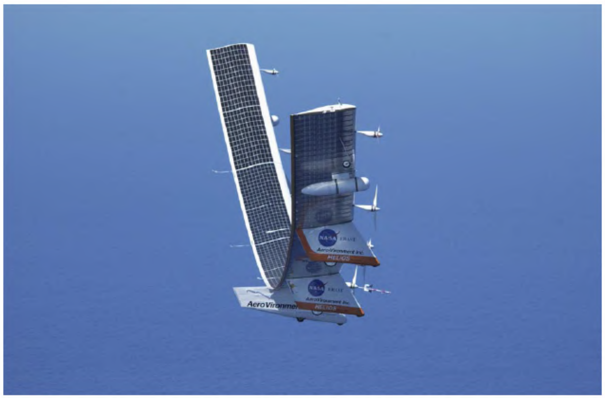


Figure 1.4: NASA Helios Prototype at high wing dihedral

The aircraft was under large airspeed fluctuations, which indicated that was experiencing large pitching motions. The amplitude of these pitching motion was doubling after every cycle. Helios started pitching down reaching an airspeed approximately **2.5** times de maximum design airspeed. These extreme conditions led to the large aerodynamic load which broke the leading edge foam sections of the right wing near the hydrogen fuel tanks, ripping the solar cells and skin off the upper surface of the wing [Noll et al., 2004].

One of the causes for the Helios crash was due to the a persistent high dihedral, which caused some instability. The persistence of the high dihedral was stated to be caused by turbulence and aeroelastic/flight control system factors. The wing twist and lift loss due to elevator deflection was caused by the aeroelastic characteristics of the aircraft and FCS. It was also reported that the wing's aeroelastic response to perturbations would increase the wing dihedral and sustain it.

The aeroelastic response of the wing was not modelled correctly, due to the use of a linear analysis which did not capture the non-linearities of the the response for small gust amplitudes. Also, the ”complexity and the interactions between the aeroelastic and stability modes made it difficult to apply time domain stability and control analysis to the vehicle” [Noll et al., 2004]. Combining this two difficulties led to relying primarily on static stability and frequency domain analyses, which do not give information about the time relationship between the wing twist and dihedral.

The non-linearities of the wing and the instabilities are mainly due to the introduction of a high AR wing, which produced a large elastic structure. At the given time there were not well established techniques or procedures to conduct vibrations tests to accurately calibrate the analytical tools [Noll et al., 2004].

The Helios prototype displayed the aeroelastic problems that arise from high AR wings, however other problems emerge from its introduction. High AR wings in some cases might not be practical, mainly in commercial aircraft, in which a major part of the fuel is stored inside the wings. Having a thin and slender wing might lead to not having enough volume to accommodate the fuel required.

Manoeuvrability of an aircraft is also affected by the AR of the wing, due its large moment of inertia when compared with a wing with a smaller aspect ratio but keeping all the other characteristics constant. During the roll movement in other to have the same angular speed a larger deflection of the ailerons need to be applied to do its higher moment of inertia.

Some times its practicality might be an issue, having a really large span can not be feasible for some cases. Airports have restriction for the maximum wingspan allowed which will influence the wing AR selected for the aircraft.

1.4 Problem Statement

Currently one wants to develop the most efficient aircraft possible and, as stated before, one of the solutions in order to achieve a higher efficiency is by introducing a high AR wing. The introduction of a high AR wing will lead to some problems, being some of them described previously.

Therefore, during the wing design process, it is critical to keep all the advantages of a high AR wing and trying to minimize the drawbacks that one will have with its introduction. A way of selecting the best design having in mind the constraints, such as having the stiffest possible structure with the least amount of material that will support the aerodynamic loads that will be under, is by doing a topology optimization to find the best solution for this problem.

A topology optimization can be conducted using different approaches, usually it is conducted using Gradient based (GB) algorithms. From the most known approaches one has the Homogenization Method [Bendsøe and Kikuchi, 1988] or SIMP [Sigmund, 2001]. Using GB algorithms brings some advantages, such as faster convergence and obtaining solutions with a higher accuracy when compared with stochastic approaches. However, they also display some drawbacks. The optimization variables, objective functions and constraints need to be continuous, the selection of the initial point it is also important because the algorithm will seek a solution in the neighbourhood of the initial point [Lee and Geem, 2004].

Meta-heuristic algorithms by combining rules and some randomness (stochastic methods), will avoid getting stuck in a local optimum having also more flexibility for the selection of the objective function, optimization variables and constraints.

Having the problem stated, the main goal of the given thesis is to implement a Topology Optimization using a meta-heuristic algorithm (Ant Colony Optimization) to optimize a high AR wing.

In order to achieve the main goal the following tasks need to fulfilled:

- Review the heuristic optimization approaches;
- Review the deterministic optimization approaches and topology optimization;
- Implement Ant Colony Optimization for a topology optimization problem;
- Model the geometry and aerodynamics loads which the wing is under.

The relevance of the use of ACO for a topology optimization has already been proved by Kaveh et al. [2008] and Luh and Lin [2009], however one wants to implement to find the best topology of a high AR wing.

1.5 Thesis Outline

One will start by introducing the Ant Colony Optimization algorithm explaining how it is implemented for the Travelling Salesman Problem, in chapter 2.

In chapter 3, topology optimization will be introduced. The approaches that are used to solve a topology optimization problem, such as Homogenization, SIMP and Level set will be described. The topology optimization of a cantilever beam subjected to a point load will be solved.

Having the ACO algorithm introduced, one will explain how to implement the ACO algorithm to solve a topology optimization problem in chapter 4. Using the code developed one will solve a literature case in order to benchmark the work developed.

A model of the wing of the NOVEMOR of the EU 7^{th} framework project will be developed in chapter 5, being developed the 3D model in *Solidworks*®, using it to perform a CFD analysis to obtain the aerodynamic loads, to ultimately apply it in the structural mesh of the wing box.

In chapter 6, The wing box cross section of the NOVEMOR of the EU 7^{th} framework will be optimized using the code developed.

A 3D optimization of the wing box will be performed in chapter 7, where one will use to different meshes to perform the optimization.

Lastly, a summary of the achievements, conclusions performed and future work are included in chapter 8.

Chapter 2

Ant Colony Optimization

2.1 Introduction to ACO

In recent years metaheuristic search techniques have been developed, being some of them nature-inspired, such as the Genetic Algorithm, Simulated Annealing and Flower Pollination. This trend of developing algorithms which try to recreate the natural processes is mainly due to the assumption that these processes are optimal [Kaveh et al., 2008], being this optimal state reached with an evolutionary process. This claim is hard to prove because there are many constraints in nature, from limited resources to the competition between species.

From these nature-inspired algorithms that have been developed in the recent years, one of the approaches used to solve the problems was to use the social behaviour of animals, such as insects, recreating the way they interact. This approach to problem solving is called Swarm Intelligence [Kennedy and Eberhart, 2001].

One can solve an optimization problem using multiple methods, that can be divided in two different categories: Deterministic or Non-Deterministic, the classification of the different methods can be seen in Figure 2.1. An algorithm is described as being deterministic if for any given input, the output of the optimization process will always be the same, *i.e.*, for the same set of design variables, constraints, objective function and initial guess the optimum found will always be the same. On the other hand, non-deterministic algorithms may generate different outputs for different runs of an optimization process. This behaviour can be explained by the randomness that some process of the algorithm exhibits.

Deterministic algorithms can be divided between Gradient Based or Enumerative. Gradient Based algorithms use the gradient of a function in order to search for the maximum or minimum of the objective function. They start from an initial point and move in the design domain using the information given by the function gradient. Due to its dependence of the initial point, it is not guaranteed that the global optimum will be found, being a local minimum/maximum found instead. Gradient Descent and Newton's method for optimization problems are two examples of optimization algorithms that are gradient based ones. Enumerative algorithms will start by considering all the set of solutions for the given problem, after considering all the possible solutions the best feasible one will be selected. In order to solve the problem, one will break the problem into 2 sub problems at each iteration. Examples of enumerative methods are: Dynamic Programming and Branch and Bound [Narendra and Fukunaga, 1977].

Non-Deterministic algorithms can be split in Stochastic and Heuristic. Both methods use some sort of randomness during the optimization process, however, in the Stochastic methods aleatory solutions will be generated until a feasible solution close to optimum comes up, on the other hand, heuristic problems have some randomness during the solution generation, being this randomness influenced by some sort of rule or prior knowledge acquired during previous iterations. Monte-Carlo sampling and Stochastic tunnelling are some examples of stochastic methods used to solve optimization problems [Hammersley and Handscomb, 1979].

The Heuristic methods can be divided in two subsets Metaheuristic and *Ad Hoc* Heuristic. The last subset, *Ad Hoc* Heuristic corresponds to algorithms which were developed to solve a specific problem, not being intended for application to solve different problems. On the other hand, Metaheuristic algorithms are developed to solve many different problems. These algorithms are defined as being higher-level procedure developed to find a sufficiently good solution to an optimization problem. The metaheuristic algorithms can be split on Single solution or Population based methods. In the Population based methods many solutions per iterations are generated, being these solutions improved throughout the optimization process, on the other, the single solution methods for each iteration just one solution is generated. One can also separate the Population methods, such as Genetic algorithm (GA), from Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), which are designated as Swarm intelligence. This concept is used to describe algorithms that are nature inspired.

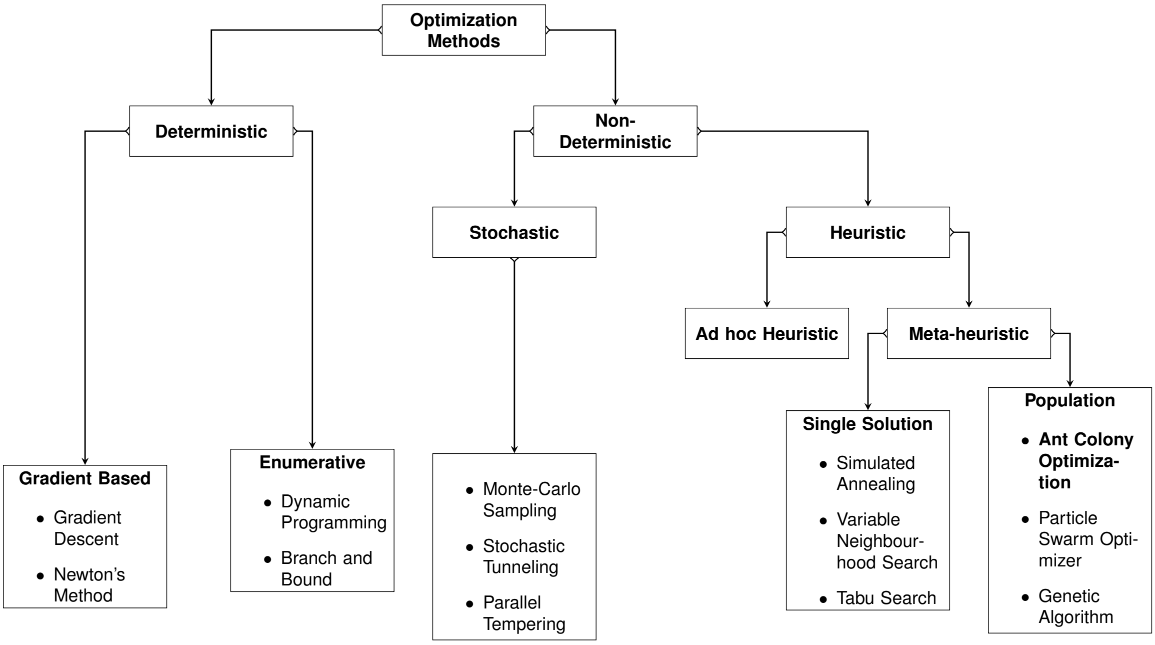


Figure 2.1: Classification of optimization methods.

Ant Colony Optimization, as stated before, corresponds to a meta-heuristic algorithm, which means that gives a sufficiently good solution for an optimization problem, an approximate solution. Due to the interaction between the ant agents in the algorithm, ACO is described as having a population-based search.

Ant Colony Optimization (ACO) is an algorithm that is inspired in the foraging behaviour of ants. Ants, when exploring the surrounding in order to find food, leave behind a trail of pheromone on the ground in order to mark the most favourable path for the other ants to follow. Ant Colony Optimization is inspired in this mechanism used by ants.

2.1.1 Stigmergy

Pierre-Paul Grassé, a French entomologist, in the middle of the twentieth century was able to observe in a specie of termites that they would react to a *stimuli* [Dorigo et al., 2006], influencing the termite which produces it and the other ones from the colony. This *stimuli* was seen as a way to interact with the other termites from the colony, and was designated as stigmergy.

Stigmergy was described and differenced from other forms of communication by Grassé, in ”which the insect or worker is stimulated by its performance” [Grassé, 1959]. It is considered as an indirect, non-symbolic, form of communication. The insects will exchange information by modifying the surrounding environment, being this information local, the insects that will access this information will be only the ones that visited the modified environment.

Ants colonies are an example where sitgmergy is observed. The way that ants change the environment is by leaving behind a chemical substance designated pheromone. So when trying to find the path between the nest and food source, the path chosen will be, most of the times, the one with the highest concentration of pheromones. Using this strategy ants are able to find the shortest path in a efficient way.

2.1.2 Double Bridge Experiments

In order to study the deposition mechanism of pheromones and its use to decide the trails, some experiments were made. The one conducted by Deneubourg et al. [1990] or Goss et al. [1989], in which the Argentine Ant had to go from the nest to the food source, being these two points connected by a double-bridge, were conducted. The setup of the experience conducted can be seen in Figure 2.2. On the bottom of Figure 2.2 one will have the nest from where the ants will depart to search food, being the food on the top. The bridge was formed by two branches that connect the same two points, so the

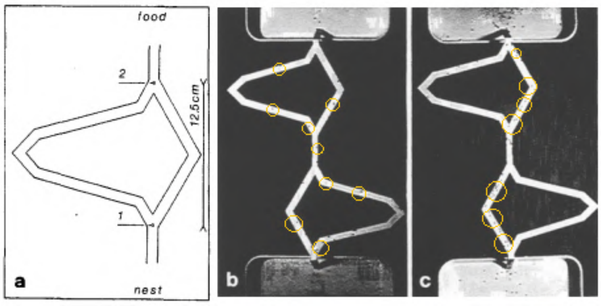


Figure 2.2: a) One module of the bridge, b) 4 min c) 8 min after the placement of the bridge. [Goss et al., 1989]

ants will have two options to go from one point to the other. The ratio between the longest and shortest branch is set to 2, meaning that the longest branch will be two times longer than the shortest.

In the beginning of the experience, the ants have no information about which branch they should choose, which will make the decision random. One expects that half of the ants will choose one branch and the other half the other one. One branch can be favoured over the other one, however, it would be due to stochastic oscillations. This can be seen in Figure 2.2 (b), where the ants are highlighted with a circle, having half of the ants are placed on the short branch and the other half on the other branch 4 minutes after the experience has started.

Being one branch smaller than the other, the ants that took the smallest branch will take less time to go from the nest to the food source, leading to a lower concentration of pheromones in the longest branch in which a bigger part of pheromone has vanished. This higher concentration of pheromones will bias the ant's decision, leading to a faster accumulation in the pheromone concentration in the short branch in the long run, being called this an *auto-catalytic* behaviour. Even having twice the size, a small percentage of ants still chose the long branch, which can be seen as a ”path exploration” behaviour. From Figure 2.2 (c) having passed 8 minutes almost all the ants use the shortest branches to reach the food from the nest.

In this experience, the short branch was removed, leaving only the long branch. After a few minutes the short branch was added. The ants were unable to move from the long branch to the short branch, being ”stuck” to the longest path. This fact can be explained with the high concentration in pheromones that the long branch had, which would bias the decision that the ants would take. Even with some ants performing path exploration, the ants will use the longest branch due to the highest concentration of pheromones, the pheromone concentration will maintain the same value for the longest branches being the smallest not used.

2.2 Optimization with ACO

Using the information above, Marco Dorigo [Dorigo and Stützle, 2004] and colleagues in 1991 developed the ACO as a novel nature-inspired metaheuristic algorithm to solve hard combinatorial optimization problems (NP hard problems).

From the initial work that Dorigo (1991) developed many different variants from this algorithm were developed, such as, $MAX - MIN$ Ant System, Elitist Ant System, Rank-based Ant System. The one that will be described in this section is the Ant System [Dorigo et al., 1996], being the first ACO algorithm developed.

Being a metaheuristic algorithm, ACO can be applied to different optimization problems by just making small changes. The combinatorial problems which the ACO can be applied to are in following form:

Combinatorial Optimization Problem $P = (\boldsymbol{S}, \Omega, f)$

- $\boldsymbol{S}$, search space, being discretized over a finite decision variables X_i, $i = 1, ..., n;$
- X_i takes values in $D_i = v_i^1, ..., v_i^{|D_i|}$

- Ω, set of constraints for the variables
- f, objective function, $f : \boldsymbol{S} \rightarrow \mathbb{R}_0^+$
- s is a feasible solution if the assignment of values to all variables satisfies the constraints. s^* is a global optimum if and only if: $f(s^*) \leq \ f(s) \ \forall \ s \ \in \ \boldsymbol{S}$

2.2.1 Travelling Salesman Problem

The algorithm implementation will be described using a well known NP-hard problem, the Travelling Salesman Problem (TSP) [Applegate et al., 2006].

The TSP is a problem where a salesman has a number of cities which he wants to visit in order to sell his products. During his round trip, the salesman just wants to visit each city once and also wants to do the shortest trip possible. This problem can be stated as finding a Hamiltonian tour of minimal length. The problem can be also described by an graph[1] (N, E), as seen in Figure 2.3, being the set of towns represented by N and the edges that connect the towns represented by E.

In the given graph (Figure 2.3), $N = 5$ and $E = 8$, the number that are on top of which edge, represent the "weight" of this edge, or the length of this edge. The length between town i and j would be given by d_{ij}. For instance, d_{32} represents the distance between town 3 and 2, being $d_{32} = 3$.

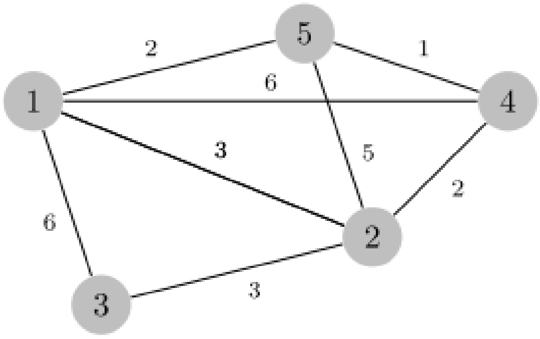


Figure 2.3: Example graph.

2.2.2 Algorithm Description

In order to solve the TSP problem, one will simulate n_a (dimension of the colony) ants moving around the graph. Each is an agent which has the following characteristics:

- able to choose one city at a time in order to find a solution. Each city will be chosen according to a probability, being this probability function of the town distance and the amount of pheromones in the edge that connects the towns;
- in order to have a Hamiltonian tour, the towns that have been already visited cannot be visited again (tabu list);

[1] Representation of a set of objects where some pairs of objects are connected by links.

- when a tour is completed, the agent will deposit pheromones in the edges (i, j) of its path accordingly to the pheromone update rule.

The different steps that the algorithm will perform during the optimization process are described in Algorithm 1.

Algorithm 1 Ant System ACO

```
Initialize AS
    • Generate, $n_a = colony\ dimension$, ants
    • Distribute randomly ants across the towns;
    • Assign an initial value of pheromones, $\tau_{ij}$, in the edges;
for $n = 1, ..., N$; $N = number\ of\ cycles$ do
    for $k = 1, ..., n_a$ do
        while Not Hamiltonian tour do
            Choose next town, using the Transition Probability Rule
        end while
    end for
    Distribute the pheromone, $\tau_{ij}$, across the edges;
end for
```

In Ant System, at the end of each cycle, one has n_a solutions for the problem, at the end of each cycle, using the solutions from every ant, the pheromones across the edges will be updated accordingly to the Pheromone Update Rule,

$$\tau_{ij}(t + n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}. \tag{2.1}$$

Considering that the begin of the cycle occurred at time t, and the end of the cycle occurred at $t + n$. Meaning that $\tau_{ij}(t + n)$ is equal to the pheromone intensity in the edge which connects town i and j at the end of the cycle. ρ is a coefficient which will simulate the evaporation that the pheromones suffers with time, being $(1 - \rho)$ equal to the evaporation of pheromones between t and $t + n$, $\rho < 1$ to simulate a constant evaporation rate. The increase of pheromones will be given by

$$\Delta\tau_{ij} = \sum_{k=1}^{n_a} \Delta\tau_{ij}^{k}, \tag{2.2}$$

being $\Delta\tau_{ij}^{k}$ the amount of pheromone laid in the edge (i, j) by the k^{th} ant. The amount of pheromone laid by each ant is given by

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & \text{if } k^{th} \text{ ant used } (i,j) \text{ edge} \\ 0, & \text{otherwise} \end{cases}, \tag{2.3}$$

being Q a constant and L_k the tour length of the solution found by the k^{th} ant.

During the process of finding a Hamiltonian tour the ants will choose the following town using the transition probability, described by

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{k \notin tabu_k} [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}}, & \text{if } j \notin tabu_k \\ 0, & \text{otherwise} \end{cases}, \tag{2.4}$$

where $\eta_{ij} = \frac{1}{d_{ij}}$ is designated as visibility. This quantity will remain the same throughout the optimization process. The importance of this quantity in the transition probability will be balanced using β. The probability of choosing the given town will be a balance between the proximity and the trail intensity present in the edge that connects both towns. α will balance the relative importance of the pheromones in the node transition rule.

The importance of these parameters during the optimization process will be shown in the following section (section 2.3) with a parametric study.

2.3 ACO Parametric Study

In order to understand the influence of the many parameters that rule Ant Colony Optimization algorithm, a parametric study was performed.

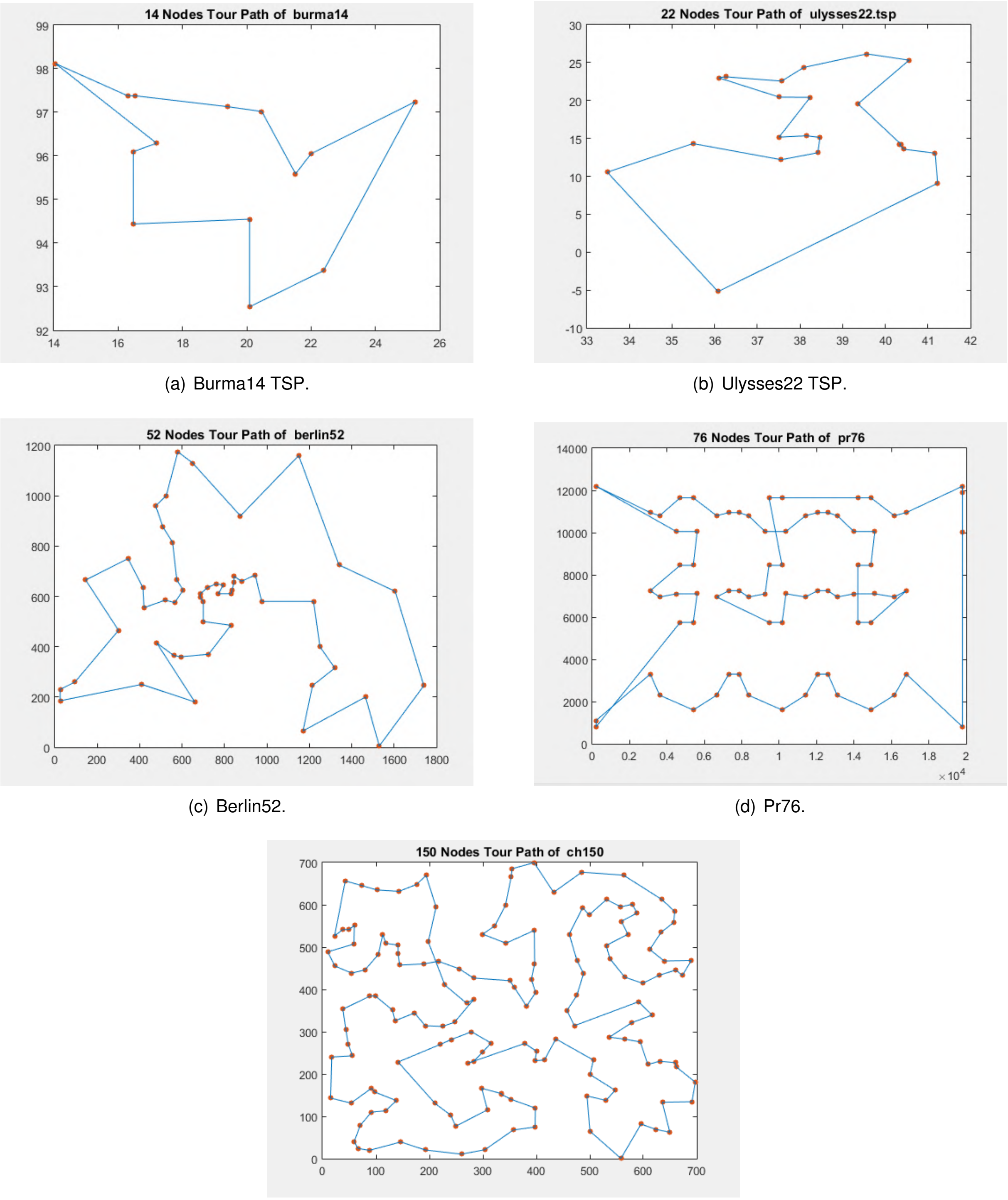
The ACO code used to perform the given parametric study was developed by Wang [2015], being the Ant System variation of ACO used in this code. In order to test the code and perform the parametric study, one used a library of TSP problems [RKUH, 2015].

One will start by solving different Travelling Salesman problems.

	Burma14	**Ulysses22**	**Berlin56**	**Pr76**	**Ch150**
Number of cities	14	22	52	76	150
Optimum [RKUH, 2015] [km]	3323	7013	7542	108159	6580
ACO [km]	3350	7016	7664	114488	6680
Error [%]	1.618	0.043	0.813	5.852	1.520

Table 2.1: TSPs solved using ACO.

The best tours found by ACO for each TSP can be seen in Figure 2.4. As can be seen in Table 2.1 the tour length for the different problems is close to the optimum solution found for the problem. The difference displayed between the optimum solution for the problem and the one found by ACO ranges between $0.043\% - 5.528\%$. Being ACO a metaheuristic algorithm, it was expected that the solution found would not be equal to the global optimum but a good approximation of the optimum. In Figure 2.5, one has the CPU time spent to find the optimum solution for the given number o cities. The CPU used to run the ACO code was a Intel(R) Core(TM) i5-4960k CPU @ 3.50GHz. The parameters used to do the following tests are summarized on Table 2.2.



(a) Burma14 TSP.

(b) Ulysses22 TSP.

(c) Berlin52.

(d) Pr76.

(e) Ch150.

Figure 2.4: Best tour found for TSPs.

Colony Dimension	60
Number of cycles	1000
ρ	0.65
α	1
β	5

Table 2.2: ACO parameters used to solve TSPs.

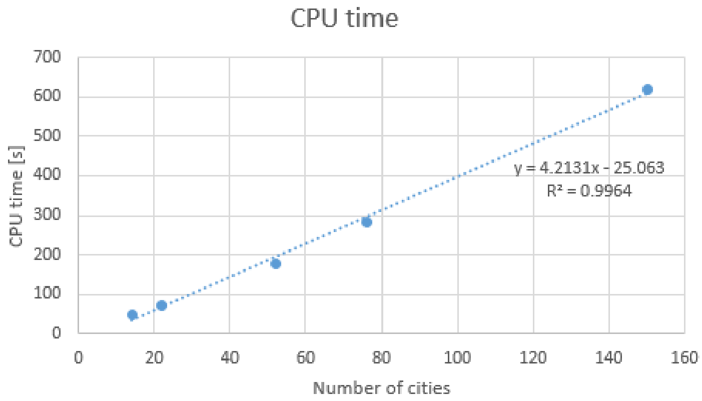


Figure 2.5: CPU time as a function of the number of cities.

The increase in number of cities will lead to an increase of time required to compute the optimum solution. From the 5 points collected one can notice that the computation time increases linearly with the increase of cities in the TSP, being $R^2 = 0.9964$ almost equal to 1, one can say that the behaviour is linear.

The parametric study of the algorithm will be performed using the Berlin52 TSP. This problem has 52 cities which will lead to a relatively small computing time, bringing some advantages when one needs to perform multiple runs of the algorithm.

One will start by changing the Colony Dimension used to solve the problem, keeping all the other parameters constant and with the values displayed on Table 2.2.

Colony Dimension	10	20	30	40	50	60	70	80
Tour Length [km]	7664	7857	7681	7681	7664	7664	7664	7664
CPU time [s]	38.81	65.05	91.06	118.58	145.30	183.94	209.41	231.60

Table 2.3: Parametric study of the colony dimension parameter.

One can notice in Table 2.3 that with the increase of the colony, the best tour length found will decrease, until a given colony dimension has been reached which will allow to explore a sufficient number of possible paths avoiding getting stuck in a local minimum. This is not true for the second point, where the colony dimension is equal to 20, although this can be explained due to the stochastic nature of the algorithm. With the increase of the colony dimension, for each iteration one will have more tours generated, so one will have a higher probability of finding a better tour than with a smaller colony dimension. The time required to do the optimization process will increase with the colony dimension, being this fact due to a higher number of solutions (ant's paths) to compute.

The next parameter being tested was α, this parameter regulates the influence of the pheromone in the node transition rule (equation 2.4).

α	0.5	1	1.5	2	2.5
Tour Length [km]	7613	7664	7792	7971	7985

Table 2.4: Parametric study of the α parameter.

As one can see in Table 2.4, the tour's length will increase with a increasing α, this fact may be due to the higher relevance given to the pheromone trail when compared with the visibility information.

β	1	2	3	4	5	6	7
Tour Length [km]	7778	7664	7664	7576	7664	7681	7681

Table 2.5: Parametric study of the β parameter.

For the β parameter, which regulates the importance given to the *visibility* information, one can conclude that increasing its value will increase the efficiency of the algorithm. However, for β above 4 the efficiency of the algorithm will decrease. This can be explained due to the fact that in the given graph (Figure 2.4(c)) one has many cities that are close to one another and the best option, usually, is to choose the closest city to the one that one is departing. Increasing the value of β one will bias the decision of the ant, "forcing" it to choose the closest node. The value of β for maximum efficiency is around 4, above this value the node selected might always be the closest node not allowing the selection of other nodes.

On Table 2.6 one has the parametric study of the number of iterations. ACO is terminated at a predefined number of iterations, being this specified by the user. The evolution of the tour length with the increase of the number of cycles is similar to the one seen for β, above a number of cycles the solution found will remain the same. Below 1000 cycles the tour length obtained displays a value superior for higher number of cycles, meaning that the algorithm needs more cycles in order to converge for the optimum solution.

Number of cycles	250	500	1000	1500	2000
Tour Length [km]	7890	7843	7664	7664	7664

Table 2.6: Parametric study of the number of iterations.

The last parameter that dictates the behaviour of the ACO algorithm is ρ, the evaporation factor for the pheromone trail.

ρ	0.35	0.45	0.55	0.65	0.75	0.85
Tour Length [km]	7664	7664	7681	7664	7664	7664

Table 2.7: Parametric study of the ρ parameter.

The evaporation rate for this problem displayed no effect on the final solution, only influencing the time required to converge for the optimal solution.

Chapter 3

Topology Optimization

Structural optimization can be separated in three types: Sizing optimization, Shape optimization and Topology optimization. The solution for the same problem using different structural optimization types are exemplified in Figure 3.1.

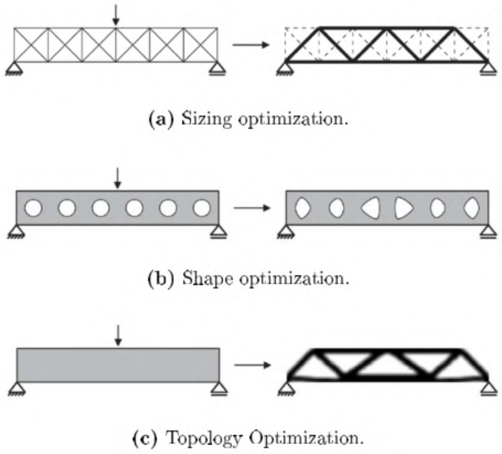
Structural optimization started with Sizing Optimization. The problems that one focuses on Sizing Optimization are related with the sizing properties of the domain, for example, finding the optimum thickness of a plate structure or finding the cross-section dimensions of the members of a truss structure. The domain will not change during the optimization process, therefore the finite element model that one will need to access the structural performance of the given structure will remain the same throughout the optimization process.

Shape optimization was developed next, where the problem defines is trying to find the optimal boundaries of a structure, some examples of this type of structural optimization include finding the optimal values for the parameters that define the middle surface of a shell structure or the location of the joints of a skeletal structure. The finite element model associated with the topology will have to change since the boundary of the domain will also change.

Sizing and shape optimization are restricted to the initial topology, therefore if one uses these kind of structural optimization, one might not be able to find a global optimum for the given problem being stuck in a sub-optimum due to the starting topology not being an optimum starting point.

Topology Optimization tries to find where one should place material in order to have the optimum topology for the given problem. The problem of finding the optimum configuration and spatial sequence of members and joints of a skeletal structure also belongs to the given optimization.

One performs Topology Optimization in order to know how the material in a prescribed design domain should be placed in order to achieve the best structural performance possible. This approach consists on repeated analysis and design update steps, guided by gradient computation or other optimization algorithm.



(a) Sizing optimization.

(b) Shape optimization.

(c) Topology Optimization.

Figure 3.1: Types of structural optimization [Félix, 2013].

3.1 Introduction to TO

Structural topology optimization can be divided into two sub fields, Layout Optimization and Generalized Shape Optimization [Rozvany, 2001].

Layout Optimization (LO) deals with optimization problems for grid-like structures. The optimization of these structures is done by simultaneously defining the connectivity between elements (topology), the location of the joints and the cross-sectional dimensions. One starts with a structure, which contains all the possible elements, being the non-optimal ones removed during the optimization process.

Generalized Shape Optimization (GSO) tries to find the best topology of the internal boundaries and the shape of both internal and external boundaries.

For a discretized GSO, one will divide the domain into a finite number of elements, each element can only have two states, contains material (1) or it is a void (0). In an exact problem, the number of elements will tend towards infinite, which means the problem that one has to solve is the assignment of material to each point of the design domain. In a GSO problem, one can also have a set of material which can be assigned to the elements, being this problem called a Multi-Material Optimization.

Topology optimization was introduced by the Australian inventor Anthony George Michel (1904) to minimize the volume of a truss subjected to stress constraints ($-\sigma_C \leq \sigma \leq \sigma_T$), being σ_C and σ_T the permissible stress in compression and in tension, respectively. If all the elements have a permissible stresses, the total volume (V) of the truss can be given by [Rozvany and Lewi, 2014],

$$V = -\sum_{C} \frac{L_i F_i}{\sigma_C} + \sum \frac{L_i F_i}{\sigma_T}, \tag{3.1}$$

being L_i the length of the member i and F_i the force which is subjected. The optimality criterion can be summarized as

$$\bar{\varepsilon} = k \; sgn \; F \text{ (for } F \neq 0), \; |\bar{\varepsilon}| \leq k \text{ (for } F = 0), \tag{3.2}$$

where $\bar{\varepsilon}$ is an admissible strain and k a positive constant. The resulting F must satisfy the equilibrium and static boundary conditions. This optimality criteria is only valid if one of these conditions are met:

- the problem is statically determinate for any choice of a stable topology;

- both permissible stresses have the same value.

From the work of Michel (1904) many developments were made. Rozvany, Olholf, Bendsøe *et al.* (1985/87) developed the first analytical solutions for the optimal perforated plates and the correct expressions for the rigidity tensor of homogenized optimal microstructures [Rozvany, 2001]. Bendsøe and Kikuchi [1988] developed the first finite element based topology optimization for higher volume fraction, where the Homogenization approach to solve topology optimization was introduced. Almost at the same time, the SIMP approach was developed by Bendsøe [1989] and later by Zhou and Rozvany [1991] or Mlejnek [1992].

3.2 Mathematical Description

An optimization problem tries to find the maximum or minimum of a function $f(\mathbf{x})$, being $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{R}^{\ltimes}$. f is designated the objective (cost) function and $x_i,\ i = 1, ..., n$ are the design variables of the problem.

In an optimization problem it is normal that the design variables have to belong to a given interval, this extra conditions that one has to satisfy are designated as constraints. The constraints may be in an equality or in an inequality form, or variable bounds.

One can express the optimization problem as [Hassani and Hinton, 1999]

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\qquad h_j(\mathbf{x}) = 0, \qquad j = 1, 2, ..., n_h \\ &\qquad g_k(\mathbf{x}) \leq 0, \qquad k = 1, 2, ..., n_g \\ &\qquad x_i^l \leq x_i \leq x_i^u, \quad i = 1, 2, ..., n\ , \end{aligned} \tag{3.3}$$

being n_h and n_g the number of equality and inequality constraints, respectively, in the optimization problem. x_i^l and x_i^u correspond to the lower and upper bonds that the design variable x_i can have.

It is defined as the feasible domain the subset of values that the design variables can have that will satisfy all the constraints in the problem, the infeasible domain corresponds to the remaining subset that it is not included in the feasible domain.

3.3 Homogenization Approach

The Homegenization method was introduced by Bendsøe and Kikuchi [1988], in which one computes the optimal distribution in space of an anisotropic material, constructed by the introduction of periodically distributed small holes in a homogeneous and isotropic material, which is able to carry the loads being subjected to and satisfy all other requirements.

Shape optimization, which was the closest to topology until the introduction of the homogenization approach, required boundary variation methods. The scheme used to implement these methods could be one where the coordinates of the nodal points of a finite element model are the design variables, requiring also an additional method to maintain the regularity of the elements close to the boundary. Another way to implement a boundary variation method could be the introduction of boundary segments, being the boundary described by a set of simple segments. However, the design space is restricted to the boundary definition (topology changes are not allowed).

One can also consider a topology optimization problem as a point-wise material/no material, which would not require shape parameters nor basic shape functions. In this case that one would require a discrete optimization algorithm, but such approach is not ideal because is unstable [Hassani and Hinton, 1998].

The given problem of shape topology is transformed into a material distribution problem using a composite material. The composite material is constituted by two parts, substance and void, being the optimal microscopic void distribution the problem to be solved. This approach ”is applied to determine macroscopic constitutive equations for the material with microscopic material distribution” [Bendsøe and Kikuchi, 1988]. In the Homogenization approach, the structural element is considered as being only defined by the loads it has to carry, its volume and the design requirements.

The topology problem can be transformed into a sizing problem by the introduction of a material function density distribution in which one considers a composite consisting of an infinite number of small holes periodically distributed, being the on-off nature of the problem in a microscopic scale and not in the macroscopic one.

One will start by choosing the design domain and the finite element discretization to be used, consisting each element of a cellular material with a specific microstructure. The finite element model generated in the beginning of the optimization process will be kept throughout the optimization.

The microstructure chosen should allow the density of the material to cover the whole range of values, from 0 (void) to 1 (solid). It also should fit the periodicity assumption and must be defined by few parameters since these parameters will be the design variables of the optimization problem. One option for a microstructure is a square cell with a centred rectangular hole, another possibility might be a generalized ellipsoid. The square cell with a rectangular hole microstructure can be seen in Figure 3.2.

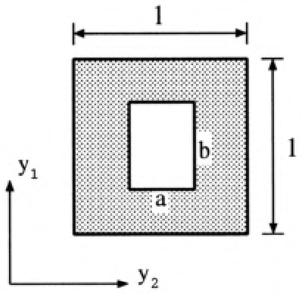


Figure 3.2: Unit cell with rectangular centred hole [Hassani and Hinton, 1998].

A constant microstructure is assumed for each element, resulting in a constant homogenized elasticity matrix for each finite element. The solid part of the microstructure is composed by an isotropic material, which will become an orthotropic material due to the rectangular hole. For the 2D (plane stress/strain) the elastic constitutive law will be given by

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} = \begin{bmatrix} \lambda D_{11} & D_{12} & 0 \\ D_{12} & D_{22} & 0 \\ 0 & 0 & D_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2 \cdot \varepsilon_{12} \end{Bmatrix}, \tag{3.4}$$

being σ_{ij} and ε_{ij}, with $(i, j = 1, 2)$, the stresses and strain and D_{ij} the elements of the reduced stiffness matrix of the material [Hassani and Hinton, 1998].

For the cellular material the elements of the matrix of elastic moduli $\mathbf{D}$ depends on a,b and θ,

$$\mathbf{D} = \mathbf{D}(a, b, \theta), \tag{3.5}$$

where the parameters a and b will be given by the asymptotic homogenization theory and θ can be found by applying a frame rotation formula. The density function will be a function of the first two parameters that define the rectangular hole of the cell (a, b), being given by

$$\rho(a, b) = (1 - a \cdot b)\rho_s. \tag{3.6}$$

With equation (3.6) one can perform the topology optimization without having to worry about the discrete formulation of the problem.

3.4 Density Approach

One of the approaches introduced after the Homogenization approach was the SIMP or power-law approach, where one uses Solid Isotropic Microstructures with Penalization for intermediate densities. SIMP was introduced by Bendsøe [1989], with the goal of removing the discrete nature of the topology optimization by introducing the density variable, a continuous variable. Also, it was introduced as an easy but artificial way of reducing the complexity of the homogenization approach and improve the convergence to a 0-1 solution.

SIMP uses a pseudo-density as a design variable, being the relation between the density variable and the effective material stiffness given by the power law [Sigmund and Maute, 2013],

$$E(\rho_i) = g(\rho_i)E_0, \tag{3.7}$$

where ρ_i is the density of the element i, E_0 is the Young's Modulus of the material and g is a penalization function, responsible for making the density variable tend towards 0 or 1. The penalization function proposed by Bendsøe [1989] is represented by

$$g(\rho_i) = \rho_i^p, \tag{3.8}$$

where p represents the penalization parameter. When $p = 1$, for the compliance objective function, the problem solved is a convex problem with a unique solution.

The selection of the penalization parameter (p) is fundamental for the convergence of the problem. For $p > 1$ and a volume constraint, the intermediate densities are penalized as can be seen in Figure 3.3 (being $p = n$).

For a low value of p one will have in the end of the optimization process many elements with intermediate densities, generating too much grey scale, while for a higher value of p one will have a faster convergence to a local minimum. Having $p > 1$ will effectively suppress the intermediate densities but will transform a convex problem into a non-convex problem which for a compliance problem one cannot guarantee the globality of the optimum [Rozvany, 2001].

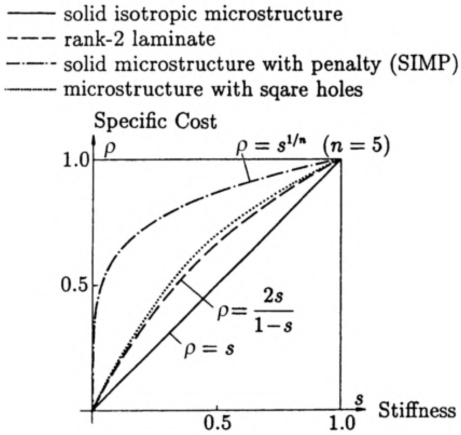


Figure 3.3: Material stiffness given by the power-law [Rozvany, 2001].

Usually the value of the penalization parameter used is equal to 3, ensuring physical meaning of elements with intermediate densities and making the density gradient equal to topological derivatives for elasticity [Sigmund and Maute, 2013].

Another penalization function introduced by Stolpe and Svanberg [2001], in order to alleviate the non-concavity generated by the SIMP scheme [Sigmund and Maute, 2013], was RAMP (Rational Approximation of Material Properties). The penalization function of RAMP is given by

$$g(\rho_i) = \frac{\rho_i}{1 + q \cdot (1 - \rho_i)}, \tag{3.9}$$

being q the penalization parameter.

The difference between RAMP and SIMP is that RAMP does not display a zero gradient for $\rho_i = 0$, which may influence the convergence properties and also alleviate the problems with low density modes in dynamic problems [Sigmund and Maute, 2013].

The penalization factors p and q need to be selected for each problem being solved in order to have a sufficient penalization to avoid grey scales. For the compliance problem, when $p = 1$ one has a convex problem. As explained before, one does not want to solve this problem with $p = 1$ because the intermediate solutions will not be penalized. On the other hand, one does not want to solve the problem with a large p because it will lead to an early convergence. As a result, it is common when using approaches with penalization parameter to start with $p = 1$ for the first iterations and gradually increase the penalization during the optimization.

The advantages that SIMP presents are [Rozvany, 2001]:

- Computational efficiency, since one will only have a free variable for which element;
- Penalization parameters have no restrictions;
- Conceptual simplicity, the algorithm does not require derivations involving higher mathematics;
- Robustness, SIMP can be used to any given set of objective function and constraints;
- Homogenization of the microstructure is not required.

The biggest disadvantage of SIMP is the dependence of the solution on the penalization parameter and not necessarily converging to the optimal solution.

3.5 Cantilever Beam Example

In order to compare the results (chapter 4) obtained for the cantilever beam problem using the ACO algorithm a topology optimization for the same problem was performed using a gradient-based approach, more precisely, using the SIMP approach. To perform this optimization, *top*, a 99 line topology optimization code written in *MATLAB*® developed by Sigmund [2001], was utilized. Top implements a topology optimization problem for a compliance minimization of statically loaded structure.

The problem that will be solved in chapter 4, is described in Kaveh et al. [2008] being the cantilever supported and loaded as can be seen in Figure 3.4.

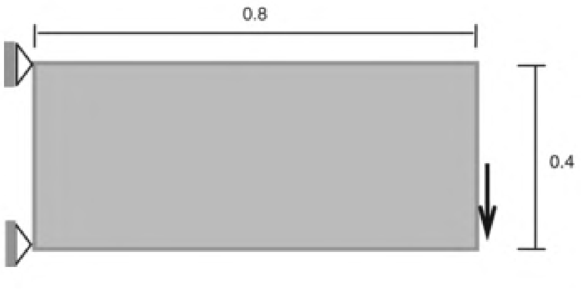


Figure 3.4: Literature case layout.

The design domain is discretized with 80 elements in the *x* direction and 40 in the *y* direction [Kaveh et al., 2008].

Top requires 5 inputs, the number of elements in the x and y direction, *nelx* and *nely*, respectively, the volume fraction, *volfrac*, the penalization parameter, *penal*, and the the filter size, *rmin*.

In order to solve the given problem, one needs to change the boundary conditions in the *top* code. The changes introduced are the following:

```
F(2*(nelx+1)*(nely+1)-(nely),1) = force;
fixeddofs = union([1:2], [2*(nely+1)-1:2*(nely+1)]);
```

The input values used by *top* to solve the topology optimization problem are listed in Table 3.1.

`nelx`	80
`nely`	40
`volfrac`	0.45
`rmin`	1.5

Table 3.1: *Top* inputs.

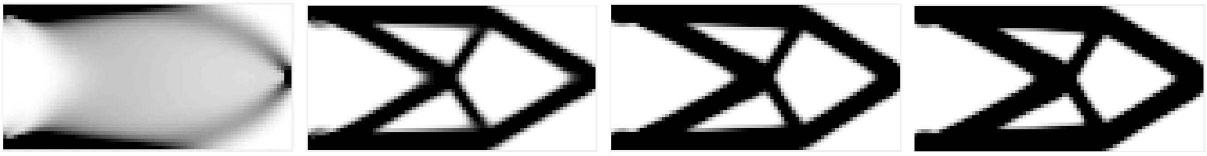
As one can see in Figure 3.5, the resulting structure of the optimization approach will be dependent of the penalization factor value. For $penal = 1$ one will have the expected grey areas, since the intermediate densities are only penalized for values $penal > 1$, as described in Figure 3.3.

Increasing the penalization factor will lead to a lower formation of grey areas because that intermediate densities are penalized. However, the algorithm will take more iterations to converge. The optimization process in *top* will stop when the maximum value of the changes in the design variables between consecutive iteration is lower than 0.01.

`penal`	1	2	3	4
Iterations	13	58	101	177

Table 3.2: Iterations required for convergence for each *penal*.

In Table 3.2, one has the iterations required for each value of *penal*. One can conclude that the decrease in the penalization factor will lead to an earlier convergence of the optimization process.



(a) `penal` = 1. (b) `penal` = 2. (c) `penal` = 3. (d) `penal` = 4.

Figure 3.5: Resulting topologies for different values of `penal`.

Chapter 4

Topology Optimization using ACO

Ant Colony Optimization, being a metaheuristic algorithm, displays some advantages when solving a Topology Optimization problem. One of the advantages that ACO presents when compared to a Gradient based algorithm is that it allows the use of a wider variety of objective functions. With a metaheuristic algorithm one just needs a function that describes the performance of the solution, using a gradient-base algorithm one needs a differentiable objective functions, due to the search process being based on the gradient of the objective function. In a metaheuristic algorithm the search process will be done using a stochastic process while using some rules to reward the best solution found.

Being the TO problem a discrete problem, applying the ACO to this problem is straightforward without having to make many changes to it. On the other hand, if one used a gradient-based algorithm to solve this problem, the optimization variables would need to be continuous, which in the end of optimization process would generate checker boards, and an additional filter would be needed to remove/add grey elements.

Gradient-based algorithms, however, are faster and require less computational expenses. On ACO one will need to do multiple Finite Element analysis at the end of each iteration, which can be problematic if the problem that one is trying to solve is complex. Furthermore, with the increase of the number of optimization variables, finding a solution using a metaheuristic algorithm can be really hard, as result of the stochastic search method for the design solutions.

4.1 ACO for a TO problem

4.1.1 State of Art

Kaveh and Shojaee [2006] employed Ant Colony Optimization algorithm to discover a optimal skeletal structure. In this problem, ACO was used to solve a sizing optimization problem. Its introduction to structural problems was mainly due to the fact that a discrete domain could be optimized in a simple way. Being ACO used for combinatorial optimization problems its application to discrete or continuous optimization of structures is straightforward. Also, its advantages rely on its memory, in which information from the past is used to improve future searches.

Another successful sizing optimization using Ant Colony Optimization was conducted by Hasançebi and Çarbaş [2011]. In this paper, the authors conducted an optimization to find the optimum size design of a pin-jointed truss which will lead to the minimum weight, being the elements of the truss chosen from ready sections. Two variants of the ACO were applied, the Ant System and the Elitist Ant, and the results of both variants were compared. The Elitist Ant displayed a better performance in the minimum weight problem for the two examples conducted. Some problems that the ACO algorithm displays were identified by Hasançebi and Çarbaş [2011], being one of them the large early concentration of pheromones in a subset of paths, making the search being mostly dominated by this subset of solutions. Meaning that, it is possible that some paths that could belong to the optimum path are discarded in an early stage.

Kaveh et al. [2008] implemented the first topology optimization using ACO algorithm. A 2D and 3D topology optimization were conducted, where the objective was to find the stiffest structure possible with a given amount of material. To find the stiffest possible structure under the given constraints the authors used the element's contribution to the strain energy. From this work, the authors were able to conclude that the algorithm was suited to apply to topology optimizations with a large number of design variables. Furthermore, due to the non-existence of grey elements, post-processing of the obtained results is not required, such as a image processing filter. Being a metaheuristic algorithm the required number of structural analysis will be higher than in the regular algorithm used in topology optimizations. Lastly, the results obtained by the author were considered ”not the most impressive”, but when using a noise cleaning filter alongside the layouts obtained displayed good quality.

Hasançebi et al. [2009] conducted a performance analysis between seven different metaheuristic search techniques: *Stimulated annealing (SA), Evolution strategies (ESs), Genetic algorithm (GA), Particle swarm optimizer (PSO), Tabu Search (TS), Ant Colony Optimization (ACO) and Harmony search algorithm (HS).* The performance of these algorithm was assessed using the problem of the design of a pin jointed structure optimization. The solution accuracy, convergence rate and reliability of the algorithms was assessed in [Hasançebi et al., 2009]. From this work, the author concluded these metaheuristic algorithms are really sensitive to a large set of parameters and issues, such as the number and type of design variables, the starting solution, the number of iterations cycles and the number of independent runs performed. Regarding the performance, the algorithms that displayed better performance were the SA and ESs. SA usually reached a better solution but showed a slower and gradual convergence rate, when comparing with the ESs which displayed a rapid and linear convergence towards the optimum.

Luh and Lin [2009] also implemented a structural topology optimization using ACO. The algorithm implemented was a modified version of ACO, in which one has elitist ants, niche strategy and memory of multiple colonies. The authors concluded that the proposed algorithm was indicated to search for innovative solutions to structural problems and for investigating optimal topologies. The structures obtained after the implementation of the algorithm displayed well defined truss-like members.

4.1.2 Problem Statement

In order to use ACO to solve a Topology Optimization problem one needs to do some changes to the original algorithm. In this section one will describe the modifications required and explain the implementation of a topology optimization of a 2D beam subjected to a point load in its extremity and constrained in two points. The implementation of the ACO in a TO problem will be done following the work of Kaveh and colleagues [Kaveh et al., 2008] as a reference.

The TO problem which will be considered is the one solved in [Kaveh et al., 2008], this problem will be used as a literature case. After solving this problem and comparing the obtained results, one will be able to solve a TO problem with a higher complexity.

Ant Colony Optimization algorithm due to its discrete nature it is a great candidate to solve a TO problem. In a TO problem one will discretize the domain with small elements which are subjected to some constraints and also to loads/torques. Then, the elements of the domain will be chosen in order to maximize or minimize the selected objective function. Most of the times, one is looking for the stiffest possible structure with the least amount of material possible. Maximum stiffness can be translated into a physical sense, in minimal strain energy or having the lowest amount of displacement possible.

In the current work, the objective function selected to optimize the structure will be the strain energy, which one wants to minimize. The strain energy can be given by

$$U = \int_V U_0 \, dx \, dy \, dz \tag{4.1}$$

$$\text{with } U_0 = \frac{1}{2E} \cdot (\sigma_x^2 + \sigma_y^2 + \sigma_z^2) - \frac{\nu}{E} \cdot (\sigma_x \sigma_y + \sigma_y \sigma_z + \sigma_z \sigma_x) + \frac{1}{2G}(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2), \tag{4.2}$$

where E is the *Young Modulus*, ν the *Poisson's Ratio* and G the *Modulus of Rigity*.

The optimization problem that one will solve can be stated as

$$\begin{aligned} &\text{minimize:} && U(\overline{\boldsymbol{u}}) \\ &\text{subject to:} && \frac{V_f(\overline{\boldsymbol{u}})}{\overline{V_f}} \leq 1, \text{ and physical constraints,} \end{aligned}$$

where, $\overline{\boldsymbol{u}}$ represents the field displacement, $V_f(\overline{\boldsymbol{u}})$ corresponds to the volume fraction of the given solution, and $\overline{V_f}$ is the maximum volume fraction allowed. The physical constraints require that the element where the load is applied and the support elements must be present in the final design, and that the structure must be a connected structure, meaning that one must be able to find a continuous path that connects all the elements with the physical constraints.

One needs to discretize the objective function, to have its representation across the discrete domain. The objective function discretized across the domain is given by

$$U(\overline{\boldsymbol{u}}) = \frac{1}{2} \sum_{e=1}^{N} \int_{V^e} \varepsilon^T(\overline{\boldsymbol{u}}) D^e \varepsilon(\overline{\boldsymbol{u}}) dV, \tag{4.3}$$

where V_e is the volume of the element in the domain, the integral represents the strain energy, being the total strain energy of the domain equal to the sum of the strain energies from each element (N), ε represents the strains and D^e is the constitutive matrix of the element.

Usually, a topology optimization is performed by assigning a relative density to each element. Having in mind the discrete nature of the algorithm, one will have only two states, the element is present or not, which can be translated as

$$\chi(x) = \begin{cases} 1 & \text{if } x \text{ is a solid element} \\ 0 & \text{if } x \text{ is a void element.} \end{cases} \tag{4.4}$$

If the material is isotropic, the constitutive matrix for the element $\boldsymbol{x}$ will be equal to

$$\boldsymbol{D}(x) = \chi(x)\boldsymbol{D}^0, \tag{4.5}$$

being $\boldsymbol{D}^0$ equal to the elasticity matrix of the solid material.

4.1.3 TS problem vs TO problem

One can do some analogies between the Travelling Salesman problem and a Topology Optimization problem, in order to use ACO to solve the TO problem. In the TS problem one has a graph (N, E), N cities connected by E edges, in a TO problem one will have a domain descritized by a E elements. The edges, E, that one has in the TS problem, will be the elements in a TO problem. The elements that will be present in the solution will be considered as the cities N.

In the TS problem, since one is trying to find the shortest path, time is outmost importance and relevance, the shortest path will also be the one that will take the less time to travel. Every move an ant makes during the process of finding a solution will be taken into account and the pheromone update will occur at every step. In a TO problem, one will use ”time” as a reference of the present iteration.

4.1.4 Flowchart

The process to solve a Topology Optimization problem using ACO algorithm can be described by the flowchart in Figure 4.1.

In the Initialization process one will:

- Generate the colony (n_a ants);
- Distribute a initial value of pheromones in the domain.

The Generate Solution process can be described by the flowchart in Figure 4.2.

One will start by distributing the n_a ants across the domain, the distribution will be done randomly, being each ant placed in a random element. The next step that each ant will make is choosing a element, the element will be chosen using the Element Transition rule. After choosing a new element for the ant path, one must check if the volume fraction that was assigned is respected. If the volume fraction was exceeded the ant path will be deleted and a new element in the domain will be assigned to that ant to discover another path. If the volume fraction is not surpassed, one must check if the Boundary Condition

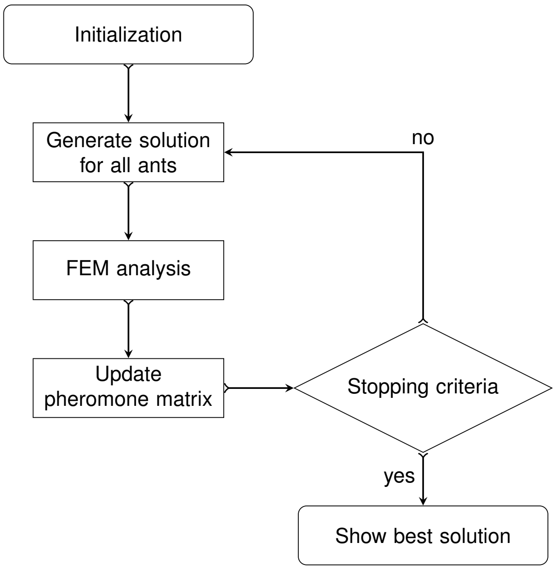


Figure 4.1: Flowchart for ACO applied for a topology optimization.

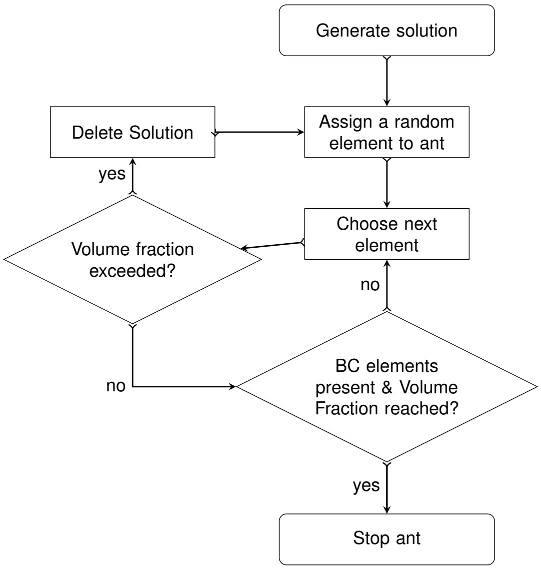

Generate solution
Delete Solution
Assign a random element to ant
yes
Choose next element
Volume fraction exceeded?
no
no
BC elements present & Volume Fraction reached?
yes
Stop ant

Figure 4.2: Flowchart for the generate solution process.

elements are present in the path. If they are not present in the path the ant must select another element. This process will be repeated until all the Boundary Condition elements are present in the ant's path and the Volume fraction which is assigned is reached.

In the given implementation the solution search will end when the given solutions have achieved the maximum volume fraction defined and the boundary conditions are met. One imposes the solution search until the maximum volume fraction has been reached, in fact the minimum strain energy will be found for the full design domain. So, the given problems that one has to solve is the distribution of that amount of material.

After all the ants have found a path that meets all the constraints, one will perform a Finite Element Method analysis to each solution. From the FEM analysis one will retrieve the Strain Energy of each element of the domain, in order to do the Pheromone Update.

This processes will repeat until one has reached the Stopping Criteria. The Stopping Criteria in ACO will be defined by the user, consequently, when one reaches the defined number of iterations the optimization process will end.

4.1.5 Element Transition Rule

Instead of having a Transition Probability, one will have an Element Transition rule. The Element Transition rule corresponds to the probability that a given element has to be selected as the next move for an ant, being given by

$$P_i = \frac{(\tau_i(t))^{\alpha}}{\sum\limits_{j=1}^{N} (\tau_j(t))^{\alpha}}. \tag{4.6}$$

The probability of an ant choosing the element i is equal to the intensity of pheromone trail present in the element i to the power α, dividing by the sum of the intensity of pheromone trail to the power α of all the elements which are valid to as a next step. $\tau_i(t)$ represents the pheromone trail in element i in the t iteration of the optimization process. The parameter α is used to control the relative weight of the pheromone trail, one should be careful when doing the tuning of this parameter, because it can lead to premature convergence to non-optimal solutions.

As it can be noticed, equation (4.6) when compared with equation (2.4) is missing the visibility part on the probability calculation. This can explained by the fact that one is using as the objective function the strain energy of the solution. Changing the elements that will be present in the solution will also change the strain energy that each element will have. Consequently, one can not know prior to the solution generation which will be effect of the presence of that element in the solution. So, the *visibility* part is removed from the probability calculation.

4.1.6 Pheromone Update

The pheromone intensity ($\Delta\tau_i^k$) laid by ant k in the element i, will be given by

$$\Delta\tau_i^k = \frac{(U_i^k)^{\lambda}}{\sum\limits_{j=1}^{N} (U_j^k)^{\lambda}}, \tag{4.7}$$

where U_i^k is the strain energy of element i of the solution obtained by ant k, λ is a parameter used to tune the influence of the strain energy in the algorithm, helping with its convergence.

After having the increment in pheromones for every element from all the solutions found by the ants, one can do the update of the Pheromone Matrix. The update of the pheromone matrix will be done using the same rule used in ACO Ant System 2.1. Instead of $\Delta\tau_i$ being a function of the length of the tour it will be given using equation (4.7).

4.2 MATLAB® Implementation

Using the work of Kaveh et al. [2008] as reference, one implemented a topology optimization using the Ant Colony Optimization algorithm using MATLAB® and interfacing it with ANSYS® to do the required finite element analysis.

As explained in the flowchart (Figure 4.1), one will have 4 main stages in the algorithm implementation. The first one, the Initialization will be held by the `input_ansys_init` and `f_initphero` which will be responsible for generating the input for ANSYS® and generate the initial pheromone matrix, respectively, the initialization of the pheromone matrix will be explained in subsection 4.2.2. The generation of feasible solutions will be the done by the function `aco_main`, being for the given optimization problem the Optimizer. `aco_main` is composed mainly by the functions `ant_init`, `trans_rule`, `bdcheck` and `ant_reset`. Its implementation will be explained in subsection 4.2.3.

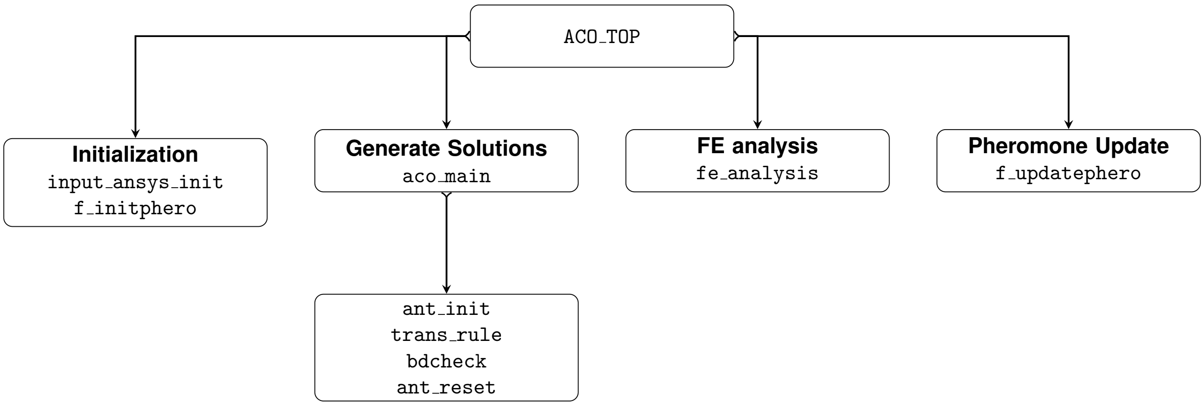


Figure 4.3: Code schematic.

Function `fe_analysis` will generate the input and call ANSYS® in batch mode for each solution found. The interface between MATLAB® and ANSYS® will be explained in section 4.3.

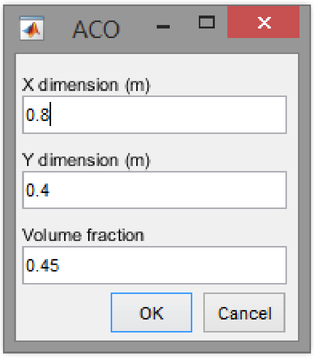
Lastly, the function `f_updatephero` is responsible for the Pheromone Update. A schematic of the code developed is represented in Figure 4.3.

4.2.1 Main function

The `ACO_TOP` script contains all the code required to do the optimization explained in section 4.1.2. The first step is to define the domain in which one will do the optimization, so, one will ask the dimensions of the domain.

The values that are introduced by default correspond to the dimensions of the beam described in Figure 3.4, which are the same as the one solved in Kaveh et al. [2008]. After having the dimensions of the domain defined one needs to define the number of elements that will discretize the domain. By default one will have $n_{ely} = 40$ elements in the y direction and $n_{elx} = 80$ in the x direction [Kaveh et al., 2008].

The next step is the definition of the dimension of the colony ($colony\_dim$), and the value assignment of the parameters used in ACO. By default, the parameters will have the value shown in Table 4.1


ACO

X dimension (m)

0.8

Y dimension (m)

0.4

Volume fraction

0.45

Figure 4.4: MATLAB® window with domain definition input.

`colony_dim`	15
`n_ite`	30
`evap` (ρ)	0.7
`alpha_p` (α)	1
`lambda` (λ)	2

Table 4.1: ACO parameters.

4.2.2 Pheromone Matrix Initialization

Having all the parameters assigned, one will initialize the pheromone matrix. At this point one needs to code the elements present in the domain, *i.e.*, assign a number to each element in such a way that the given number would indicate the position of this element in the domain. ANSYS® when meshing assigns a number to each element. To have a simple transition from the ANSYS® to MATLAB®, one decided to keep and use this numbering. The first element is the one on bottom corner on the left, being the number given sequentially until the bottom corner on the right is reached. The element $n_{elx} + 1$ will correspond to the one on top of the element number 1, being this process repeated until one has reached the top corner on the right, the element $n_{elx} \cdot n_{ely}$. The domain discretized and numbered can be seen in Figure 4.5.

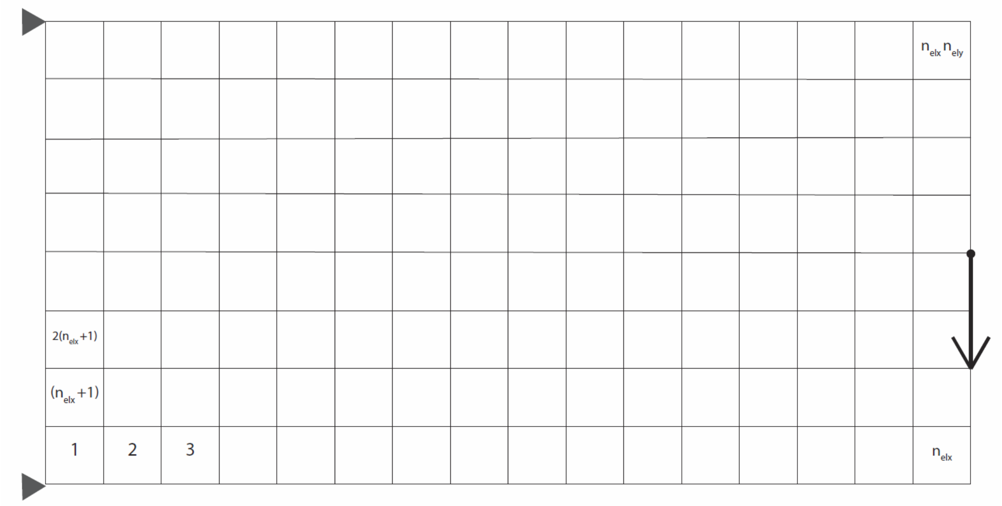


Figure 4.5: Discretized domain with element numbering.

The mapping of the elements in the domain will be done by function `funmap`. This function will receive as input the number of the element or the (i, j) position of the element and will convert it respectively to the (i, j) position or element number. Having the position of the element, the number will be assigned in the following way,

$$k = j + (i - 1) \cdot n_{elx}. \tag{4.8}$$

Knowing the number of the element one can retrieve the position, using,

$$\begin{aligned} i &= ceil(\frac{k}{n_{elx}}) \\ j &= k - n_{elx} \cdot (i - 1), \end{aligned} \tag{4.9}$$

where `ceil` is a MATLAB$^{\circledR}$ function which rounds the value to the nearest integer greater than or equal to the value obtained.

Instead of using a connectivity matrix to retrieve the neighbours of the present element one decided to number the elements as explained. The given numbering of the elements allows to avoid the use of a connectivity matrix decreasing the computation time. Being the dimension of the connectivity matrix $N_{elem}XN_{elem}$, where N_{elem} is the number of elements which the domain has, with the increase of the number of elements the handling of this matrix would be quite difficult, requiring a large amount of memory available. The computation time would increase because of the generation of this matrix and because for every step taken by an ant one would need to search the neighbours by looking the given element in the connectivity matrix. For a regular domain the implementation of the given numbering helps to reduce the computation time, since for a given element, using some rules, one can know the numbers of the neighbours elements.

In Kaveh et al. [2008] the pheromone matrix is initialized assigning a constant value to each element of the design domain. Having a constant of pheromones for each element will make it difficult to select the most important elements in the first iterations. In order to make the convergence faster and turn the solution generations for the first iteration faster, one decided to implement a pheromone initialization similar to the one done by Luh and Lin [2009].

The pheromone matrix in Luh and Lin [2009] is initialized using the normalized stress value assuming that all the elements have material present. One decided to initialize the pheromone matrix assuming that all the elements were also present but. instead of using the normalized stress value, the pheromone were initialized using the pheromone update rule (equation (4.7)). Assuming a solution which contains all the elements present in the design domain, is done a FEM analysis of the full domain with the loads and constraints present in the optimization problem. After retrieving the strain energy from each element, the pheromone intensity will be given by the Pheromone Update rule, being $\tau_{ij} = \Delta\tau_{ij}$.

Initializing the Pheromone matrix using the Pheromone Update rule instead of the normalized stress value will make the values obtained for the pheromone intensity in the initialization process to have a magnitude closer to the ones that will be obtained in the following iterations.

4.2.3 ACO Main

The optimization algorithm ACO is defined within the function `aco_main`. After having the pheromone matrix initialized one can start the optimization process.

One will start by assigning a random element of the domain to each ant. This will be done by generating a random integer which belongs to the interval [$1\ ;\ n_{elx} \cdot n_{ely}$].

Having all the ants an assigned element, the path construction will start. The elements that the ants can choose correspond to the ones that belong to the neighbourhood of that element. In order to not have a disconnect solution and a void post-processing of the solution generated, one demanded that the elements which the ant could choose would only be the ones that share one edge with the present one. Requiring that the elements must share a edge in order to be a candidate as next element will guarantee that the solution generated will not be disconnected, *i.e.*, all the elements connected by at least one edge, and not by a single point. If the elements were only connected by a single point the structure would not be able to withstand torques.

In Kaveh et al. [2008] it is used an 8- elements neighbourhood, which will demand for each solution a connectivity check. If a disconnected solution is found one might add elements in order to make it connected or discard it. In the present implementation one used a 4-elements neighbourhood to avoid the connectivity analysis.

The 4-elements neighbourhood can be seen in Figure 4.6. Being (i, j) a generic point of the design domain, the grey elements correspond to valid candidates for the next step of the ant from (i, j).

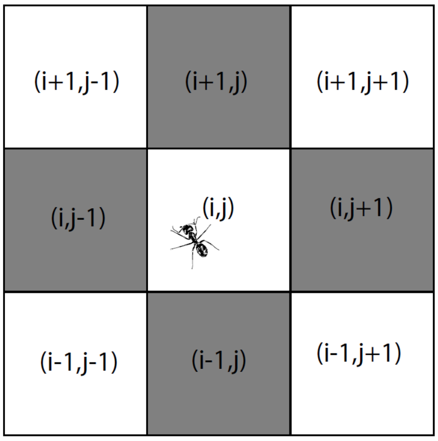


Figure 4.6: Elements connectivity on the optimization problem.

From these 4 elements, only 3 are candidates to be the next element in the ant's path, one will not allow the ant to choose the previous element visited. This will avoid the ant to get stuck between two elements.

Knowing the 3 possible elements that the ant can choose as next element one will apply the Element transition rule (equation (4.6)), this process will be conducted by the function `trans_rule`. After having the probability of choosing each element, one will calculate the probability density function ($P(e_{ij})$). A random number within the interval [$0, 1$] will be generate using the MATLAB® function `rand`. If $rand\_num \leq P(e_{ij})$, the element e_{ij} will be selected.

In order to keep the code fast, one decided that before checking if the boundary conditions were met and the volume fraction constraint was not surpassed one would check if the element selected was present in the ant's path. If the element was present in the ant's path the volume fraction and the boundary conditions will remain the same as they were when the ant selected the previous element. If not, one must check the if the boundary conditions were met and the volume fraction constraint was not surpassed.

Having checked the volume fraction constraint and boundary conditions, there are three possible scenarios. The most common is, the boundary conditions were not met and volume fraction constraints was not surpassed, so the ant will select another element. Another one is the ant's path was surpassed the volume fraction constraint and the path will be cleared and a new initial position for the ant will be assigned. Lastly, the boundary conditions were met and the volume fraction has reached the given value, in this case the ant search will stop [Wu et al., 2009].

Having all the ants finished the construction of the paths, one can perform the FEM analysis of the given solution.

4.2.4 Noise Cleaning Filter

In order to improve the results generated by ACO for a TO problem Kaveh et al. [2008] introduced a noise cleaning filter during the pheromone update, to prevent the formation of tiny members in the structures obtained.

The noise cleaning filter will change the strain energy of the element, modifying the strain energy of an element with the strain energy of the neighbour ones. The modified strain energy will be given by

$$\hat{U}_i = \frac{\sum\limits_{e=1}^{n^i} H_e U_e}{\sum\limits_{e=1}^{n^i} H_e}, \tag{4.10}$$

where H_e is the filter being given by

$$H_e = V_e \left[r_{min} - r(i,e) \right], \; e \in \left\{ 1, 2, ..., n^i \right\}, \tag{4.11}$$

where V_e is the volume (or area in a 2D problem) of the element e, r_{min} is the minimum size allowed for the structure members, $r(i,e)$ is the distance between element i, which strain energy is being modified, and element e, a neighbour of element i that satisfies $r(i,e) \leq r_{min}$, n^i is the number of elements that satisfy the last condition.

The introduction of the filter modifies the strain energy of the elements, being $\hat{U}_i$ the new strain energy of the element i. For any given elements of the domain even if this element does not belong to the solution found, depending of the r_{min} and the surrounding elements, this element might have a strain energy ($\hat{U}_i$) different from zero. Being the value of the modified strain energy related linearly with the distance between elements, the change that an element will cause in the surrounding element will decrease as the distance for the element increases.

The filter implementation is accomplished by changing the strain energy of the element by the modified strain energy obtained using equation 4.10 during the pheromone update. The linear spread of the strain energy across the surrounding elements will also spread the pheromone in the surrounding element which will make the ant travel across more elements in the neighbour of the present element when compared with the case where the filter is not used. Ultimately, it will lead to the formation of larger members making the tiny ones will disappear from the solutions.

4.3 ANSYS®/MATLAB® Interface

The finite element analysis will be conducted by the finite element tool **ANSYS APDL**®. The interface between FE tool and the code developed will be done by using input files generated in MATLAB® and them calling ANSYS® in batch mode.

The batch command used was the following:

```
ansys145 -p ane3fl -dir "D:\ACO\ANSYS"
-j "ants" -s read -l en-us -b -i "D:\ACO\input.txt" -o "D:\ACO\ANSYS\discard"
```

Being `ansys145` the executable to run APDL®, the first path corresponds to the job folder, the second path is the input file generated by MATLAB®, being the last one the output folder.

The FE analysis will be conducted by the function `fe_analysis`, generating the input for each ant and calling the ANSYS® in batch mode.

In order to define the elements that will be present in the current solution being tested, one used the ANSYS®' **Birth/death element** function [ANS, 2013]. This function will ”kill” the selected element, meaning that to the given element will be assigned a small value to the element stiffness. By default this value is equal to 10^{-6}, the value used is not equal to zero in order to avoid ill-conditioned matrices. The elements that are not present in the ant's path will be deactivated using the birth/element function.

Having the strain energy of each element of all the solutions found the Pheromone matrix will be updated using the Pheromone update rule (equation (4.7)). Having the updated the Pheromone matrix, one will have another iteration, until the number of iterations initially defined has been reached.

4.4 Literature Case

Having the code developed, one needs to solve a simple case in order to validate its performance and be sure that the outcome of the optimization is valid. In Kaveh et al. [2008], two 2D problems were solved, being one of these cases selected to compare the output of the code developed. The one selected is the cantilever beam with a point load at its extremity (Figure 3.4). The modulus of elasticity (Young's modulus) and the Poisson's ratio used are equal to $7.9 \cdot 10^9$ and 0.30, respectively. These values correspond to the values used by Kaveh et al. [2008].

4.4.1 Force Selection

The force used during the optimization process needs to be chosen carefully. If a force with a large magnitude is selected, one might not get valid results, since the FEM analysis conducted is a linear analysis. Multiple finite analysis with the full domain were performed and the displacements obtained were registered. From the value of the maximum displacement one can conclude if a linear analysis is valid or not, being valid only for small displacements.

Table 4.2 displays the values of the strain energy and maximum displacement for multiple forces.

Force [N]	10^4	10^5	$5 \cdot 10^5$	10^6	10^7
Strain energy $[J]$	3.20	$319,82$	7995.43	31981.70	$3.20 \cdot 10^6$
Maximum Displacement $[m]$	$6.40 \cdot 10^{-4}$	$6.40 \cdot 10^{-3}$	$3.20 \cdot 10^{-2}$	$6.40 \cdot 10^{-2}$	0.640
Max. Disp./Y dimension $[\%]$	0.16	1.60	8.00	15.99	159.91

Table 4.2: Strain energy for the full domain.

For a force equal or superior to 10^6 $[N]$, the maximum displacement of the full domain, when compared with its Y dimension $(0.4\ m)$, has the same order of magnitude, therefore one can no longer consider as being a small displacement and the linear analysis conducted is no longer valid. The optimization process conducted with the code developed for the given domain (Figure 3.4) must be done with a force lower or equal to $5 \cdot 10^5$ $[N]$.

4.4.2 Literature Optimum

In order to compare the results obtained using the code developed with the results obtained by Kaveh et al. [2008], one modelled the optimum structure, Figure 4.7, and performed a finite element analysis to have a value of the strain energy of the structure.

The optimum solution obtained by Kaveh et al. [2008], using the same parameters as the one in Table 4.1 with a noise cleaning filter during the pheromone update, is shown in Figure 4.7.

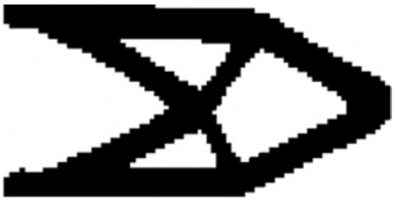


Figure 4.7: Optimum solution for the cantilever subjected to a point load [Kaveh et al., 2008].

Now, performing a FEM analysis for the optimum solution one obtained the following strains displayed in Table 4.3.

Force [N]	10^4	10^5	$5 \cdot 10^5$
Strain energy $[J]$	4.89	488.75	12218.70
Maximum Displacement $[m]$	$9.78 \cdot 10^{-4}$	$9.78 \cdot 10^{-3}$	$4.89 \cdot 10^{-2}$
Strain Full Domain/Optimum $[\%]$	65.44	65.44	65.44

Table 4.3: Strain energy for the optimum solution found in Kaveh et al. [2008].

4.5 Results

The solution of Kaveh et al. [2008] for the given problem was obtained using a noise cleaning filter to avoid the formation of undesirable tiny members. One started by performing the optimization process without using the noise cleaning filter and using the same parameters as the one used by Kaveh et al. [2008], given in table 4.1, corresponding to the default parameters of the code. The force was set equal to $10^4\ N$ to conduct all the optimizations shown bellow.

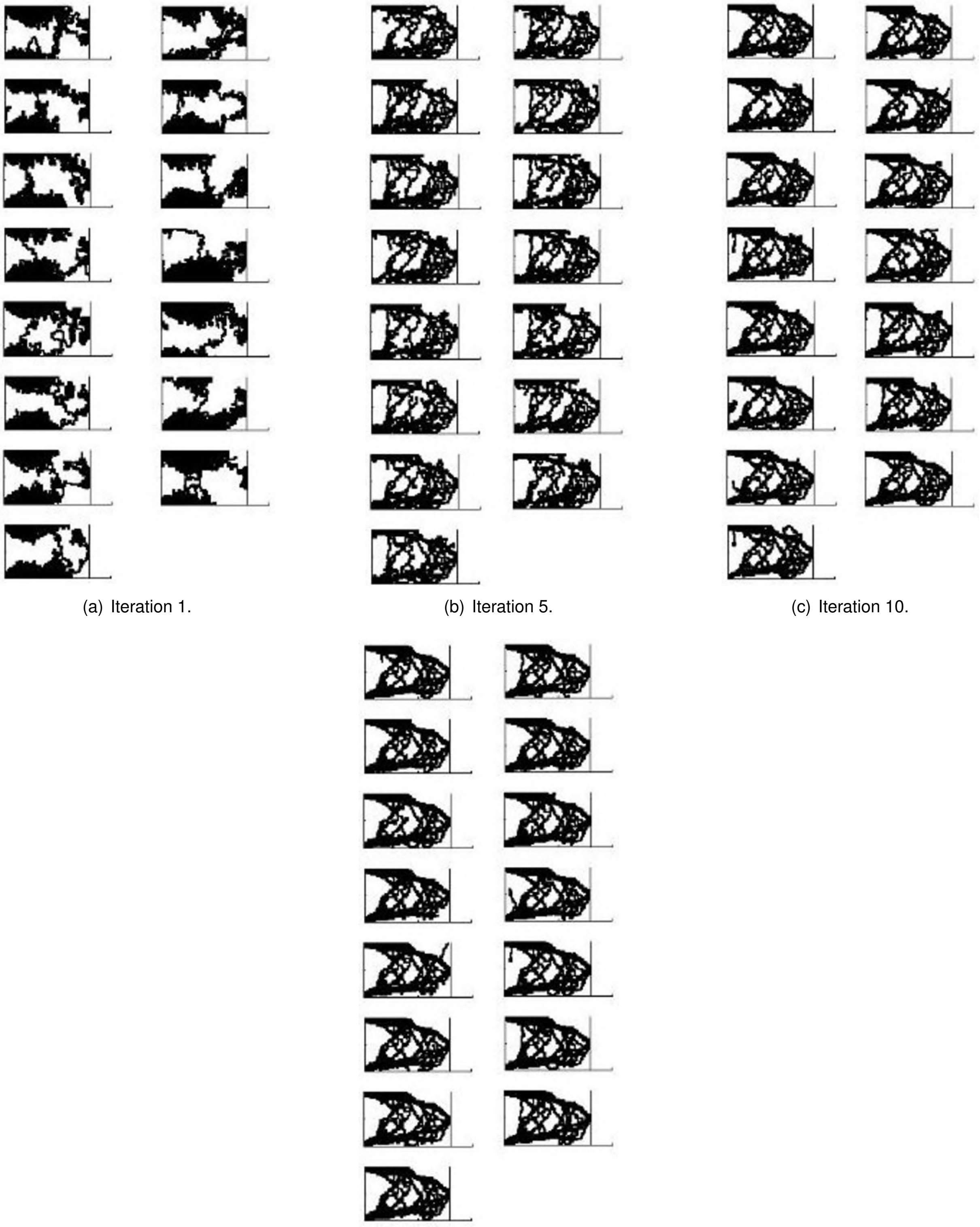
Figures 4.8 and 4.10 show the evolution of the solutions found throughout the optimization process. As it can be seen, at the end of the first iteration the solutions found have few features in common between them, having a large concentration of elements in some areas of the design domain. The designs found display a poor performance due to the distribution of material in the design domain being far away from the optimal one. The best design in the given iteration displayed a strain energy equal to $72.122J$.

Being the first iteration, the ants will tend to choose the elements which have the largest strain energy from the surrounding ones, as a result of the pheromone initialization being made using the strain energy of the design domain full. Lacking the information of previous iterations, results in the generation of designs which will present a poor performance.

From the first iteration to the fifth iteration, one can notice that all the designs found show a resemblance between them. The layouts at this point start to get the shape of a truss-like structure, displaying some members which are constituted in width only by a element. These members will affect the performance of the structure, resulting in a strain energy equal to 8.165 J which is still far from the minimum value of strain energy obtained.

From iteration 10 until the end, the solutions obtained suffer small changes in topology, indicating that the solution found by the algorithm has converged. As it can be seen in Figure 4.11, the minimum strain energy obtained for each iteration throughout the optimization process shows a decreasing trend in the first iterations, remaining approximately constant during the last ones. From Figure 4.11, one removed the minimum strain energy from the first iteration to have a better resolution for the last iterations. The behaviour of the minimum strain energy confirms that the designs throughout the last iterations suffer small changes. As it can be seen in Table 4.4, the strain energy has converged. From iteration 29 to 30 the strain energy changes $0.006\,J$, which correspond to a relative change of 0.1%.

The best topology found at the end of the optimization process displayed a strain energy equal to 5.786 J being 15.53% larger than the solution found by Kaveh et al. [2008]. As one said before, the



(a) Iteration 1.

(b) Iteration 5.

(c) Iteration 10.

(d) Iteration 15.

Figure 4.8: Solutions obtained without filter for iteration 1,5,10 and 15.

Iteration	28	29	30
Strain energy $[J]$	5.828	5.792	5.786

Table 4.4: Minimum strain energy obtained for the last 3 iterations.

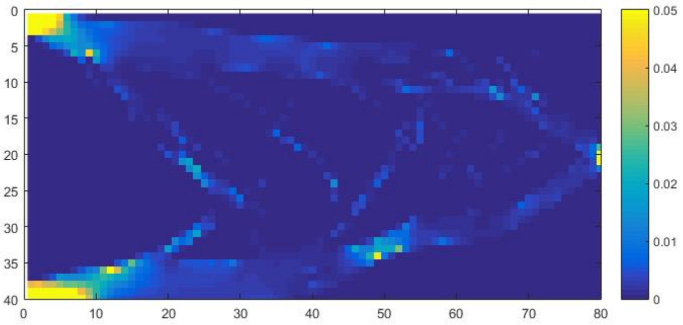
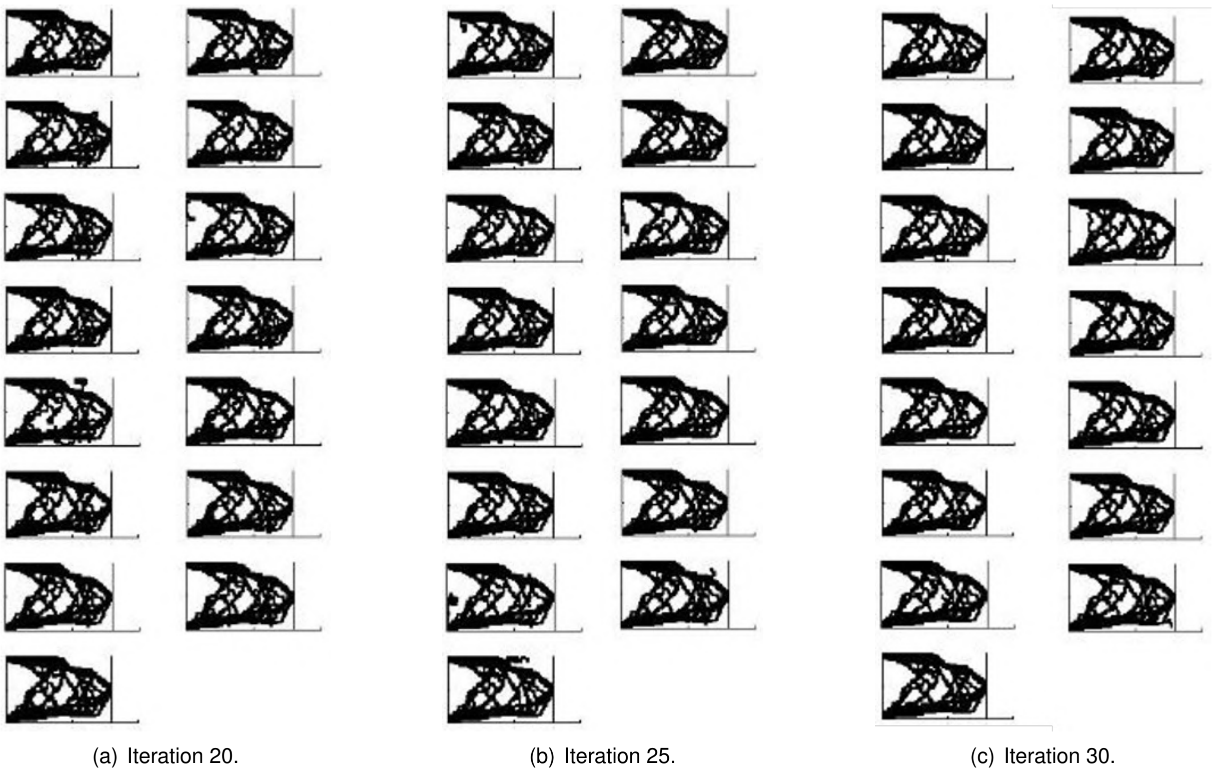


Figure 4.9: Colormap of the normalized pheromone matrix for iteration 30.



(a) Iteration 20.

(b) Iteration 25.

(c) Iteration 30.

Figure 4.10: Solutions obtained without filter for iteration 20, 25 and 30.

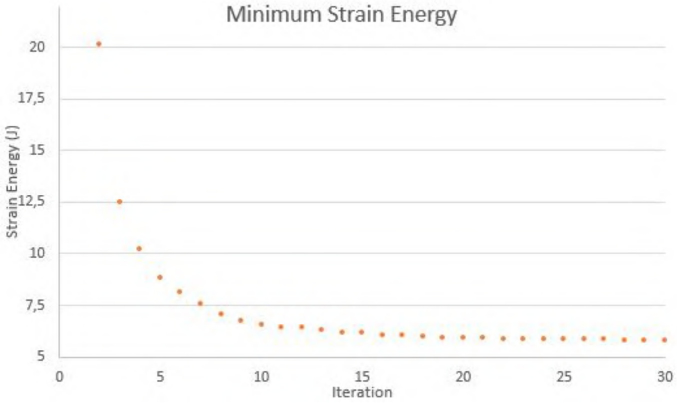


Figure 4.11: Minimum strain energy for each iteration.

optimization process was conducted without the use of the noise cleaning filter, being the literature optimum obtained with the application of this filter. The introduction of the filter will reduce the many small members that the structure displays, to form bigger member which will lead to a stiffer structure, reducing the overall strain energy. This can explain the difference between the values of strain energy. Furthermore, when making a visual comparison between the structures obtained (Figure 4.10 (c)) with the one from the literature (Figure 4.7), one can notice that the literature optimum displays well defined members in the middle, whereas, the optimum solutions found have many members composed by few elements. The literature solution also displays symmetry in relation with the longitudinal axis, being the point load applied at the middle of the cantilever one expected than the optimum structure also displayed symmetry. The structures obtained do not display symmetry, this fact can be explained by the heuristic behaviour of the algorithm.

In Figure 4.9, it is plotted a colormap of the pheromone matrix at the end of iteration 30. In order to have a more explicit representation of the data, one normalized the values of the pheromone matrix. After the normalization of the matrix, one noticed that there was a big discrepancy between some values of the matrix, as a result, a scale for the colormap between $[0, 0.05]$ was chosen. The elements highlighted in a yellow color have a pheromone intensity equal or superior to 0.05.

The distribution of pheromones across the design domain resembles the solutions found. Indeed, most of the times, the ants will choose the elements which have the highest concentration of pheromones, so the ant's path will follow closely the pheromone distribution.

4.5.1 Parametric Study

A parametric study of the parameters that rule the ACO algorithm was performed. When comparing with the implementation for a TS problem one can notice that the parameter β is missing. As has been noted, in the given implementation the "visibility" information is missing due to the constrains of the problem itself. On the other hand, one has the introduction of the λ, which will control the influence of the strain energy during the pheromone update.

To perform the parametric study, one kept all the parameters constant just changing one of them at a time. The parametric study of the parameters λ, α, ρ and the `col_dim` (Colony Dimension) were performed without the noise cleaning filter. With the introduction of the noise cleaning filter one introduced the r_{min} parameter, which will define the smallest dimension of the formed members, being a parametric study of this parameter also performed. The standard values chosen for the parametric study were the one present in Table 4.1.

One started by performing a parametric study of the parameter λ, the importance of this parameter was explained before.

λ	1	1.5	2
Strain energy $[J]$	8.381	6.044	5.785

Table 4.5: Minimum strain energy for λ.

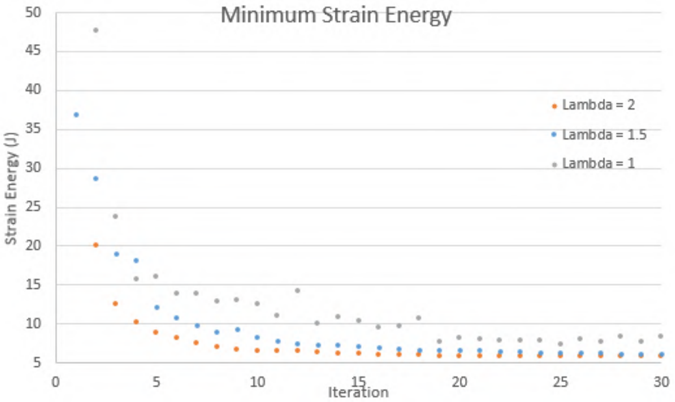
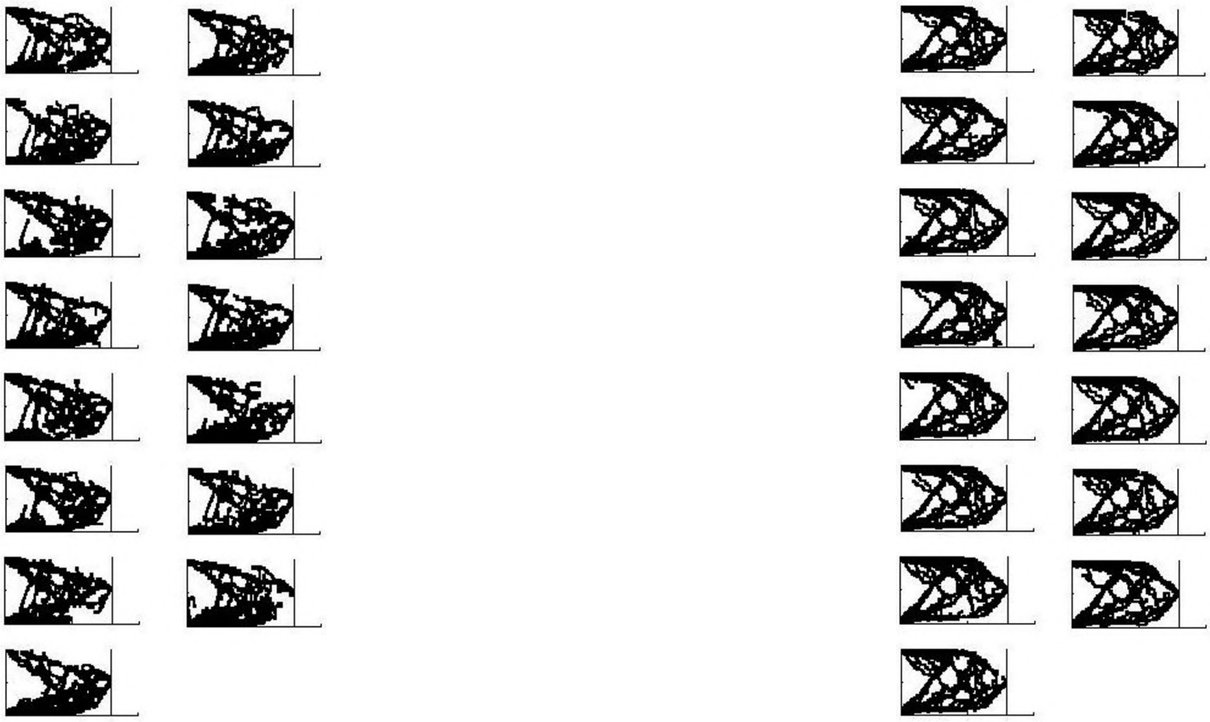


Figure 4.12: λ parametric study.



(a) λ = 1.

(b) λ = 1.5.

Figure 4.13: Solution for different λ.

Looking at Figure 4.12, one can conclude that with the increase of λ the structures obtained will have a better performance. The increasing λ will lead to a lower minimum strain energy, as seen in Table 4.5. By increasing λ from 1 to 1.5, one decreased the strain energy by $2.338\ J$, which represents a 27.90% decrease. When comparing the results for $\lambda = 2$ and $\lambda = 1$, one decreased the strain energy by $2.595\ J$, representing a 30.96% decrease. The difference in the results obtained for $\lambda = 2$ and $\lambda = 1.5$, have a smaller discrepancy when compared with $\lambda = 1$, but an improvement can also be seen. The strain energy decreased by $0.257\ J$, corresponding to a 4.25% decrease.

The influence of λ is not only prescribed to the minimum strain energy obtained, influencing also the converge of the algorithm. For a smaller λ, one can notice that the algorithm will require more iterations in order to convergence, furthermore for $\lambda = 1$ the algorithm displays some fluctuations in the minimum strain energy value. With the increase of λ, the algorithm had a faster convergence without displaying this fluctuations in the value.

The structure obtained (Figures 4.13 and 4.10 (c)) for the different values of λ will also display different features. For $\lambda = 1$, the final structures look really irregular, displaying many holes and being composed by small members. The overall appearance of the structures when comparing with the literature optimum have small resemblances. The resemblances between the literature optimum and the structure obtained with the ACO implementation increase with the increase of λ. However, the structures lack the well defined member in the middle, displaying instead many tiny members. One can notice a reduction in the number of small members present in the structures with the increase of λ. The reduction of this members can explain the increasing performance of the structures.

The colony dimension (`col_dim`) was the next parameter to be studied.

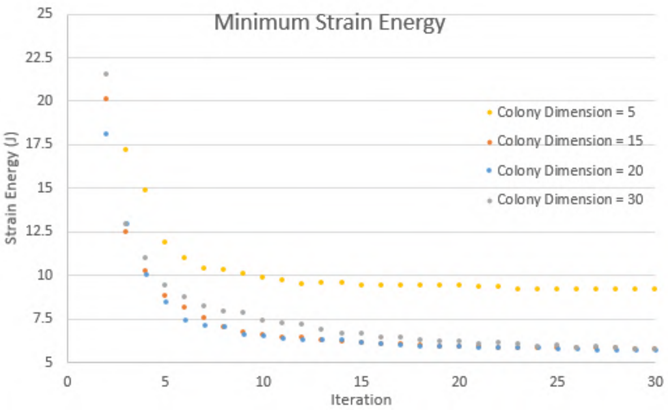


Figure 4.14: Colony dimension parametric study.

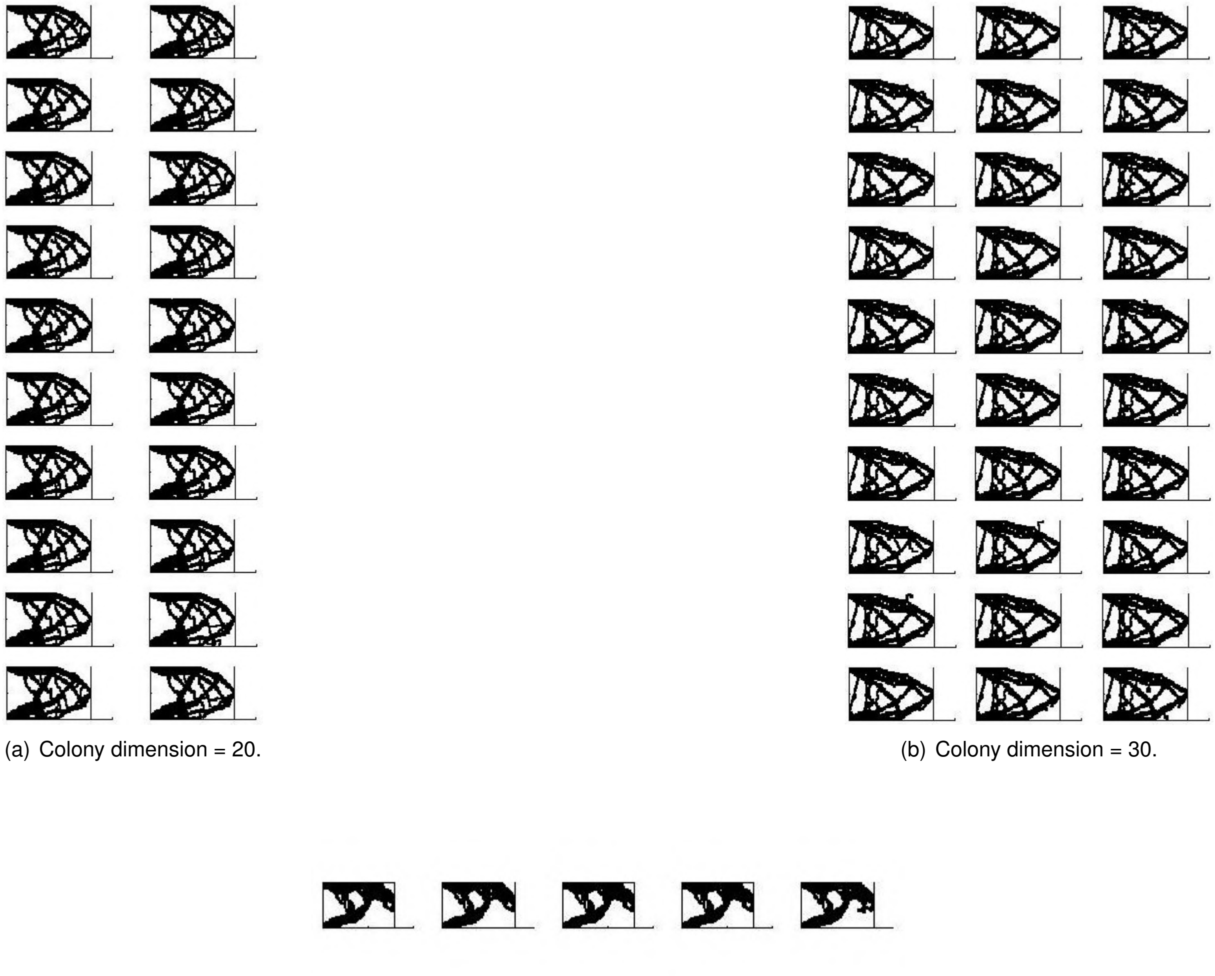
`col_dim`	5	15	20	30
Strain Energy $[J]$	9.226	5.786	5.718	5.775

Table 4.6: Minimum strain energy obtained for different values of `col_dim` at iteration 30.

For a `col_dim` = 5, one obtained the highest value of strain energy (Table 4.6), being 3.439 J (37.28%) higher than the value obtained for `col_dim` = 15. For `col_dim` = 15, 20 and 30, the minimum strain energy suffered small changes having a value really close for the three cases, being the biggest difference in strain energies equal to 0.0688 J, which corresponds to a decrease equal to 1.19% from `col_dim` = 15 to 20.

For a small colony dimension the layouts generated at each iteration will not be enough to find the optimal solution. One can notice this fact for the case where `col_dim` = 5, the solutions found are far from the optimal ones that the algorithm is capable of finding. Due to the small dimension of the colony during the search procedure, the algorithm got stuck in a local minimum. With the increase of the colony, one will have a wider range of search avoiding getting stuck in a local minimum, explaining why the value of strain energy is practically constant for a colony dimension bigger or equal than 15.

The structures obtained with `col_dim` = 5 (Figure 4.15 (c)), when compared with the ones obtained with different values of `col_dim`, display different features. One can observe that the region where the



(a) Colony dimension = 20.

(b) Colony dimension = 30.

(c) Colony dimension = 5.

Figure 4.15: Solutions for different `col_dim`.

force is applied is just connected to the main structure by a member, which will decrease its performance, increasing it strain energy. For the other values of `col_dim` (Figure 4.10 (c), 4.15 (a) and (a)), the structures obtained all display a truss-like aspect showing only small differences in their topology. The similarity between structures was expected due to the close values of strain energy obtained.

The parametric study of the α parameter was not possible to perform. When α was increased to 1.5 the algorithm was not able to find solutions that would meet the constraints for the first iteration, even after more than 24 hours. Considering that the running time required to complete the optimization process ranged from 4 to 8 hours depending of the different parameters selected for the optimization, one concluded that it was not feasible to use a higher value of α.

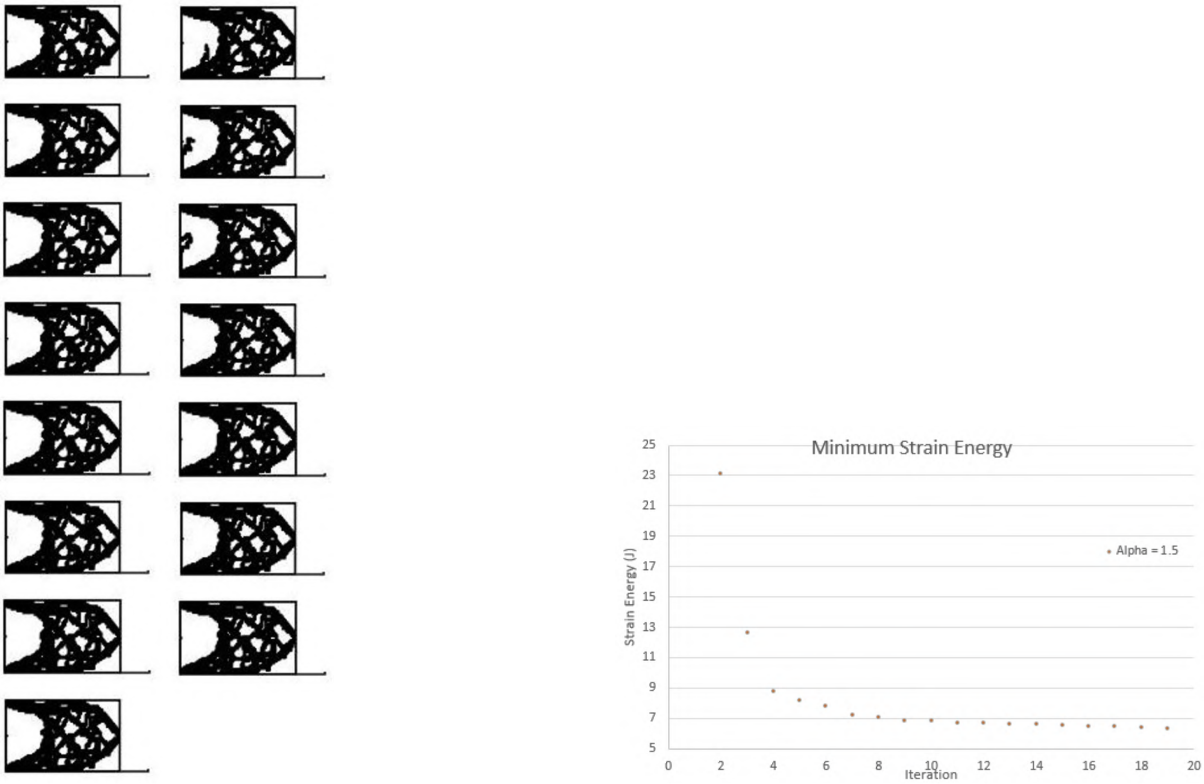
In order to see the effect that the changing α had in the algorithm one assigned $\alpha = 1$ for the first two iterations of the optimization process, increasing its value to 1.5 on the following iterations. This would allow for the algorithm to find the solutions for the first two iterations, doing the pheromone update which would guide the ants during the following iterations to find feasible solutions in a more expedite way.

Changing the α after the second iteration, resulted in a faster solution generation comparing with run

Iteration	17	18	19
Strain Energy $[J]$	6.450	6.414	6.307

Table 4.7: Minimum strain energy obtained for iteration 17, 18 and 19.

where one defined $\alpha = 1.5$ from the beginning. However, the decrease in time for the solution generation occurred only for the first 10 iterations. After this point, the time required to complete a iteration increased from iteration to iteration, until the code got stuck at iteration 20. Throughout the iterations, one expects that the time required to generate the solution will decrease due to the accumulation of pheromones in a set of elements which form a feasible solution. So, the ants during the path construction have a path well defined by the pheromone distribution across the design domain. However, this was not observed for $\alpha = 1.5$.



(a) Solutions found for iteration 19.

(b) Mininum strain energy.

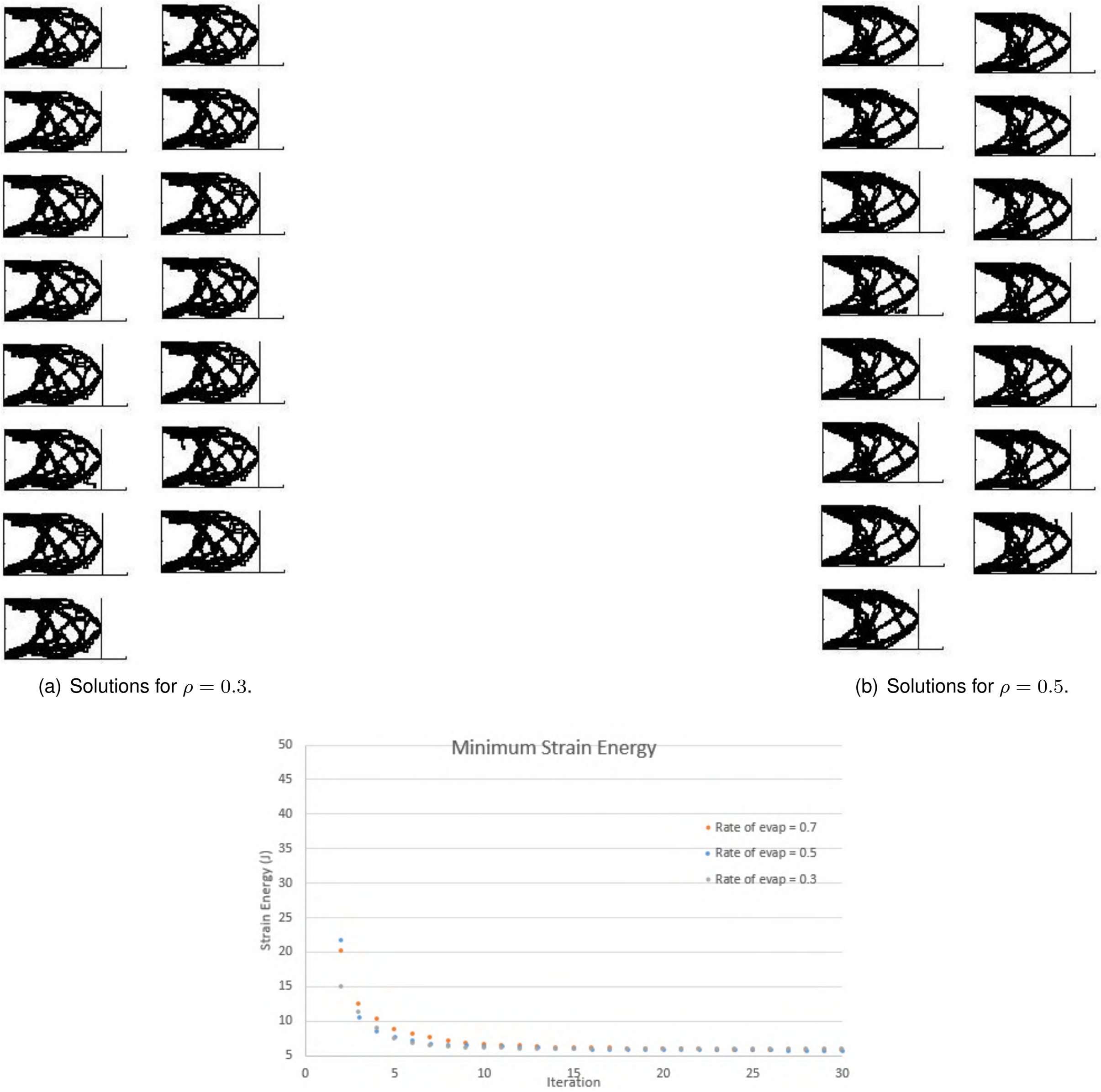
Figure 4.16: $\alpha = 1.5$.

The solutions found after the algorithm completed 19 iterations are represented in Figure 4.16 (a).

As can be seen in Figure 4.16 (b), the minimum strain energy decreased throughout the iterations, although for the last iterations the magnitude of the decrease is small (Table 4.7), hinting the convergence of the solutions found.

Before the noise cleaning filter implementation, the parametric study of the ρ parameter was performed.

The structures obtained (Figure 4.10(c), 4.17 (a) and (b)) for the different values of ρ display some resemblances. This observation is supported by the value of minimum strain energy obtained for the



(a) Solutions for $\rho = 0.3$.

(b) Solutions for $\rho = 0.5$.

(c) Mininum strain energy.

Figure 4.17: ρ parametric study.

three cases.

ρ	0.3	0.5	0.7
Strain Energy $[J]$	5.912	5.751	5.786

Table 4.8: Minimum strain energy obtained for different values of ρ.

From the strain energy for different values of ρ (Table 4.8), one can conclude that the rate of evaporation will have a minor influence in the final solution obtained.

Analysing Figure 4.17(c), one can notice that with the decrease of ρ one has a slight quicker convergence. This can be explained by the role that the rate of evaporation has during the pheromone update. This parameter is used to simulate the evaporation of the pheromones, which will vanish with time. ρ will give us the fraction of pheromone intensity that remained from the previous iteration.

The increase of the rate of evaporation (decrease of ρ) will require a lower number of iterations for the initial pheromone distribution to fade allowing for the trail of pheromones, that will lead to the optimum solution, to be highlighted.

Having the noise cleaning filter implemented, a parametric study of the r_{min} parameter was performed. As one said before, r_{min} will define the smallest size that a member from the structure can display. In order to have a more explicit way of presenting r_{min}, one decided to represent r_{min} as a function of the number of elements that the member should have instead of a dimension (d_{min}),

$$d_{min} = r_{min} \cdot \frac{dim_y}{nel_y}, \tag{4.12}$$

being dim_y the y dimension of the domain and nel_y the number of elements in the y dimension.

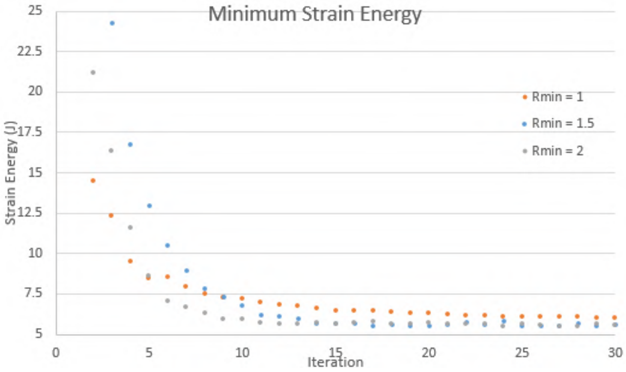


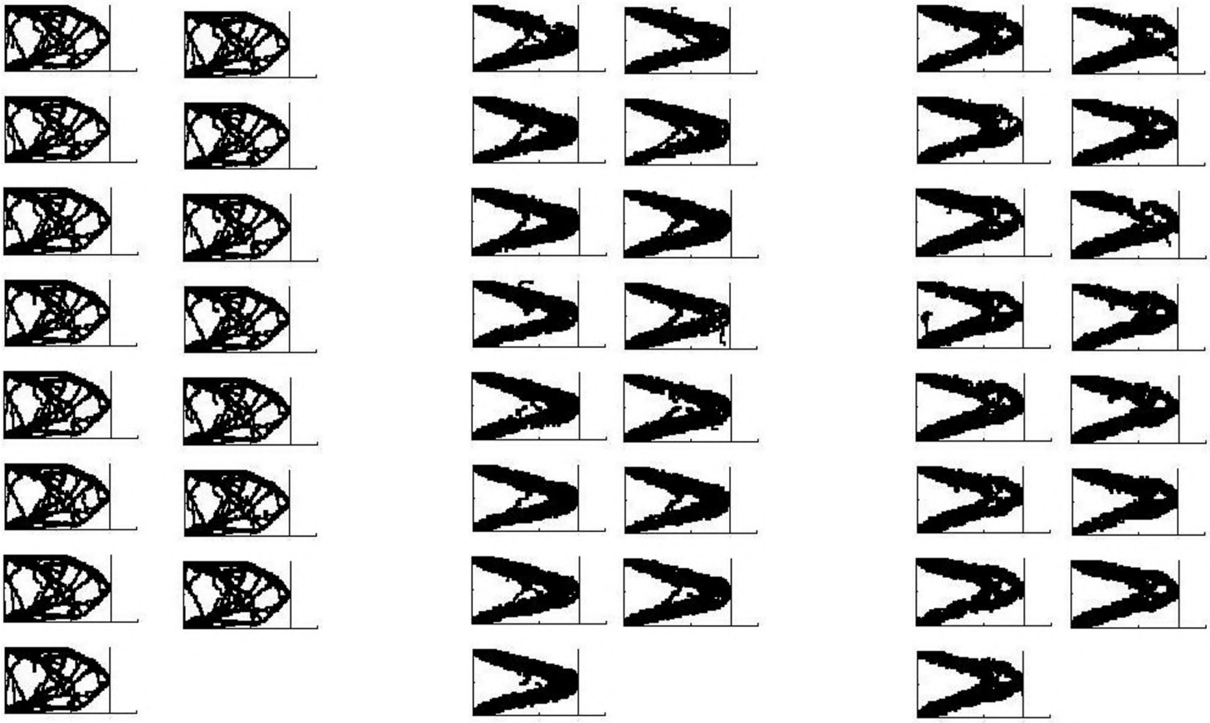
Figure 4.18: r_{min} parametric study.

As it can be seen in Table 4.9, the introduction of the filter led to a decrease in the strain energy for a r_{min} bigger than 1. For $r_{min} = 1.5$, there was a reduction of 0.193 J, being equal to a relative decrease of 3.34%. The decrease for $r_{min} = 2$ was equal to 0.205 J, corresponding to a relative decrease of 3.55%.

$\mathbf{r_{min}}$	1	1.5	2
Strain Energy $[J]$	6.0289	5.5931	5.5812

Table 4.9: Minimum strain energy obtained for different values of r_{min} at iteration 30.

The structures obtained with the implementation of the filter display some differences from the ones obtained without it. With the increase of r_{min} the members that form the structures increased in size, and its number reduced. Being the final structures composed mainly by few members with a size defined by the value of r_{min}. For $r_{min} = 1$, the structure display many members when compared with the structures obtained with different values of r_{min}. For $r_{min} = 1.5$ and 2 the smallest members disappear from the structures, leading to the formation of wider members, being most of the elements placed in the outer members of the structures.



(a) $r_{min} = 1.$ (b) $r_{min} = 1.5.$ (c) $r_{min} = 2.$

Figure 4.19: Solutions for different r_{min}.

4.5.2 Optimum Solution

After having performed a parametric study, one changed the default parameters used in the literature case in order to find a better solution than the one found using the given parameters (Table 4.1), but keeping the volume fraction constraint (`vol_f = 0.45`), as summarized in Table 4.10.

`colony_dim`	15
`n_ite`	30
`evap` (ρ)	0.3
`alpha_p` (α)	1
`lambda` (λ)	2.2
r_{min}	1.2

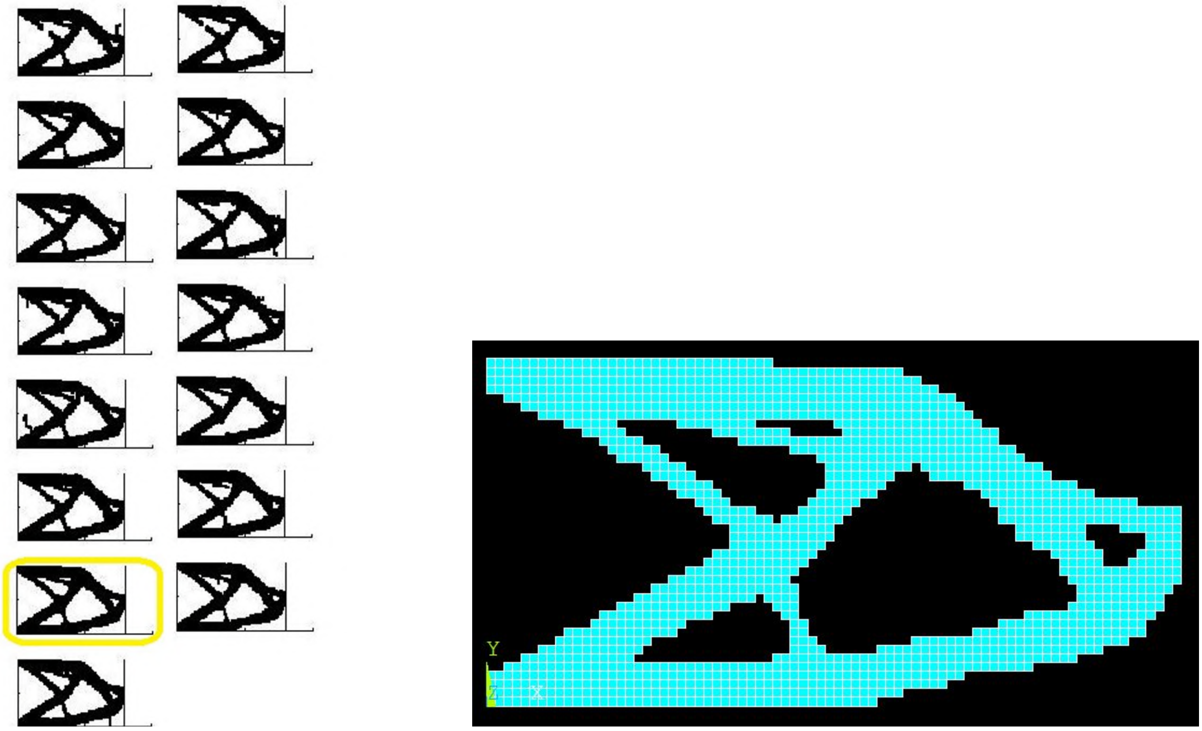
Table 4.10: ACO parameters used to obtain the best solution.

Using the information obtained with the parametric study, one decided to keep the colony dimension equal to the default value (`col_dim` = 15) because the increase of the colony dimension above 15 does not bring further benefits. Increasing the dimension of the colony above 15 ants will only increase the computation time required for the optimization run. For all the runs, one observed that the algorithm at the end of the runs had converged, so one keeps the number of iterations that the code would perform.

The value of ρ was decreased from 0.7 to 0.3 in order to avoid getting stuck in a local minimum.

Lastly, the value of λ was slightly increased and the r_{min} was set equal to 1.2 elements. The performance of the algorithm increased with the increasing λ, being that the reason why one increased this parameter to 2.2. One restricted the increase to 0.2, to avoid an early convergence to a suboptimal solution, due to the higher importance that the algorithm will give to the elements with a higher strain energy. The noise cleaning filter was used to find the close to optimum solution being the value of r_{min} set equal to 1.2 in order to avoid the formation of small members, avoiding also the formation of only two exterior members, as seen in Figures 4.19 (b) and (c).

Using the given set of parameters, the best solution for the run corresponds to the 13th ant, which corresponds to the solution highlighted in Figure 4.20 (a), being Figure 4.20 (b) a closer look of the structure.



(a) Solutions found at iteration 30.

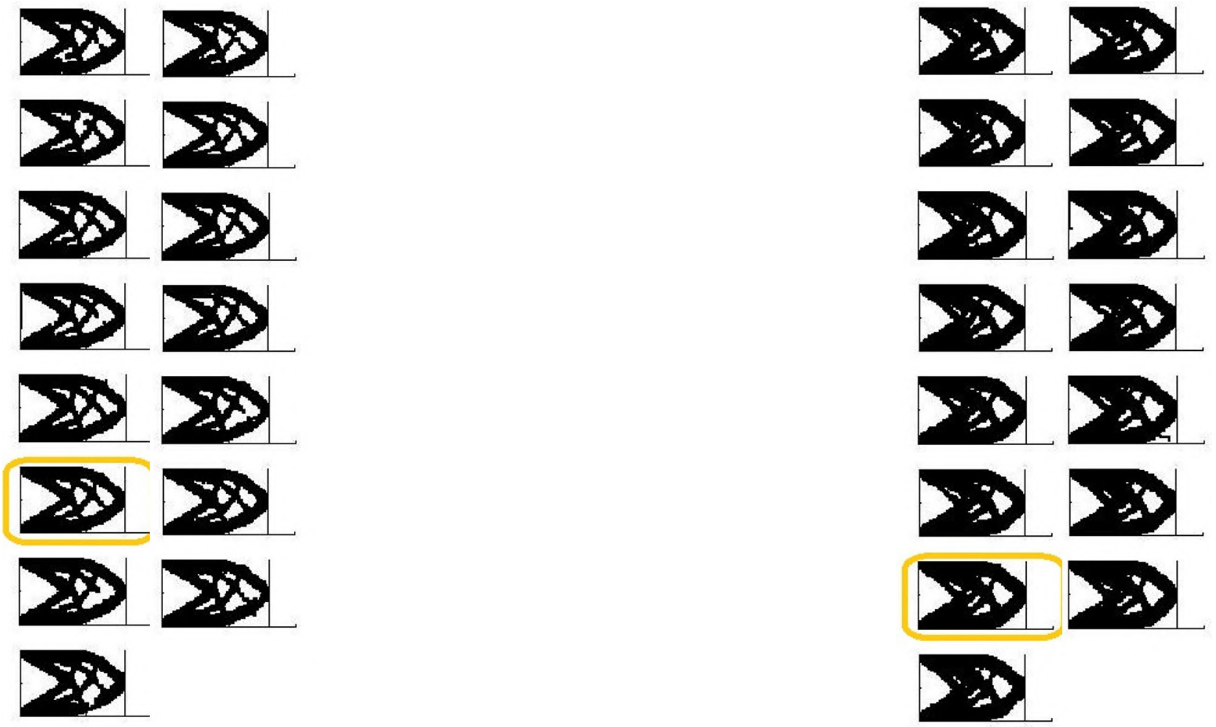
(b) Best solution found.

Figure 4.20: Best solutions found using ACO.

The best solution displayed a strain energy equal to $4.875\ J$, matching the strain energy of the best solution found by Kaveh et al. [2008] ($4.887\ J$). The solution displays some resemblances with the literature one, such as the overall shape and the central member which increases the stiffness of the structure. On the other hand, the solution obtained does not display symmetry as the one from the literature. The lack of symmetry, as previously explained, is due to the random nature of the ACO metaheuristic algorithm..

Keeping the parameters used to obtain the above solution (Table 4.10), one increased the volume fraction `vol_f`. As one explained before, without a volume fraction constraint, the minimum strain energy will be obtained when the domain is full. Removing material from the domain, will lead to an increase in the minimum strain energy. Assuming that for each value of volume fraction, one is able to reach the optimum material distribution for the given problem. Decreasing the volume fraction will increase the minimum strain energy for the optimum solution.

The solution obtained for different values of `vol_f` can be seen in Figure 4.21. The layouts highlighted correspond to the best solutions found by the algorithm at the end of the optimization process.



(a) vol_f = 0.55.

(b) vol_f = 0.65.

Figure 4.21: Solutions for different values of vol_f.

vol_f	0.45	0.55	0.65
Strain Energy $[J]$	4.875	4.077	3.657

Table 4.11: Minimum strain energy obtained for different values of vol_f at iteration 30.

In Table 4.11, the minimum strain energy obtained after the optimization process for different values of volume fraction is shown. The results support what has been stated before, the minimum strain energy decreases with the increase of the volume fraction.

Chapter 5

High Aspect Ratio Wing Model

The wing used for the topology optimization problem that will be solved is based on the NOVEMOR project of the EU 7^{th} framework reference wing. This wing was designed to present an high aspect ratio, $AR = 12$, while keeping the wing area, sweep, dihedral angle and the MTOW constants and equal to the reference wing [Spada, 2014].

5.1 Wing 3D Model (Solidworks®)

Planform Dimensions		
b	$[m]$	36.33
b_{r_b}	$[m]$	7.13
b_{b_t}	$[m]$	11.03
c_r	$[m]$	5.71
c_b	$[m]$	3.01
c_t	$[m]$	1.3
MAC	$[m]$	3.02
Λ	[°]	25
Γ	[°]	4.5

Table 5.1: Planform dimensions of the NOVEMOR 7^{th} framework reference wing with AR = 12.

Using the information from Table 5.1, one developed a 3D model in Solidworks® of the wing. During the development of the the 3D model some simplifications were made when comparerd to the model used in Spada [2014]. The engine was not included in the model, which led also to removal of the pylon and the additional rib present in the model to increase the stiffness, which allow the wing to support the additional weight of the engine. The model is composed by 3 main parts, the torsion/wing box, 14 ribs and the skin of the leading edge and trailing edge.

The middle part of the wing, usually called torsion box/wing box, which in a conventional wing with two spars corresponds to the volume delimited by the spars and the reinforced section of skin. The wing box was modelled as a solid, that is bounded by the upper and lower part of the airfoil and by two vertical lines that are at 20% and 75% of the airfoil chord, the layout of the cross-section of the wing box can be seen in Figure 5.1.

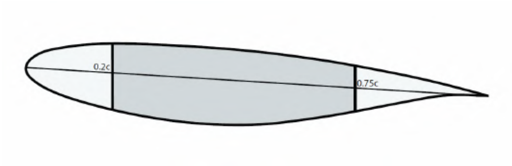


Figure 5.1: Root airfoil, grey area corresponds to the wing box cross-section.

Being the wing box modelled as a loft from rib $n-1$ to rib n, one defined the wing box cross-section at which rib using it as a profile for the loft resulting in the optimization domain seen in Figure 5.2.

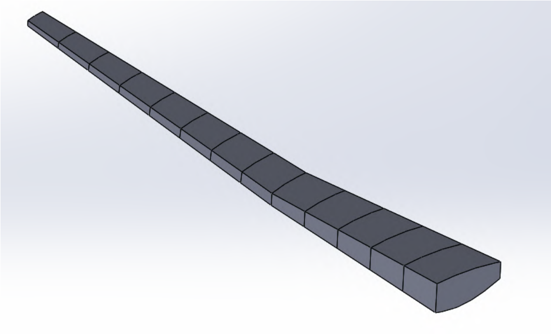
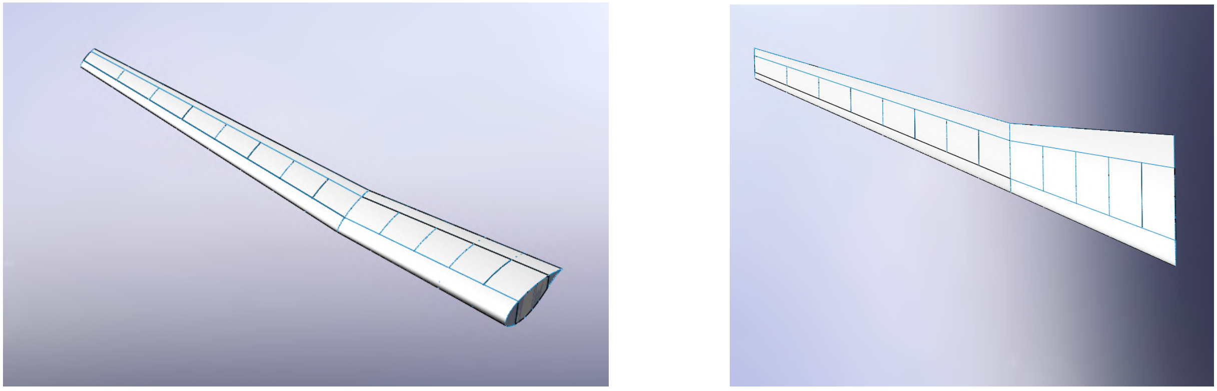


Figure 5.2: 3D Solidworks® model of the wing box.

The full Solidworks®) 3D model of the NOVEMOR 7^{th} framework reference wing with an $AR = 12$ can be seen in Figure 5.3.



(a) Isometric view.

(b) Top view.

Figure 5.3: Wing NOVEMOR 3D Solidworks®) model.

5.2 Flight Condition

In order to access the loadings to which the wing will be subjected, one requires the Cruise Condition. The cruise point is equal to $M = 0.78$ at an altitude of 38000 feet. Having the cruise point and assuming the ISA model, one can retrieve the properties of the air [Toolbox, 2015], summarized in Table 5.2.

Knowing the air speed and the Mach number one is able to calculate the module of the velocity of the undisturbed flow, being $V_\infty = 230.154\ m/s$.

Having the module of the velocity of the undisturbed flow one needs to calculate the angle of attack. In order to calculate AoA, one will assume a trim condition in cruise, meaning that the sum of forces will be equal to zero. Also, assuming that the aircraft is flying with a small AoA one will say that the lift required to trim the plane will be equal to the $MTOW = 568980\ N$, $L_{req} = MTOW$.

Density	D	$[kg/m^3]$	0.332
Speed of Sound	a	$[m/s]$	295.070
Pressure	P	$[Pa]$	$2.06 \cdot 10^4$
Temperature	T	$[K]$	216.650

Table 5.2: Air properties at 38000 feet in ISA model.

In order to calculate the AoA_{trim}, one used the values of lift calculated by David Brandão [Brandão, 2015] using the *APAME* software [Filković, 2015]. APAME is a 3D Panel method program, which is able to calculate aerodynamic forces and moments. For subsonic attached flows, this software presents some benefits when compared to the regular CFD programs, being the most important one the computation time. This factor in some applications, as optimizations with multiple disciplines (such as CFD and FEM), can be decisive.

Many values of lift were calculated for different angles of attack for the cruise condition designated above (Table 5.2 and $M = 0.78$), being one able to construct the graph shown in Figure 5.4.

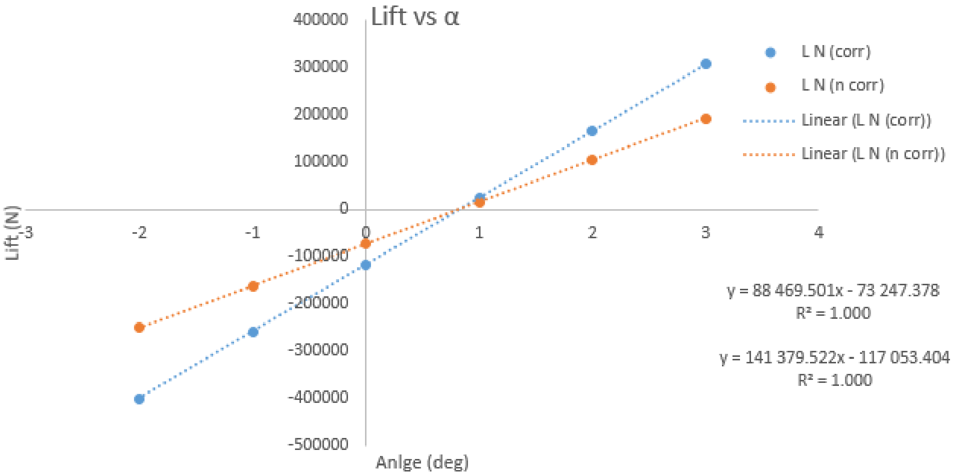


Figure 5.4: Lift vs AoA with and without compressibility correction (APAME).

Having the lift, with and without the compressibility correction, for multiple AoA one did a linear regression in order to have the lift equation vs AoA.

$\mathbf{AoA_{trim}}$		
Without correction	[°]	7.261
With correction	[°]	4.854

Table 5.3: AoA_{trim} obtained using APAME.

Since the $M = 0.78$, the compressibility effects, that can be neglected if $M \leq 0.3$, have to be taken into account. So, the $AoA_{trim} = 4.854$° will be used to calculate the components of the velocity in order to do the CFD analysis in CFX®.

5.3 CFD Analysis

To have the aerodynamics loads that the wing box will be subjected in the given flight condition, one performed a CFD analysis in ANSYS® Workbench using the CFX® module.

5.3.1 Control Volume

One started by defining the control volume that would contain the wing. Using the root chord and wing half span as reference, and having the Y axis in the chordwise direction, the Z axis in the spanwise direction and, finally, the X axis in the vertical direction, the control volume was defined as having $30 \cdot c_r$ m in $-Y$ direction and $60 \cdot c_r$ m in Y direction. In the spanwise direction one assigned that the control volume would have b m. The X direction of the control volume one assigned $30 \cdot c_r$ m. The actual values are summarized in Table 5.4 and the corresponding mesh is shown in Figure 5.5.

$60 \cdot c_r$	$[m]$	342.60
$30 \cdot cr$	$[m]$	171.30
b	$[m]$	36.33

Table 5.4: Control volume dimensions.

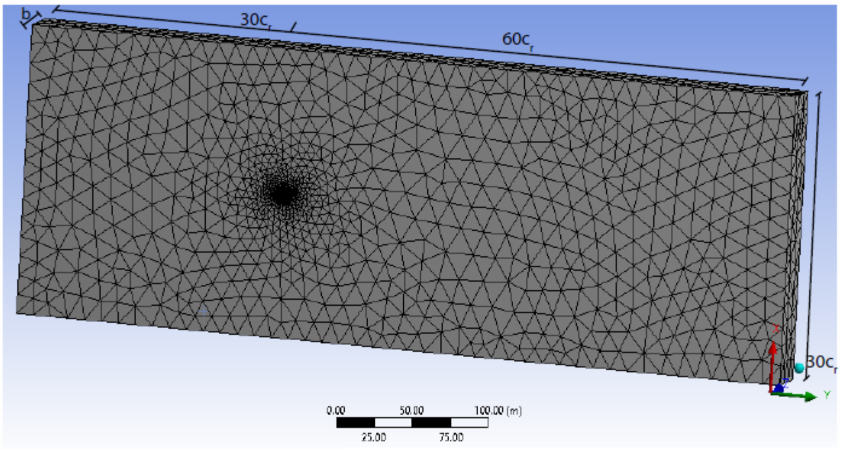


Figure 5.5: CFD meshed domain.

Having the dimension of the control volume defined, the next step in order to implement a CFD analysis is the definition of the Fluid properties and Boundary conditions.

5.3.2 Boundary Conditions

The ANSYS CFX® has a wide range of turbulence models [Brederode, 2014], such as, $k{-}\varepsilon$, SST (Shear Stress Transport), BSL Reynolds Stress and SSG Reynolds Stress or without turbulence, corresponding to the Laminar model. For the CFD analysis one tested with the Laminar, $k - \varepsilon$ and SST.

The fluid used was Air Ideal Gas at $38000\ ft$, having the reference values given in Table 5.2.

The Boundary conditions that one can select in CFX® correspond to *Inlet*, *Outlet*, *Opening*, *Wall* and *Symmetry*. For the *Inlet*, *Sides*, which correspond to the two faces perpendicular to the X axis and the face that is opposite to the root of the wing, and *Outlet* one assigned a *Inlet* boundary condition with the following components of the velocity $(u, v, w) = (19.473, 229.329, 0)$ in the Global coordinate frame. One also assigned a *Zero Gradient* to the turbulence in the boundary. To wall perpendicular to the X axis which intersects the root of the wing one assigned a boundary condition of *Symmetry*. Lastly, to the *Profile* boundary one assigned a *Wall* with no slip boundary condition. One assigned a *Smooth Wall* roughness to the given wall.

5.3.3 Mesh Convergence Test

One performed a convergence test for the aerodynamic mesh of the CFD analysis. Since one wants to find the lift generated by the wing under the given conditions, the lift was used as the variable to evaluate the convergence of the mesh.

CFX® only displays the forces in the direction of the axis from the Global Coordinate frame. In order to have the lift from the X and Y forces, one must calculate the lift using

$$Lift = force_X \cdot \cos(AoA_{trim}) - force_Y \cdot \sin(AoA_{trim}). \tag{5.1}$$

To obtain better results, the size of the elements close to the profile boundary has decreased, being also introduced layers of elements with a constant growth rate. One wants the mesh to be finer closer to the profile, where the the interactions between the profile and fluid are stronger.

Elements $[10^6]$	0.7823	2.5635	7.509	8.711
L $[N]$	246378	285361	276706	272560

Table 5.5: Aerodynamic mesh convergence test.

One can observe that the lift will change by $4146\ N$ when the elements are increased from 7.509 to 8.711 millions, corresponding to a relative change of 1.5%. The fluctuation in lift between the last two meshes is small, so one can conclude that the value of lift was converged, as seen in Table 5.5. The mesh selected to perform the CFD analysis was the one with 8.711 millions elements, resulting in the mesh seen in Figure 5.6.

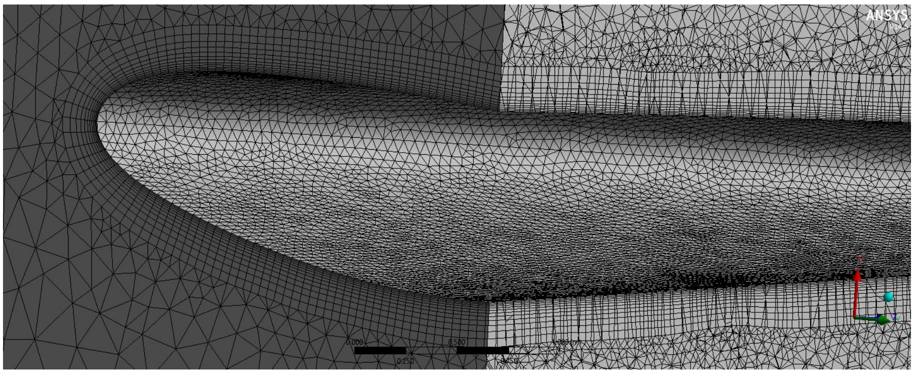


Figure 5.6: CFD mesh refinement detail.

5.3.4 CFD Results

Using the mesh obtained after the convergence test, one calculated the lift, comparing it with the lift computed by APAME. The values of lift obtained using the Shear Stress Transport (SST) turbulence model are displayed in Table 5.6.

	$\mathbf{L}$ $[N]$	$\mathbf{L/L_{req}}$ [-]	$\mathbf{force_X}$ $[N]$	$\mathbf{force_Y}$ $[N]$	$\mathbf{force_Z}$ $[N]$
SST	272560	0.96	273230	-36560	-1362

Table 5.6: CFD analysis results.

The lift obtained during the CFD analysis was calculated for half wing, meaning that the value of lift that one expects to obtain it is around half of the $MTOW$, $\frac{MTOW}{2} = 284490\ N$.

Being APAME a 3D panel method, the type of analysis that it will conduct is an inviscid analysis, in which one will assume that the fluid has no viscosity.

As one can see, the values of the lift obtained is lower than half of the $MTOW$, being the difference equal to 4%. Since APAME is a panel method, were the fluid is considered to be perfect, one expected that the lift obtained would be lower than the one computed with APAME. This can be explained by the presence of real fluid instead of a perfect one, leading to the generation of a boundary layer across the wing. The boundary layer's displacement thickness will change the airfoil's effective camber, decreasing the lift produced by the airfoil operating at a given AoA, decreasing the lift with the increase of the adverse pressure gradients, *i.e.*, with the increase of AoA [Brederode, 2014].

The current CFD analysis was also conducted with other models, such as the $k - \varepsilon$ turbulence model and the Laminar model. For these two turbulence models, the velocity components residues fluctuated around $10^{-2} - 10^{-3}$ across the 100 iterations performed, not showing a convergence for an acceptable value, as seen in Figure 5.7 for the Laminar model.

The resulting pressure distribution, for 8.711 millions elements and using the SST turbulence model, across the wing due to the flow can be seen in Figure 5.8.

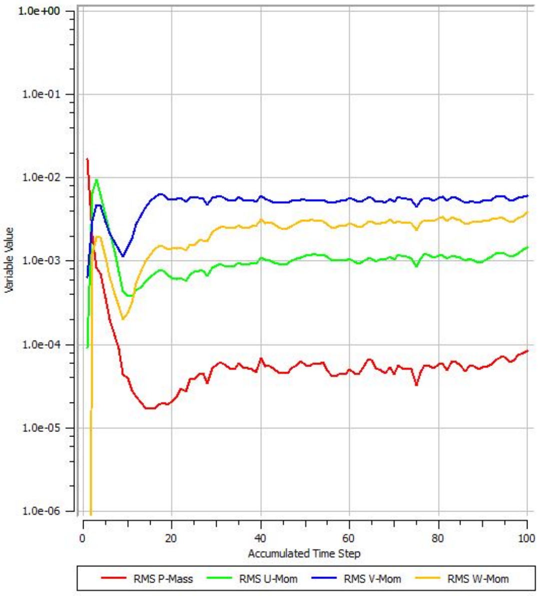
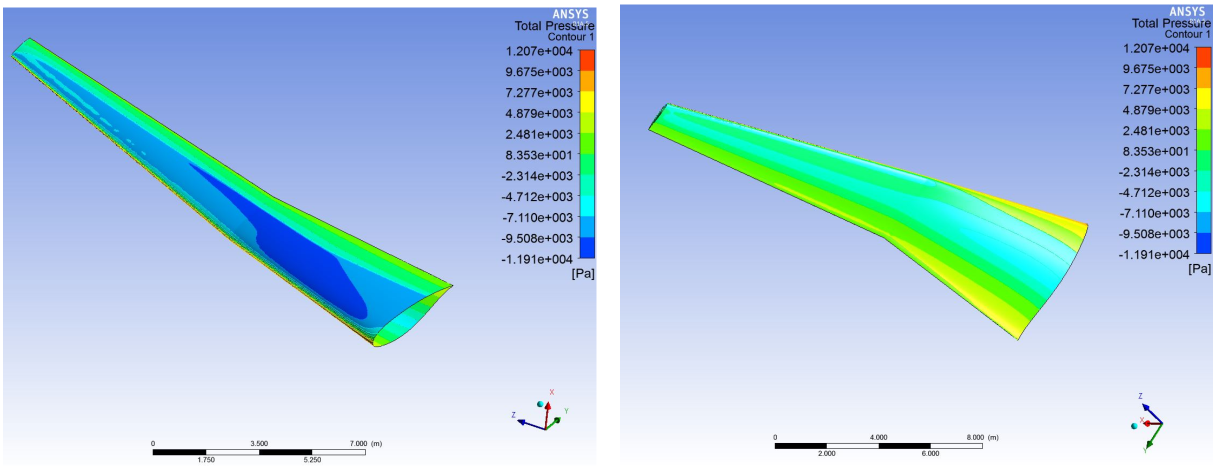


Figure 5.7: Residues value of the CFD analysis using the Laminar model.



(a) Upper wing.

(b) Lower wing.

Figure 5.8: Pressure distribution across the wing using SST turbulence model.

5.4 Structural Model

After having the aerodynamics loads that the wing box will be subjected to, one needs to generate the structural mesh for the wing box in order to perform a FEM analysis.

During the optimization process, multiple FEM analyses will be performed, being the number of FEM analyses equal to the colony dimension times the number of iterations. Therefore, the structural model used during the optimization process should be as simple as possible to avoid a long computation time. Bearing this in mind, one decided to remove all the components of the wing leaving just the block corresponding to the wing box.

The material selected for the wing box was the Aluminium 7065 - T651 [Metals, 2015]. This aluminium is used for many applications in the aerospace field, such as, aircraft fittings, gears, fuselage and missile parts. Its mechanical properties are summarized in Table 5.7.

D	$[kg/m^3]$	2810
E	$[GPa]$	71.7
ν	$[-]$	0.33
Tensile Yield Strength	$[MPa]$	503
Compressive Yield Strength	$[MPa]$	503
Tensile Ultimate Strength	$[MPa]$	572

Table 5.7: Aluminium 7075-T651 mechanical properties. [Metals, 2015]

5.4.1 Wing Box Cross-section Discretization

In order to have a simpler connectivity between elements of the wing box cross section, one used a mapped surface mesh in the wing box. Since one wants to optimize the cross section of the wing box, the discretization of the cross section will correspond to the most important part during the mesh generation. The higher number of elements in the wing box cross section will also mean a higher number of possible combinations of designs obtained during the optimization process.

For the wing box cross-section one defined that the number of division of the vertical edges would be equal to 18, and the number of divisions of the horizontal edges would be equal to 64, resulting in 1152 elements in the wing box cross-section. These elements are repeated across the span of the wing. The wing box cross-section discretized can be seen in Figure 5.9.

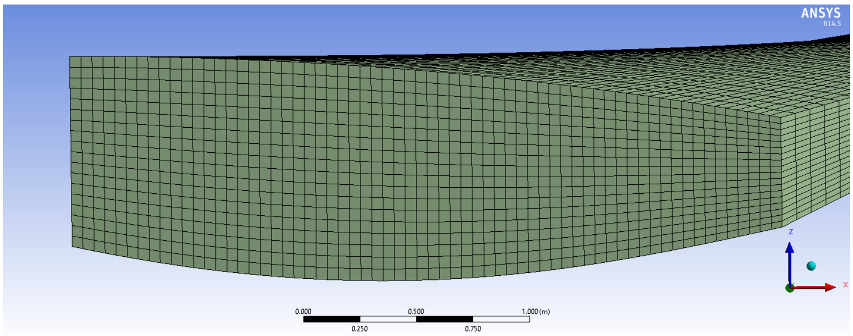


Figure 5.9: Wing box cross-section discretized.

5.4.2 Mesh convergence test

Having defined the elements that one wants to have in the wing box cross-section, one must test the mesh to find convergence. One wants to find the least number of division on the longitudinal (spanwise) edges to have convergence. The minimum number of division will be given by doing a convergence test, and finding the number of elements for which it has reached convergence. Since one wants calculate the strain energy of each element, and being the strain energy a function of the displacement vector, one used the Maximum Displacement as the convergence variable being analysed. The results are summarized in Table 5.8.

Elements	29952	44928	59904
Max. Displacement $[mm]$	5.291	5.323	5.171

Table 5.8: Structural mesh convergence test.

The maximum displacement of the wing for the given aerodynamic loading displayed small fluctuation with the increase of the number of elements of the structural mesh. The number of elements of the cross-section were kept constant throughout the convergence test of the structural mesh, being the number of elements increased by increasing the number of divisions of the longitudinal edges.

The element used to conduct the FEM analyses was the Solid186, a higher order 3-D 20-node solid element exhibiting a higher quadratic displacement behaviour, each node has three degrees of freedom [ANS, 2013].

The maximum displacement occurred on the tip of the wing as expected, corresponding to the same node for the three FEM analyses. Increasing the number of elements from 29952 to 44928 led to a change of $3.2 \cdot 10^{-2}\ mm$ in the maximum displacement, corresponding to a relative change of 0.60%. Using a finer mesh (59904 elements) the maximum displacement obtained was equal to $5.171\ mm$. When compared with the 44928 elements mesh, the maximum displacement changed by $0.152\ mm$, corresponding to a relative change of 2.84%. When compared the finest mesh with the coarsest one, the maximum displacement had a variation of $0.120\ mm$, corresponding to a relative change of 2.26%.

Increasing the number of elements in order to obtain a finer mesh resulted in small changes in the maximum displacement value, therefore, one can say that the value of the maximum displacement converged. As stated before, one wants to spend the least time in the FEM analyses so the mesh chosen to conduct the FEM analyses during the optimization process was the one with 29952 elements, corresponding to 26 divisions in the spanwise direction. The final mesh can be seen in Figure 5.10.

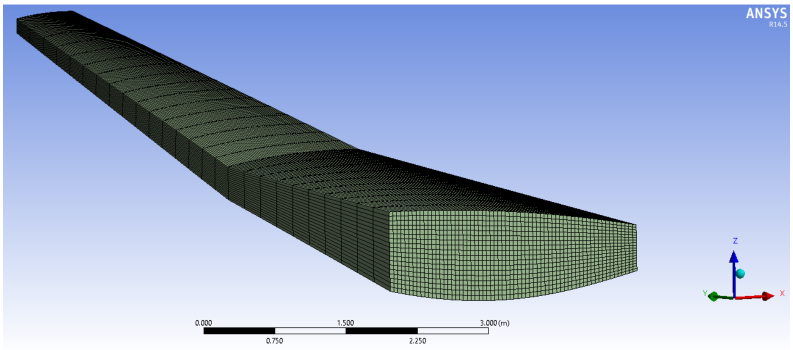


Figure 5.10: Final structural mesh of the wing box, used in the structural topology optimization.

The pressure distribution interpolated for the given mesh can be seen in Figure 5.11.

The deformed shape of the wing box subjected to the pressure distribution described in Figure 5.11 and having the nodes fixed at the root can be seen in Figure 5.12. The maximum displacement is equal to $5.921\ mm$, as was obtained during the convergence test, occurring at the tip of the wing. The displacements that the wing box will suffer are really small when compared with the displacements of the regular aircraft wings which can reach the value of few meters. For a usual high aspect ratio wing

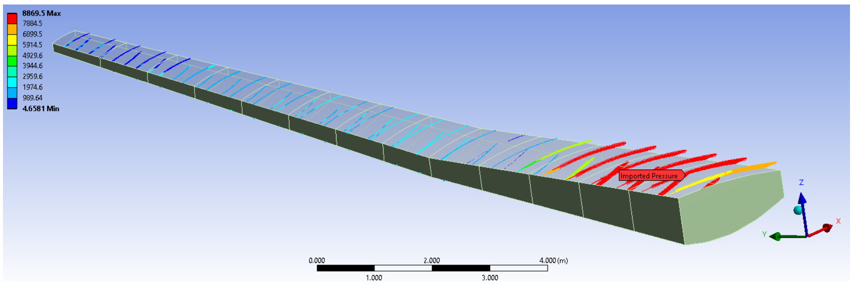


Figure 5.11: Interpolated pressure distribution for the wing box structural mesh.

one expects a big deflection, although, one is doing the given analysis with the wing box domain full, so the expected deflection of the wing box is expected to be small. A regular wing box is composed usually by two spars and a reinforcement close to the skin, resulting in a much smaller volume fraction when compared with the full wing box.

The total strain energy of the full wing box, using the given mesh and pressure distribution, is equal to $\mathbf{8.214}\ J$.

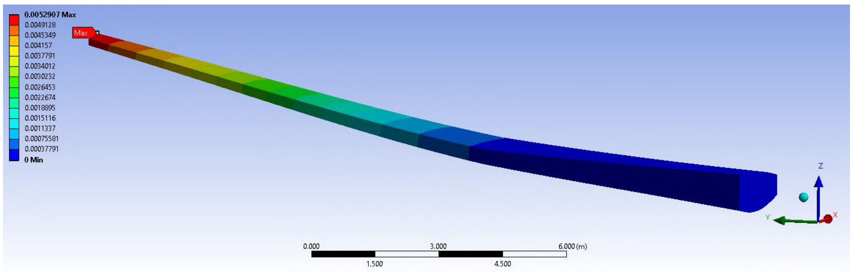


Figure 5.12: Deformed shape of the wing box.

Chapter 6

Wing Box Cross-section Optimization

One will start by the optimization of the wing box cross section. This problem will also correspond to a 2D optimization being the analysis of structure's performance conducted in 3D.

In order to optimize the wing box cross-section of the high AR wing, some modifications to the code were required. Firstly, for the optimizer the problem solved will be two-dimensional, meaning that the optimizer will remain almost unaltered, requiring only the modification of the boundary conditions that the solutions need to satisfy. Secondly, the structural problem solved is three-dimensional, having in mind that the optimization problem is 2D, one will need to relate the 3D structural elements with the 2D elements which represent the wing box cross-section. Lastly, the finite elements analysis will be conducted using the structural model developed in section 5.4.

6.1 Boundary Conditions

Looking at the literature case solved (section 4.5), the problem displayed three boundary conditions, two points with no displacement and the point load. However, the wing box cross-section optimization will be a more complex problem.

In the given model, one has a pressure distribution across the upper and lower surfaces of the wing box. The pressure distribution will demand that all the elements of the upper and lower surfaces must be present in the solutions generated, due to this pressure being applied in the elements which cover these surfaces. Therefore, the solutions generated must contain these elements in order to be possible to perform a FEM analysis of the structure. The connectivity between the upper and lower surfaces must also be ensured, being this assured if the elements from the upper and lower surface are present in the solution.

6.2 3D Elements vs 2D Elements

The goal of the optimization will be the same as the one for the literature case solved (chapter 4), the minimization of the wing box's strain energy. Having the same goal, both optimization problems will also share the same objective function (equation (4.1)), meaning that the pheromone update will be conducted in a similar way as for the literature case.

During the solution generation, the ant's path (2D) will define the shape of the cross-section. Each 2D element chosen by the ant will correspond to a set of 3D structural elements. This set to the 3D elements correspond to the ones which are arranged longitudinally behind the one selected by the ant, allowing to have a constant cross-section along the wing's span. If a element does not belong to the ant's path a row of 3D elements will disappear.

The pheromone update will be conducted using the strain energy as explained in chapter 4 using equation (4.7). The value of the $\Delta\tau_i$, increase in pheromone intensity for element i, will depend of the strain energy of the element i. A relation between the strain energy of the 3D elements and the 2D one must be assigned in order to perform the pheromone update. Since a 2D element represents a column of 3D elements of the wing box, the presence or not of every element of these column must be taken into account.

The "strain energy" ($u_{i_{2D}}$) of the 2D elements will be equal to the sum of the strain energy ($u_{ij_{3D}}$) of the set of 3D elements which are arranged longitudinally. The relation is given by,

$$u_{i_{2D}} = \sum_{j=1}^{N_{long}} u_{ij_{3D}}, \tag{6.1}$$

where N_{long} is the number of longitudinal divisions of the wing box.

6.3 Results

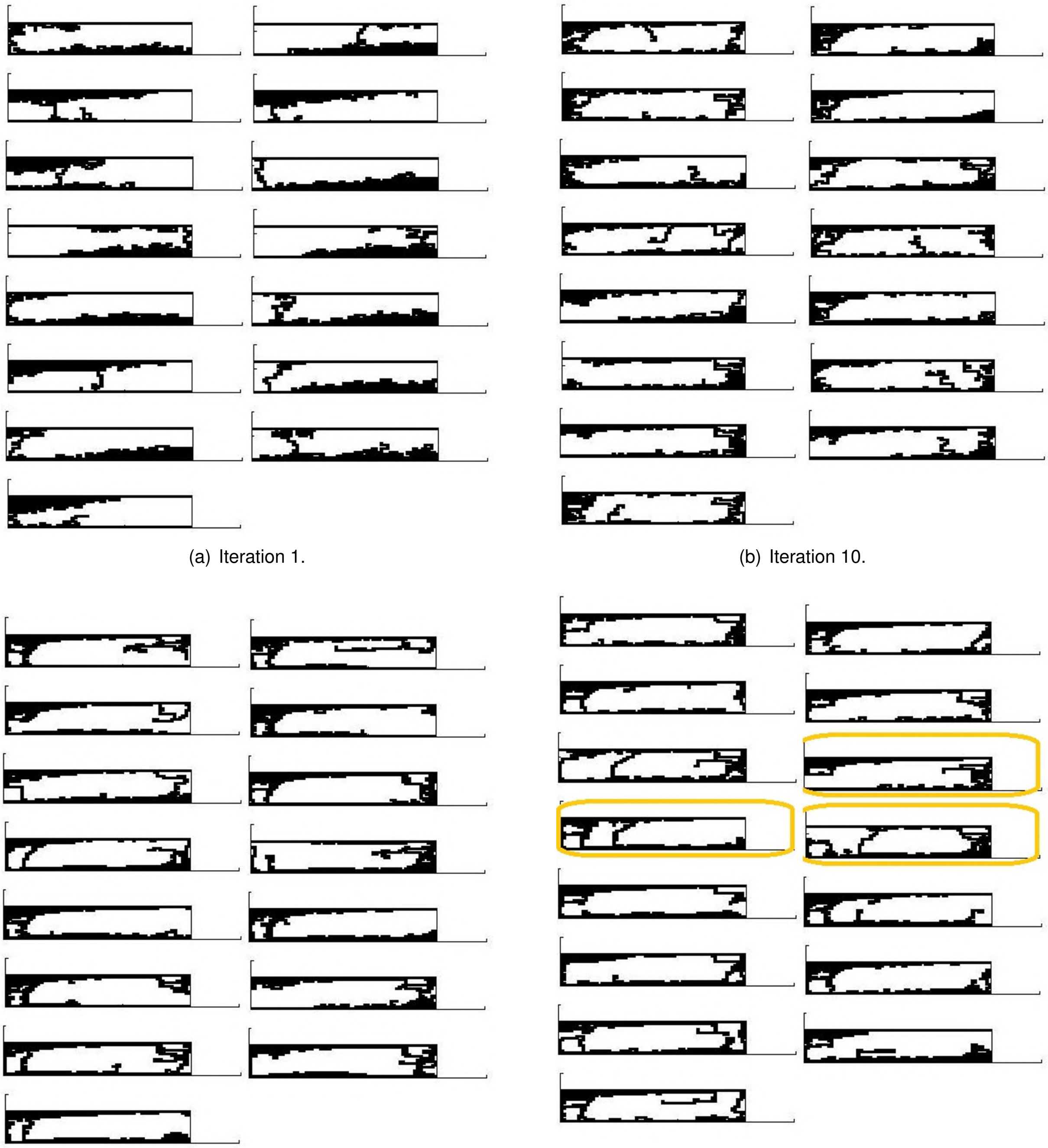
The optimization of the wing box cross-section was performed using the value of the parameters with which one obtained the best solution for the literature case, given in Table 6.1. The CPU used to perform the optimization was the Intel(R) Core(TM) i5-3350P CPU @ 3.10Ghz.

For the optimization problem one will have `nelx = 64`, corresponding to the number of division in the chordwise direction, and `nely = 18`, equal to the number of divisions on the z direction, as seen in Figure 5.10.

`colony_dim`	15
`n_ite`	30
`evap` (ρ)	0.3
`alpha_p` (α)	1
`lambda` (λ)	2.2
r_{min}	1.2

Table 6.1: ACO parameters used to obtain the best solution.

One decided to conduct the first optimization of the cross-section with a smaller volume fraction than the one used in the literature case, being `vol_f = 0.35`. This way, the structure obtained will display roughly 35% of the weight of the full wing box.



(a) Iteration 1.

(b) Iteration 10.

(c) Iteration 20.

(d) Iteration 30.

Figure 6.1: Wing box cross-section for iteration number 1, 10, 20 and 30 for `vol_f = 0.35`.

Iteration	26	27	28	29	30
Strain energy $[J]$	14.329	15.025	15.177	14.798	14.783

Table 6.2: Minimum strain energy obtained for the last 4 iterations for `vol_f = 0.35`.

The evolution of the wing box cross-section throughout the optimization process can be seen in Figure 6.1. For the first iteration (Figure 6.1(a)), the distribution of elements is mainly random. There is the generation of a region with high density of active elements but, on the other hand, other regions almost do not have any active elements. For all the solutions, the lower and upper part of the structure

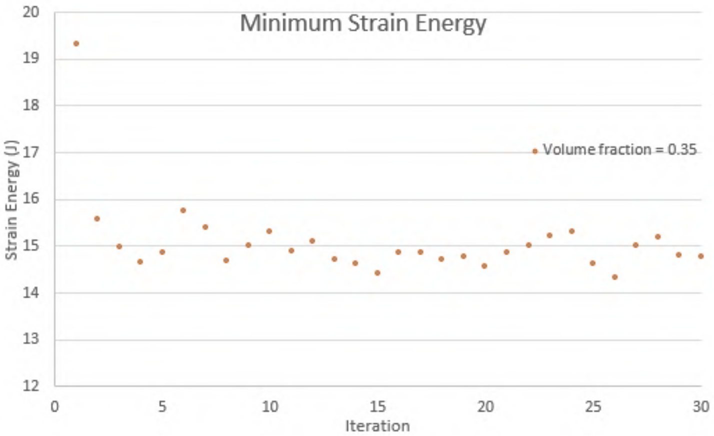


Figure 6.2: Minimum strain energy throughout the cross-section optimization for `vol_f = 0.35`.

usually is connected by a single member, explaining the high strain energy of the structure. The solutions found by the colony also due not resemble each other.

For the other iterations, one can notice that the ant's path will select mainly the elements which are close to the bottom right and upper left corner. In the middle part of the wing box cross-section few elements are present, generating in few cases a member which connects the bottom skin with the upper one. These designs have a performance not as good as the ones which do not display this member.

As seen in Figure 6.1(d), the solutions found at the end of the optimization process, a major part of the solutions display a connection between the upper and lower skin in both sides of the wing box, the ones which do not display this feature have a higher strain energy than the ones which display, such as the solutions from ant 6, 7 or 8 (highlighted in Figure 6.1(d)) with a strain energy equal to $34.457\ J$, $16.401\ J$ and $19.863\ J$, respectively.

One can also notice that the upper left and lower right corners of the wing box display a higher concentration of elements. The presence of the elements in these regions will increase the stiffness of the structure, when subjected to torsion. Being the angle of twist inversely proportional to the torsional constant of a section, the torsional constant of a cross-section will increase if one has a higher concentration of material close to the boundaries of the cross-section. So the higher concentration of material in the corners can be explained as a form of making the structure stiffer when subjected to torsion.

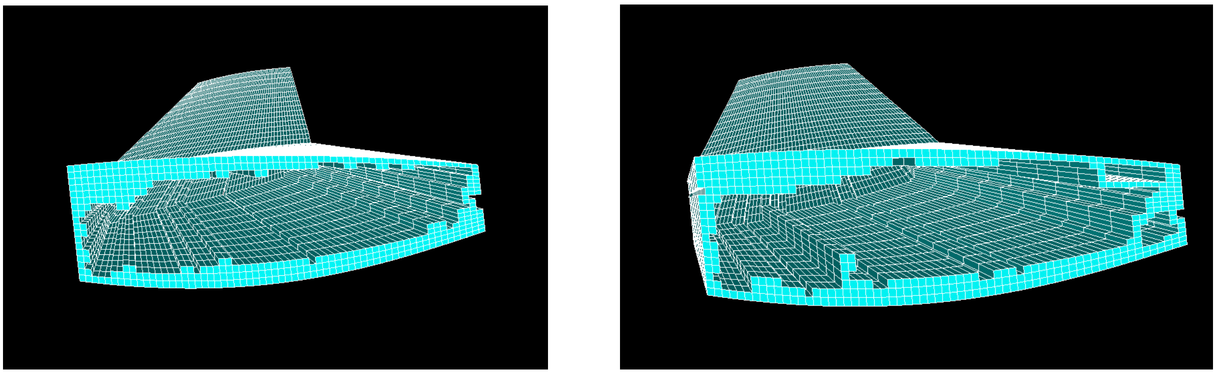
The lower and upper skin have also been reinforced in many of the solutions found, displaying usually two layers of elements close to the skin.

As seen in Figure 6.2, the minimum strain energy after the first iterations displays a fluctuation around the value of $15\ J$, being the minimum strain energy equal to $14.783\ J$ for the last iteration. The solution which displayed the minimum strain energy across the iterations displayed a strain energy equal to $14.329\ J$, being obtained at iteration number 26.

In Table 6.2, one has the minimum strain energy for the last 5 iterations. As it can be seen, the maximum change in module of the strain energy will be equal to $0.696\ J$ between iteration 26 and 27, corresponding to a relative change of 4.86 %. These small fluctuations can be explained with the stochastic behaviour of the algorithm and the small volume fraction used. During the solution generation, the ants will be placed in a random element of the domain in order to start the path's construction. Having a small volume fraction constrain, if the first element assigned does not belong to the optimum solution and is placed in a region which is not usual to have active elements the solution, some elements might be 'wasted' when trying to find the optimal solution, leading to the generation of a cross-section with a higher strain energy.

Figure 6.3 shows the wing box cross-section with the minimum strain energy for iteration 26 (Figure 6.3(a)) and iteration 30 (Figure 6.3(b)). The strain energy for the solution found in iteration 26 is equal to $14.329\ J$ and $14.783\ J$ for iteration 30.

When compared, both solutions display some resemblances, such as the high concentration of elements on the upper left corner and a layer of elements close to the bottom skin of the wing box.



(a) Iteration 26.

(b) Iteration 30.

Figure 6.3: Best cross-sections found for iteration 26 and 30 for `vol_f = 0.35`.

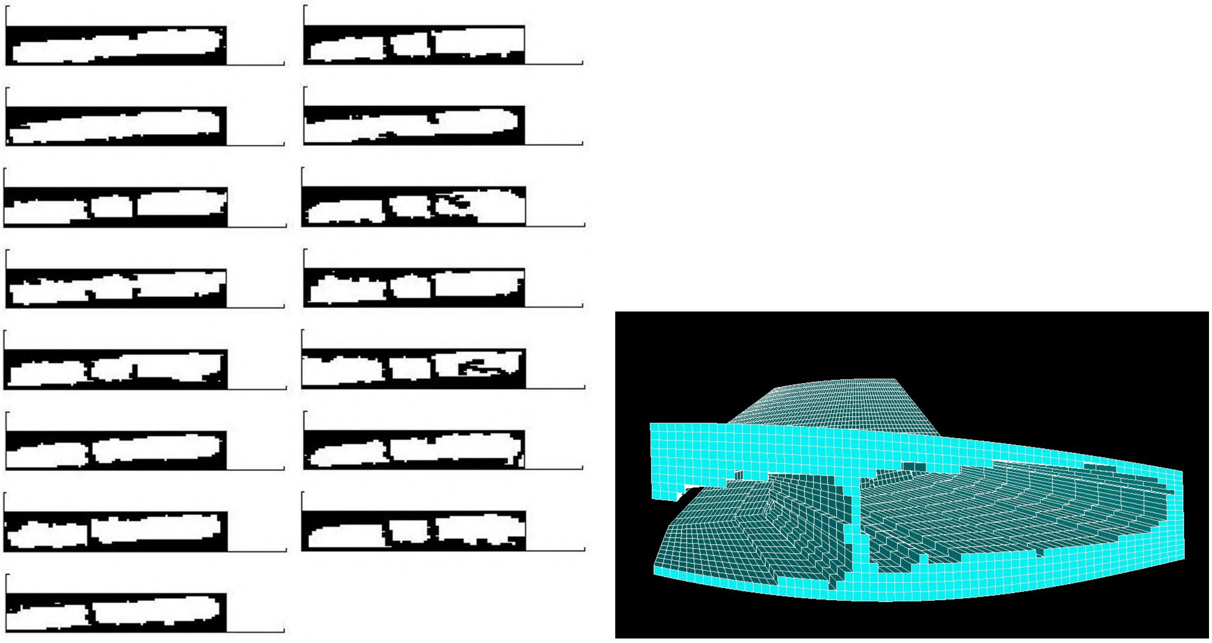
For iteration 26, close to the upper skin one can notice some separate rows of elements, resembling to the skin stiffeners to which the skin is bolted too. The best solution for iteration 30 as shown in Figure 6.4(a), when compared with the one for iteration 26, is formed a hollow member in the lower and upper right corners, which the best solution for iteration 26 is missing. A higher concentration of elements can also be seen close to this region for iteration 30, leading to a lower concentration of elements close to left upper corner as can be seen in iteration 26.

Afterwards, the volume fraction (`vol_f`) was increased to 0.45 in order to observe the changes in topology of the wing box cross-section, all other parameters were kept constant, being its value in Table 6.1.

As expected, the solutions obtained at iteration 30 displayed a lower strain energy as the one obtained for a volume fraction equal to 0.35, being equal to $11.617\ J$. When comparing the results for both values of volume fraction, there is a relative decrease of 21.42 % when the volume fraction is increased from 0.35 to 0.45.

The best solution obtained for `vol_f = 0.45`, when compared with the best solutions obtained for `vol_f = 0.35`, one can notice the presence of a member that connects the bottom and upper skin at one third of the chordwise length, being also connected on the right side of the wing box.

Besides the presence of this member which is replacing the connection between the upper and bottom skin on the left side, the elements distribution across the wing box resembles the one for `vol_f=0.35`,



(a) Solutions.

(b) Best solution.

Figure 6.4: Wing box cross sections for iteration 30.

just having more elements in the same regions where the elements are distributed.

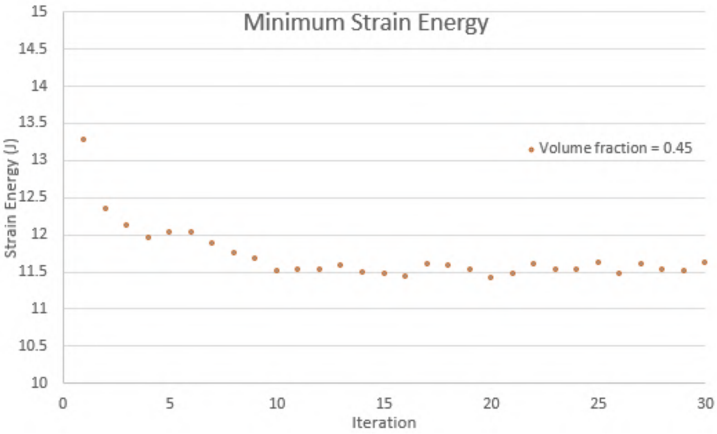


Figure 6.5: Minimum strain energy throughout the cross-section optimization for `vol_f` = 0.45.

With the increase of the volume fraction, one can notice also that the fluctuations of the minimum strain energy throughout the optimization process will have a lower magnitude, close to $0.1\ J$ between iterations, as illustrated in Figure 6.5. This fact can be explained with the higher number of elements that the ants can select, meaning that the the absence of some elements from the solution will have a lower impact in the overall performance of the cross-section obtained.

To complete the optimization processes it was required 7 hours 11 minutes and 36 seconds for a volume fraction equal to 0.35 and 8 hours 11 minutes and 47 seconds for a volume fraction equal to 0.45, corresponding to 450 FEM analyses for each run, 15 per iteration for 30 iterations.

Chapter 7

Wing Box Optimization

Having performed the optimization of the wing box cross-section using ACO, one extended the implementation in order to obtain the optimal 3D distribution of material for the wing box.

When compared with the cross section optimization performed in chapter 6, the given optimization will have more freedom than if the topology optimization is restricted to the cross-section design.

7.1 3D Implementation

The structural model used to perform the FEM analyses will be the one developed in section 5.4. The given structural model is composed of 29952 elements, corresponding to 1152 elements in the wing box cross-section, 18 in the y direction and 64 in the x direction (chordwise direction), and 26 elements in the z direction (spanwise direction). If one used all the elements of the structural model as design variables, the computation time could increase drastically, so one decided to group elements, forming blocks with the structural elements using these blocks as design variables.

7.1.1 Optimization Blocks

In the first instance, each block will be formed by 4 elements in the z and x direction and 1 element in the y direction, the given referential corresponds to the one used during the structural model generation (Figure 5.10), forming a 4 by 4 by 1 block composed of 16 elements of the structural mesh, since one is using the same structural model as the one used in the wing box cross-section optimization. For the cross-section optimization one gives emphasis to the cross-section discretization, meaning that the discretization of the cross-section will be finer than the mesh required for the FEM analysis. As explained before, one wanted to have as much freedom as possible for the optimization having at the same time the minimum computation time possible. The resulting structural mesh had a finer discretization of the cross-section than in the longitudinal direction, therefore the blocks will be constituted only by one element in the y direction.

For the last run, each block will be formed by 2 elements in the z and x direction and 1 element in the y direction, resulting in a 2 by 2 by 1 block composed of 4 elements of the structural mesh.

For the optimization problem one will have two domains: one where the optimization is performed, corresponding to the elements that the ants can select to form their path; the other one will correspond to ones used to solve the FEM analysis, the elements of the structural mesh. In order to differentiate both elements, the elements of the structural mesh will be referred as the structural elements (st). One kept the direction of the axis used during the literature case optimization (chapter 4), x direction as the chordwise direction, y being the third direction the z equal to the chordwise direction.

7.1.2 Boundary Conditions

The elements that compose the upper and lower surface of the wing box, in which the pressure distribution will be applied, must be present. One decided that these elements will not be a part of the optimization process, being always present from the start in the solutions generation. When compared with the cross-section optimization problem, where the solution generated needed to have the elements that correspond to the skin, in the 3D optimization this condition is satisfied from the beginning. Nevertheless, the connectivity between the upper and lower surface must be ensured.

For the optimizations with the blocks composed of 16 structural elements one will demand that the solution generated must have at least 4 blocks that are connected with the upper skin and the same amount for the lower skin of the wing box. For the blocks composed of 4 structural elements the number blocks that must be connected with the surfaces will increase to at least to 8, which will correspond to 4 blocks of 16 structural elements. If an ant's path has the given number of blocks present close to the upper and lower surface and the volume fraction specified has been reached the ant will be stopped.

7.1.3 Connectivity

The connectivity between blocks during the optimization process will be identical to the one used during the 2D optimizations. Instead of demanding that the elements must share at least one edge, the corresponding of the edge in 3D will be the surface of the element. This demand will result in a 6-elements neighbourhood during the element selection for the ant's path.

7.1.4 Optimization Domain

The resulting elements for each direction are given in Table 7.1.

	st	$4 \cdot 4 \cdot 1$	$2 \cdot 2 \cdot 1$
`nelx`	64	16	32
`nely`	18	4	8
`nelz`	26	26	26
`nelem`	29952	1664	6656

Table 7.1: Number of elements for each direction and different set of blocks.

As it can be seen, the number of structural members will correspond to the ones obtained in chapter 5. For the $4 \cdot 4 \cdot 1$ blocks (composed of 16 structural elements), the number of elements in the y direction (nely) will be equal to 4, because one has removed two columns of elements, which will correspond to the upper and lower skins of the wing box, renaming 16 elements in this direction, which divided by 4 will give the `nely = 4` for the $4 \cdot 4 \cdot 1$ blocks, happening the same for the $2 \cdot 2 \cdot 1$ blocks.

The resulting optimization mesh for the $4 \cdot 4 \cdot 1$ blocks is shown in Figure 7.1, where the cluster of structural elements can be seen in grey or white being the skin elements represented in green.

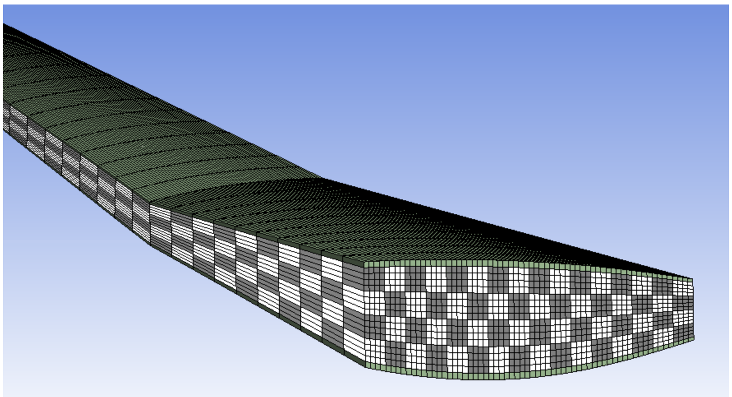


Figure 7.1: Structural mesh vs optimization mesh for blocks $4 \cdot 4 \cdot 1$.

The CPU used to perform the optimization was the Intel(R) Core(TM) i5-3350P CPU @ 3.10Ghz.

7.2 Results

For the given problem, 3 runs were performed, two with 1664 and one with 6656 optimization blocks, referred as coarse and fine mesh, respectively. The two runs with the Coarse mesh were performed with a volume fraction `vol_f = 0.35` and `vol_f = 0.45`, in order to compare with the results obtained for the cross-section optimization. The run with the fine mesh was performed with `vol_f = 0.35`.

The value of the ACO parameters used during the given runs will be equal to the ones used in the cross-section optimization, shown in Table 6.2.

7.2.1 Coarse Mesh

The structures obtained for `vol_f = 0.35` and `vol_f = 0.45` at the end of the optimization process can be seen in Figures 7.3 and 7.4, respectively. Figures 7.3(a), 7.3(b), 7.3(c) and 7.3(d) corresponds to a representation of the wing box without the layer of elements which form the skin for `vol_f = 0.35`. The same can be seen in Figures 7.4(a), 7.4(b), 7.4(c) and 7.4(d) for `vol_f = 0.45`.

For `vol_f = 0.35`, the structure obtained displays most of the elements close to the upper and lower surface of the wing box, being covered mainly by a layer of blocks.

Along the spanwise direction, one can notice connections between the upper and lower surface, the biggest member that will make the connection is placed after the wing's break. Other connections can be seen in the solution, but these members are usually composed of one or two blocks. The wing box close to the root is mainly hollow, being filled the inside close to the wing break.

A higher concentration of elements can be seen close to the wing root, the elements concentration decreases in the spanwise direction. Close to the tip few elements are present, being connected to the rest of the structure by one member. As it can be seen in the pressure distribution across the surface of the wing box (Figure 5.8), in the region close to the tip the magnitude of the pressure will be lower than the one close to the root, meaning that fewer elements will be necessary in that region to support the loads.

For `vol_f = 0.45`, the structure obtained displays many traits that the one obtained for `vol_f = 0.35`. Some differences can be seen in both structures can be seen close to the root, where for the latest case the structure is not hollow. Being the wing box close to the root filled with blocks.

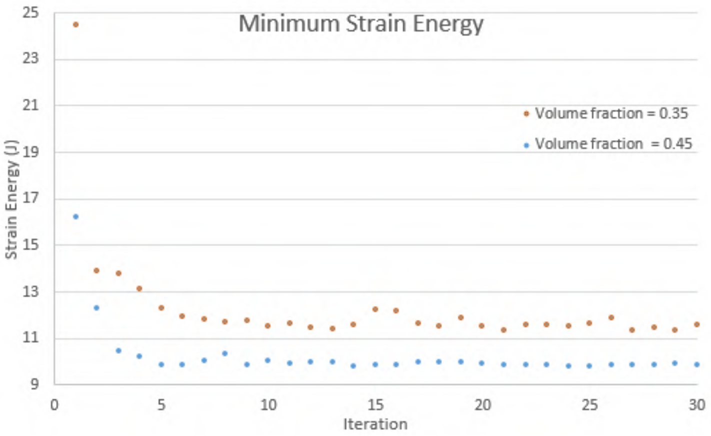


Figure 7.2: Minimum strain energy throughout the wing box 3D optimization for `vol_f` = 0.35 and 0.45.

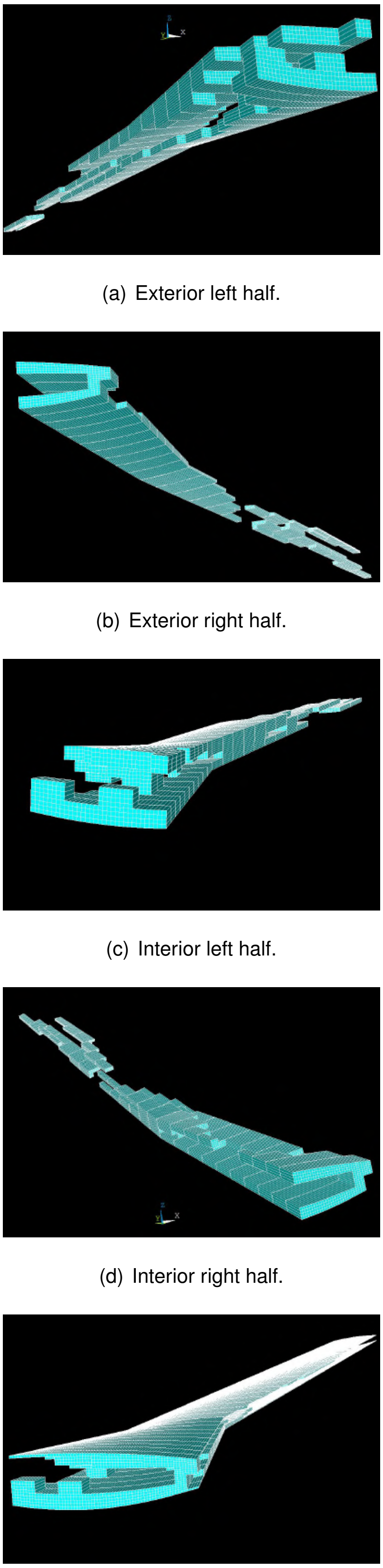
The connection between the upper and lower surface can also be seen in the same regions for both cases, although, for `vol_f = 0.45` the formed members are composed of more blocks than the ones for the solution with a lower volume fraction.

For `vol_f = 0.35` the minimum strain energy obtained at iteration 30 was equal to $11.583\ J$, being equal to $9.830\ J$ for `vol_f = 0.45`. The strain energy obtained for `vol_f = 0.45` is smaller than the one for `vol_f = 0.35`, as was expected.

The minimum strain energy after the first iterations will remain almost constant, as shown in Figure 7.2, displaying small fluctuations for `vol_f = 0.35` with a magnitude close to the $0.6\ J$, which for iterations 26 and 27 corresponds to a relative change of 4.85%.

For `vol_f = 0.45` the fluctuation when compared with the above case are considerately small, being around $0.02\ J$.

To complete the optimization processes it was required 71 hours 30 minutes and 39 seconds for a volume fraction equal to 0.35 and 76 hours 57 minutes and 50 seconds for a volume fraction equal to 0.45, corresponding to 450 FEM analyses for each run, 15 per iteration for 30 iterations.



(a) Exterior left half.

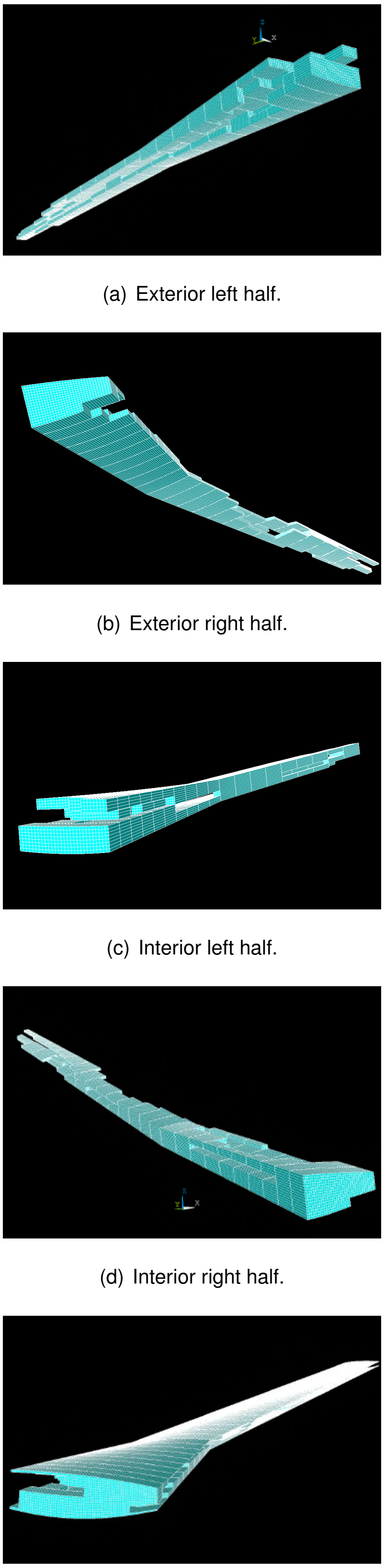
(b) Exterior right half.

(c) Interior left half.

(d) Interior right half.

(e) Wing box with skin.

Figure 7.3: Wing box solution for `vol_f = 0.35`.



(a) Exterior left half.

(b) Exterior right half.

(c) Interior left half.

(d) Interior right half.

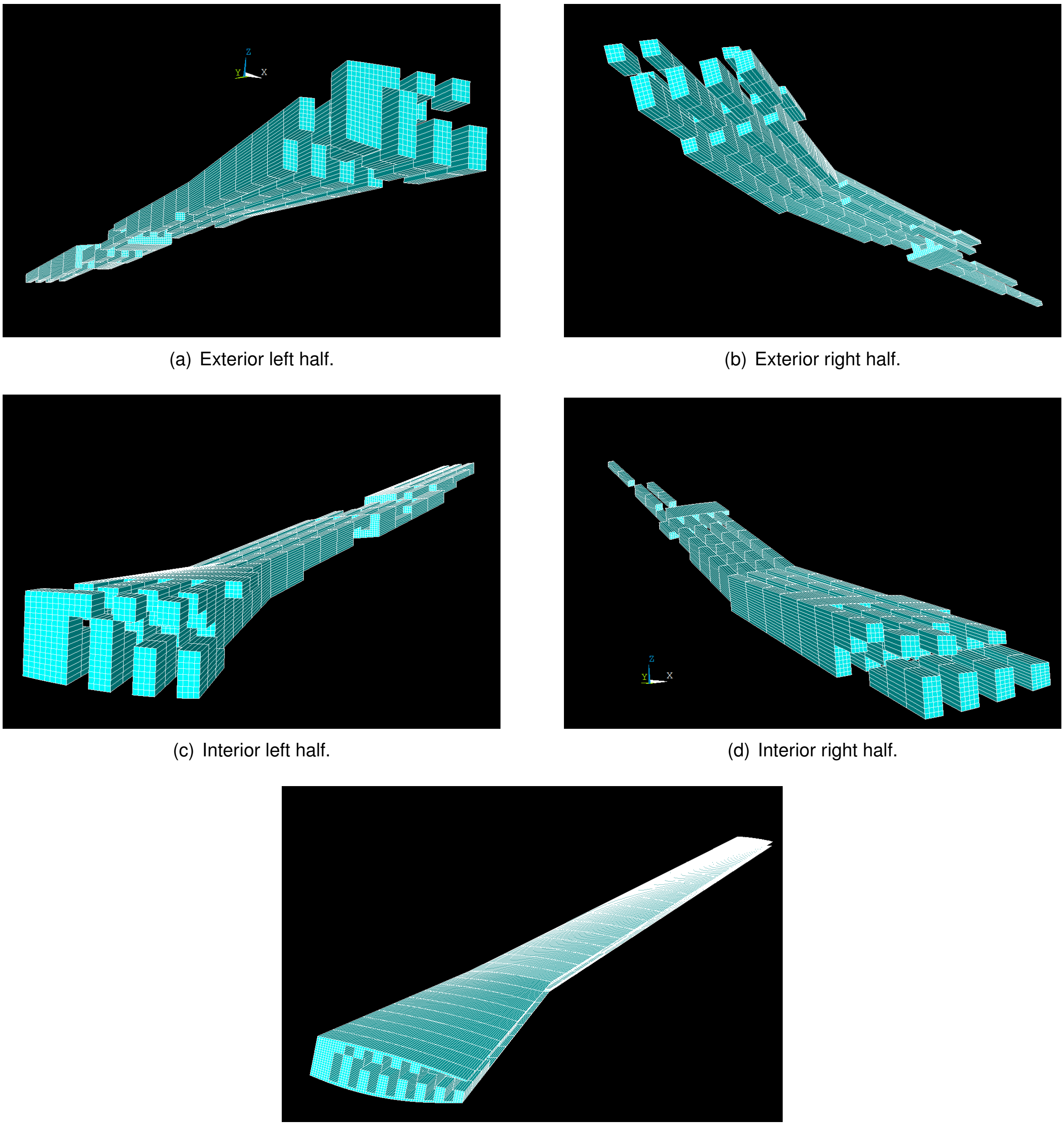
(e) Wing box with skin.

Figure 7.4: Wing box solution for `vol_f = 0.45`.

7.2.2 Fine Mesh

Using $2 \cdot 2 \cdot 1$ blocks, one performed the given 3D optimization. The best solution at the end of the optimization process is represented in Figure 7.5. The strain energy of the given structure is equal to $11.129\ J$.

The solution obtained when compared with the one obtained for the $4 \cdot 4 \cdot 1$ blocks displays a different topology. One can notice blocks aligned along the spanwise direction, which will connect to each other at the middle of the wing. These blocks can be seen close to the upper and lower surfaces of the wing box, being formed usually by two blocks side by side.



(a) Exterior left half.

(b) Exterior right half.

(c) Interior left half.

(d) Interior right half.

(e) Wing box with skin.

Figure 7.5: Wing box solution for `vol_f = 0.35`.

The root cross-section of the solution obtained when compared with the 1664 blocks for a volume fraction equal to 0.35, the presence of the blocks line instead of a distribution of blocks around the boundary of the wing box for a hollow structure close to the root.

A higher concentration of blocks can be seen in the first half of the wing box (root to break) as was visible for the 1664 blocks solution, displaying fewer blocks close to the tip of the wing box.

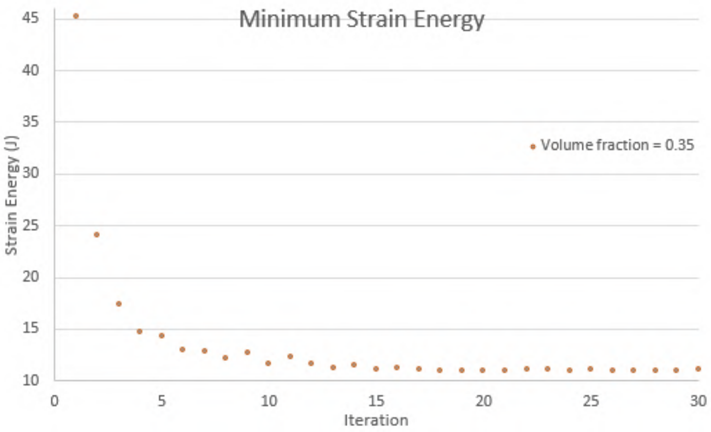


Figure 7.6: Minimum strain energy throughout the wing box 3D optimization for `vol_f` = 0.35.

The evolution of the minimum strain energy throughout the iterations of the optimization process can be seen in Figure 7.6. The minimum strain energy value has converged, displaying small fluctuation close to $11\ J$ with a magnitude around $0.05\ J$.

With the increase of blocks during the 3D optimization, one can notice a slight decrease in strain energy from $11.583\ J$ for the coarse mesh to $11.129\ J$ for the fine mesh. This decrease can be explained with the increase of freedom that the algorithm will have due to the increase of blocks available to select during the solution generation. Nonetheless, one should note that the given algorithm is meta-heuristic relying in a stochastic process to select the elements using at the same time some rules, meaning that for different runs with the same parameters, one can have different results.

To complete the optimization process it was required 102 hours 49 minutes and 54 seconds, corresponding to 450 FEM analyses, 15 per iteration for 30 iterations.

7.3 2D vs 3D Results

In Table 7.2, the values of strain energy for the cross-section and 3D optimization are shown, for the 3D optimization with `vol_f = 0.35` the table entry displays the values for the coarse mesh and fine mesh, respectively. When the strain energy for the best solutions found in the cross-section and the 3D optimization is compared, one can notice that for both values of volume fraction one has a lower strain energy value for the 3D optimization. Although, for both cases the volume fraction is the same, one will have more structural elements for the 3D optimization, because the elements that form the upper and lower skin of the wing box do not belong to the optimization problem, being always present. So for the same volume fraction, one always has more elements for the 3D optimization case. Nevertheless, the difference in strain energy is quite large and part of that difference may be due to the higher freedom in selecting the elements for the solution.

`vol_f`	0.35	0.45
Cross-section $[J]$	14.783	11.617
3D $[J]$	$11.583/11.129$	9.830
CPU time $[s]$	$257439/370194$	277070

Table 7.2: Minimum strain energy for the cross-section and 3D optimization.

From the cross-section optimization for the 2D optimization, one can also notice a drastic increase in the running time. This fact can be explained with the numbering of the elements. The structural elements are numbered in a crescent way in the spanwise direction, so one is performing the cross section optimization the elements which are removed from the matrices that the ANSYS® needs to solve will be sequential, taking less time to re-arrange the matrices. For the 3D optimization, the re-arrange of the matrices takes more time because the elements that need to be removed from the matrices are more dispersed.

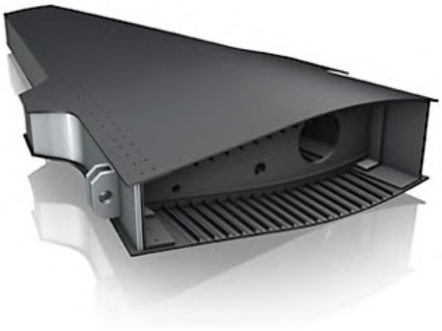


Figure 7.7: Conventional wing box.

Observing Figure 7.7, which represents a conventional layout of a wing box, one can notice that the solutions obtained for the cross-section optimization have some traits in common with the conventional one, the presence of two structural elements which correspond to the spars and the reinforcement close to the skin. From the conventional wing box one can notice the absence of a structure similar to the ribs. The structure obtained after performing the 3D optimization has fewer similarities with the conventional wing box than the ones obtained during the cross-section optimization.

Chapter 8

Conclusions

8.1 Achievements

The main objective of this work was to solve a topology optimization problem using the Ant Colony Optimization algorithm, being the optimization problem the minimization of the strain energy of a high aspect ratio wing box.

After doing a review of the ACO algorithm applied to a NP-hard combinatorial problem, one performed a parametric study for the many parameters that rule its behaviour using the Travelling Salesman problem as the combinatorial problem to be solved. Having performed this parametric study, using an already developed code, making one able to understand better how the algorithm worked.

Afterwards, a review of the Topology optimization was conducted, where one introduced the most used approaches to solve this problem, such as the Homogenization approach and Density approach.

Having the review of the ACO algorithm and Topology optimization, using Kaveh et al. [2008] as the main reference, one explained how to implement the given algorithm to a topology optimization. Some changes were made to the implementation conducted by Kaveh et al. [2008], being the most meaningful one the neighbourhood system used and the pheromone initialization. One used 4-elements neighbourhood for the 2D optimization problem as the one used in Wu et al. [2009], instead of the 8-elements one, in order to avoid a connectivity analysis of the solutions generated. This allowed that every solutions generated will be composed only by elements selected by the ant's during the solution generation, avoiding the introduction of elements placed randomly to transform a disconnected solution in a connected one. The pheromone initialization used during the given implementation was the one used in Wu et al. [2009]. Using the strain energy of the elements with the full domain to perform the initialization will bring a more physical meaning than initializing with a constant value the pheromone matrix, helping also to have a more expedite solution generation during the first iterations.

With the implementation of the ACO for a topology optimization problem performed and to validate the code developed one solved the 2D problem of a cantilever beam with a point load applied and compared the results obtained with the ones obtained by Kaveh et al. [2008] performing along side a new parametric study of the algorithm. Taking into account that the problem solved is different from the ones usually solved by the algorithm, new parameters were added to algorithm. As a result of the parametric study one was able to find the set of parameters that suited best the algorithm resulting in the best solution found for the literature problem, resulting in a structure with a strain energy equal to 4.875 J, matching the best solution found in the literature case with a strain energy equal to 4.887 J. This set of parameters were used to perform the topology optimization of the high aspect ratio wing.

A structural model with the aerodynamic loads for the NOVEMOR 7^{th} framework reference wing with AR = 12 was developed. Using the $MTOW$ of the aircraft and the flight condition, one predicted the angle of attack required using APAME a panel-method software, being equal to $4.854°$. Using the given value, one conducted a CFD analysis with CFX®, being the lift obtained equal to 272560 N for half wing. Afterwards, the structural model for the wing box was developed using ANSYS® Workbench.

The topology optimization of the wing box was first performed only allowing changes in the cross-section, meaning that the wing box has a constant cross-section along the spanwise direction. Two tests were conducted, one for a volume fraction equal to 0.35 the other for 0.45 with 1152 elements. The minimum strain energy obtained at the end of the optimization process was equal to 14.783 J and 11.617 J, for a volume fraction equal to 0.35 and 0.45, respectively. With the increase of the volume fraction one has a decrease in the strain energy of the solutions obtained, this is related with the fact that the minimum strain energy will be obtained when the domain is full.

Lastly, a 3D topology optimization of the wing box was conducted. Three runs were conducted, two with 1664 optimization blocks and volume fraction equal to 0.35 and 0.45 and another one with 6656 blocks and a volume fraction equal to 0.35. The minimum strain energy obtained for the first two runs at iteration 30 was equal to 11.583 J and 9.830 J, for volume fraction equal to 0.35 and 0.45, for the third one the minimum strain energy was equal to 11.129 J.

ACO has some advantages to implement in a topology optimization problem, such as its simplicity to apply to a discrete problem as the topology optimization problem, whereas most of the deterministic approaches usually require continuous design variables needing additional strategies to overcome this problem. Another advantage that the ACO brings is the flexibility to use any given objective function not being restricted to the differentiable ones.

On the other hand, ACO displays some drawbacks, making it not the most adequate algorithm to solve a topology optimization. Being a meta-heuristic algorithm requires that the performance of the solutions generated to be accessed at each iteration, which for the given problem, even having its complexity reduced, leads to high computational time. The increase of the number of design variable is also strictly related with the computation time.

In conclusion, ACO does not correspond to the most efficient algorithm to solve the minimum compliance problem (minimization of the strain energy).

8.2 Future Work

Due to time and computational restrains, the problems implemented solved have a small scale when compared with the usual topology optimization problems. An increase of the design variables to give more freedom during the solution generation, would be interesting to see implemented. In order to overcome the increase in computation time, the introduction of a penalty function for the volume fraction in the objective function could be introduced. This change would only influence the pheromone update during the optimization process. The computation time would be decreased due to the fact that the ant's path which have surpassed the volume fraction assigned would not be discarded instead its performance would be penalized accordingly with volume fraction of the given solution.

The wing box was the only part of the wing optimized, the remaining structures that compose the wing were removed to turn the structural model simpler, the introduction of the remaining components of the wings would make the given model more realistic.

If a further topology optimization of the wing was performed, it would be interesting to consider the wing as solid which would be discretized corresponding this discretized domain to the optimization domain.

In order to further understand and have more information to compare ACO algorithm with a Deterministic approach to solve the topology optimization problem, a benchmark using the structural model developed should also be conducted.

Bibliography

D. Anderson and S. Eberhardt. *Understanding Flight*. McGraw-Hill, 2001.

ANSYS Manual. ANSYS Inc., 275 Technology Drive, Canonsburg, PA, 2013. Release 14.5.

D. L. Applegate, R. E. Bixby, and W. J. Cook. *The Travelling Salesman Problem: A Computational Study*. Princeton University Press, 2006.

M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, (1): 193–202, 1989.

M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.

Boeing. http://www.boeing.com/commercial/787/design-highlights/characteristics/787-8/, 2015. Accessed in November 2015.

D. Brandão. Computationally efficient framework for static aeroelastic analysis of flexible wings including viscous flow effects. Master's thesis, Instituto Superior Técnico, 2015.

V. Brederode. *Aerodinâmica Incompressível*. IST Press, 2014.

T. C. Corke. *Design of Aircraft*. Prentice Hall, 2003.

J. L. Deneubourg, S. G. S. Aron, and J. Pasteels. The Self-Organizing Exploratory Pattern of the Argentine Ant. *Journal of Insect Behavior*, 3(2):159–168, 1990.

M. Dorigo and T. Stützle. *Ant Colony Optimization*. The MIT Press, 2004.

M. Dorigo, V. Maniezzo, and A. Colorni. Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Man, Machine and Cybernetics-Part B*, 26:29–41, 1996. February.

M. Dorigo, M. Birattari, and T. Stützle. Ant Colony Optimization, artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, pages 28–39, November 2006.

EU. Flightpath 2050 Europe's Vision for Aviation. Technical report, European Comission, 2011. Maintaining Global Leadership & Serving Society's Needs.

FAA. Glider flying handbook, 2013. Federal Aviation Administration, U.S. Department of Transportation.

A. Filippone. Data and performance of selected aircraft and rotorcraft. *Progress in Aerospace*, 36: 659–654, 2000.

D. Filković. http://www.3dpanelmethod.com/, 2015. APAME, Accessed in September 2015.

L. S. Félix. *Topology Optimization of Aircraft Structures Including Self-weight*. PhD thesis, Universidade Técnica de Lisboa Instituto Superior Técnico, September 2013.

S. Goss, S. Aron, S. Deneubourg, and J. L. Pasteels. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989.

P.-P. Grassé. La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes Natalensis et Cubitermes sp. La théorie de la sigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–81, 1959.

J. M. Hammersley and D. Handscomb. *Monte Carlo Methods*. London Chapman and Hall, 1979.

O. Hasançebi and S. Çarbaş. Ant colony search method in pratical structural optimization. *International Journal of Optimization in Civil Engineering*, 1:91–105, 2011.

O. Hasançebi, S. Çarbaş, E. Doğan, and M. S. F. Erdal. Performance evalutaion of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Computer and Structures*, 87: 284–302, January 2009.

B. Hassani and E. Hinton. A review of homogenization and topology optimization II - analytical and numerical solution of homogenization equations. *Computers & Structures*, 69(6):719–738, 1998.

B. Hassani and E. Hinton. *Homogenization and Structural Topology Optimization: Theory, Practice and Software*. Springer, 1999.

ICAO. Aviation outlook, Environmental Report. Technical report, ICAO, 2010.

L. K. L. Jr. *The Evolution of Modern Aircraft*. NASA History Office, 1985. NASA Scientific and Technical Information Branch Washington, D.C.

A. Kaveh and S. Shojaee. Optimal design of skeletal structures using ant colony optimization. *International Journal for Numerical Methdos in Engineering*, pages 563–581, 2006.

A. Kaveh, B. Hassani, S. Shojaee, and S. Tvakkoli. Structural topology optimization using ant colony methodology. *Engineering Structures*, 2008.

J. Kennedy and R. C. Eberhart. *Swarm Intelligence.* Morgan Kaufmann Publishers, 2001.

K. S. Lee and Z. W. Geem. A new structural optimization method based on the harmony search algorithm. *Computers and Structures*, (82):781–798, January 2004.

G.-C. Luh and C.-Y. Lin. Structural topology optimization using ant colonu optimization algorithm. *Applied Soft Computing*, 9(4):1343–1353, 2009.

A. S. Metals. http://asm.matweb.com/search/specificmaterial.asp?bassnum=ma7075t6, 2015. ASM, Inc. Accessed in June 2015.

P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computer*, C-26(9):917–922, 1977.

NASA. http://www.nasa.gov/centers/armstrong/news/factsheets/fs-058-dfrc.html, 2015. Accessed in November 2015.

T. Noll, J. Brown, M. Perez-David, S. Ishmael, G. Tiffany, and M. Gaier. Investigation of the helios prototype aircraft mishap. Technical report, NASA, 2004.

M. Patil and D. Hodges. On the importance of aerodynamic and structural geometrical nonlinearities in aeroelastic behavior of high-aspect ratio wings. *Journal of Fluid and Structures*, 19:905–915, 2004.

RKUH. http://www.iwr.uni-heidelberg.de/groups/comopt/software/tsplib95/, 2015. Accessed in November 2015.

J. Roskam and C.-T. E. Lan. *Airplane Aerodynamics and Performance*. Design, Analysis and Research Corporation, Lawrence, Kansas 66044, U.S.A., 1^{st} edition, 1997.

G. Rozvany. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. *Structural and Multidisciplinary Optimization*, 21(2):90–108, 2001.

G. Rozvany and T. Lewi, editors. *Topology Optimization in Structural and Continuum Mechanics*, volume 549. Springer, 2014. CISM International Centre for Mechanical Sciences.

A. Schleicher. http://www.alexander-schleicher.de/en/flugzeuge/ash-30-mi/, 2015. AS, Accessed in November 2015.

O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001.

O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

C. Spada. Aeroelastic Analysis of Nonlinear High Aspect Ratio Wing. Master's thesis, Instituto Superior Técnico, October 2014.

T. E. Toolbox. http://www.engineeringtoolbox.com/standard-atmosphere-d_604.html, 2015. Accessed in September 2015.

USAF. *U-2 Utility flight handbook*, March 1959. Copy no 7.

Wang. http://www.mathworks.com/matlabcentral/fileexchange/14543-solving-symmetrical-and-dissymmetrical-tsp-base-on-ant-colony-algorithm, 2015. Accessed in November 2015.

C.-Y. Wu, C.-B. Zhang, and C.-J. Wang. Topology Optimization of Structures Using Ant Colony Optimizatio. pages 601–607. Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009. GEC -2009.

X.-S. Yang. *Nature-Inspired Optimization Algorithm*. Elsevier, Middlesex University, UK, February 2014.