INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Fuel Consumption Optimization using Neural Networks and Genetic Algorithms

Kevin Simões Spencer

Dissertation for the Degree of Master of

Aerospace Engineering

Jury

President: Prof. Fernando José Parracho Lau

Tutor: Prof. António José Nobre Martins Aguiar

Vogal: Prof. Maria Alexandra dos Santos Gonçalves de Aguiar Gomes

October 2011

Acknowledgments

I would like to express my gratitude to all those who helped and supported my ideas. A special thank you to all my family and loved ones for their support and patience. To my best friend Olga for the patience and support. To my tutor for his support and availability, To Rui Soeiro from TAP for his patience and availability in explaining to me the procedures used. To Professor José Carlos Fernandes Pereira and Professor José Manuel Chaves Pereira for the help given. To LEMAC´s manager and staff for their help and availability during this process. To Carlos Lagoa and Gonçalo Pardal for the wise help. To TAP for trusting me with such valuable information.

Resumo

Qualquer industria procura reduzir os seus desperdícios e optimizar os seus procedimentos. A indústria do transporte aéreo não é diferente e tenta maximizar os resultados por voo e reduzir o custo por lugar, disso dependendo a sua própria sobrevivência. Uma ferramenta que permita a qualquer companhia aérea prever o seu consumo de combustível ao longo do ano permitindo a aquisição de combustível quando esse fosse mais barato e seleccionar o avião/rota que minimizem o consumo de combustível da frota é da maior importância. Esta dissertação pretende criar a ferramenta com esse fim usando dados operacionais da TAP referentes ao ano 2010. Os aviões usados são os da frota actual da TAP, composta por 55 aviões. Uma ferramenta simples para a previsão de alguns parâmetros referentes ao estado do tempo para melhorar a prestação do algoritmo também foi desenvolvida, tal como uma interface gráfica para facilitar a introdução de dados e a manutenção do sistema. O software foi treinado, validado e testado para a frota, rotas, e destinos da TAP. Um algoritmo que muda de tipologia foi desenhado usando algoritmos genéticos e várias redes neuronais foram treinadas. A optimização do consumo de combustível foi conseguida usando algoritmos genéticos para a detecção da mínima global. Além disso o algoritmo fornece uma estimativa do $C0_2$, produzido por cada voo e o avião que deverá ser usado. O algoritmo também permite ao utilizador excluir aviões não disponíveis e maximizar a utilização da frota.

Palavras-chave:

Consumo de combustíveis, redes neuronais, optimização, algoritmos genéticos, ocupação de frota, transformação de topologia.

Abstract

Every industry thrives to reduce its waste and to optimize the procedures used. The air transport industry is no different and tries to maximize the revenues per flight and minimize the seat cost, its own survival being dependent on this. A feature that would allow any given airline the potential of predicting its fuel consumption throughout the year, select the plane/flight that minimizes the whole fleet consumption, make more precise estimations on the cost of fuel during a certain period of time and acquire the fuel when it is at its lowest price is of the utmost value. This dissertation proposes to create a tool needed for such purpose using TAP operational data from the year 2010. The plane population used is the current TAP own fleet, composed of 55 planes. A simple tool to predict a small number of weather parameters was also created to improve the efficiency of the algorithm as well as a graphical interface to allow an easy data input and maintenance. The software was trained, validated and tested for TAP fleet, flights and destinations. A topology shifting algorithm was designed using genetic algorithms and several neural networks were trained. The optimization of the fuel consumption was achieved using genetic algorithms to detect the global minima. Besides the fuel consumption parcels of each flight the algorithm provides an estimation of the CO_2 produced and the plane that should be used. The algorithm also allows the user to easily exclude planes that are not available, and to maximize the use of the fleet.

Keywords

Fuel Consumption, Neural networks, Optimization, Genetic Algorithms, Fleet allocation, Topology Morphing.

Table of Contents

Table of figures

Table of tables

Table of equations

Abbreviations

AI - Artificial Intelligence

NN – Neural Networks

GA - Genetic Algorithms

MSE - Mean squared error

APU - Auxiliary Power Unit

BPL - Back Propagation Learning algorithm

CG - Center of Gravity

ULD - Unit Load Device

SESAR - Single European Sky ATM Research

AFOB - Actual Fuel On Board

IATA – International Air Transport Association

AZFW – Actual Zero Fuel Weight

ZFW – Zero Fuel Weight

AFOB –Actual Fuel On Board

FOB – Fuel On Board

FRMN –Fuel Remaining

HLDDT – Holding Distance Fuel

ADDT – Additional Fuel

DOW – Dry Operating Weight

1. Introduction

1.1 Introduction

This chapter consists in the work definition. Here, the goals and the motivation of this work are explained.

1.2 Motivation

The aviation industry consumes 2% of all fossil fuels burnt. In 2005, the total jet A-1 fuel consumption represented 208 thousand million litres. Jet fuel, being the major component in the cost of operations for commercial airlines (Rao, 1999; Adams, 1997). For every $1 increase in oil price, United States airlines face an additional annual cost of $425 million (Alexander, 2004). Airlines have improved fuel efficiency by 14% in the last 10 years and those numbers continue to improve by investing in new aircraft and enhanced operations. The improvements made by airliners focus on the induced drag which increases with weight. If we take into account that flight altitude affects both parasite drag and engine efficiency it becomes clear that a good flight programming and the adequate airplane selection can and do make a difference.

Efficient flight planning is one of the most important factors on airlines operation departments. A good flight plan can, not only lead to a safer flight, but also lead to a more confident crew and significant fuel savings. The calculation of the actual fuel used on a flight depends on various factors and it's a nonlinear calculation. Many of these factors have very complex relations among them and are very difficult to predict. Because of the stochastic nature of those parameters, any normal linear approach would be flawed from conception. The knowledge of the fuel to be consumed by the fleet is extremely valued in today's airlines. Such knowledge can help airlines buy the fuel they will use throughout the year when the price is at its lowest and plan when to invest on the fleet. Such prediction could represent many months of calculation in the current system and would not take into account the weather, the plane degradation and the overall variability of the flight fuel consumption. To overcome that, a system would have to be created to learn all the variances of the fuel consumption throughout the largest possible time sample and, based on that information, future consumption could be predicted. Learning algorithms and artificial intelligence have been a part of complex problem solving for many years. Those algorithms find a pattern and can be instructed to mimic immensely complex processes that otherwise would be impossible to mimic. With that computational power virtually any problem can be solved, given the proper hardware. Based on the advantages of neural networks and genetic algorithms a complete fuel prediction system was developed.

1.3 Objective

The purpose of this thesis is to develop a tool that would allow the flight planner to introduce all the origins, destinations and dates of a set of flights and then the algorithm would not only select which plane would be more efficient doing a certain flight as it would as well produce an estimation on the fuel consumed taking into account weather parameters, plane degradation and available planes. The algorithm would also produce an estimation of the CO_2 emissions, the flight payload, several fuel consumption parcels and would thrive to optimize the fleet utilization. This would be achieved by training a neural network to predict fuel consumption based on as few user inputs as possible, and optimizing the fleet overall consumption using genetic algorithms to achieve the global minimum fleet consumption. The algorithm is to be tested, comparing its outputs to the real ones. Due to the learning characteristic of the algorithm, the same software can virtually be applied to all the future flights, needing only minor updates.

1.4 Approach

To better understand the problem at hand a complete design of the flight planning process had to be made. The flight plan establishes all possible procedures that occur on a given flight. Among those factors, the fuel amount is of paramount importance. Those values are derived from performance calculations, weather predictions and regulatory defined values. Regulations establish the minimum amounts of fuel to be used on a flight, taken into account the number of alternative airports, their distance and flight path. Although those minimum values are defined by law, the actual values which are greater than the minima are defined by the airline's fuel policy. To simulate all those parameters, a dataset that respects all the regulatory conventions would have to be used. It had to contemplate all the planes in the fleet and all the destinations presently flown. That dataset should also document flights throughout the year. This last condition is of paramount importance, since a big dataset concentrated in only a month or a season would not give the algorithm a sense of change of consumption throughout the year due to weather changes and load factors. Such dataset was made available from TAP in the form of the fuel data for all the flights flown in the year 2010. These flights were operated by the 55 planes that compose the fleet and to all destinations currently on TAP's operation. The dataset contains 71,855 flights operated by TAP. A detailed analysis to the dataset showed that the dataset had few flight data other than the fuel consumption. As explained later, the neural network needs as many input arguments as possible to correctly predict the results. As the Chaos Theory Butterfly Effect states that the presence or absence of a butterfly flapping its wings could lead to creation or absence of a hurricane. Ideally, if all the butterfly wing flapping parameters and the overall initial conditions were known, the consequences of that flapping could be predicted with great accuracy. The same theory also states that the initial conditions are of great importance in predicting very complex outcomes. As such, the algorithm should have as many relevant inputs as possible to efficiently predict the output. The fuel consumption on any given flight is greatly influenced by the weather conditions. With that in mind, a good fuel consumption prediction depends greatly on

the weather parameters. Given the complexity of the weather and the limited computational power available a reduced number of weather parameters would have to be chosen for the prediction. The following parameters were chosen: maximum temperature, average temperature, minimum temperature, average humidity, average pressure, wind direction, wind maximum speed and maximum gust. The temperatures were presented in degrees Celsius, the humidity in percentage, the wind direction in cardinal direction and the wind and gust in km/h. Those parameters were chosen firstly due to their relevancy to aeroplane performance and due to their overall availability for study. Due to the ease of access, up to date data and the option to export data to an easily manipulated file [30] were chosen. An algorithm was designed to obtain those parameters from all TAP destinations from 2006 to 2010. The data were available for all the days of the mentioned years. That information was stored in a matrix to be used as the target for the training of the weather prediction neural network. This approach is not new and is well documented. The usual approach used in many weather prediction agencies, predicts the weather using vast computer farms and extremely heavy and slow neural networks to predict the state of the weather. This approach demands computation power far greater than the one available for this study. As such, a different approach had to be created. The main problem of training neural networks is that the number of neurons and layers may be infinite.
Considering the Equation 1 for combinations:

$$C(n,r) = \frac{n!}{(n-r)!} \quad (1)$$

With *r* equal to the number of layers and *n* equal to the maximum number of neurons in each layer,
Even with a relatively small network, with only two layers and a maximum of 20 neurons per layer there are 380 possible combinations. A considerable network with 10 layers would have 670.44 thousand million possible neuron configurations. Clearly a too large number of simulations to be even remotely applied on a reasonable timeframe. With a very powerful computer, each network could be trained on 60 seconds which would represent 1.27 million years of computation, not to mention the trouble of evaluating all those networks and the time needed to load the input data. Thus, a new approach had to be taken. MATLAB©'s Neural Network Toolbox allows us to create custom neural networks. The software is built with algorithms that allow the training to be as fast as possible and using the mean squared error as an optimization function, giving the best weights between neurons as possible. A complete Evolutionary network would require that the software dictates the weights and biases inside the Neural Network to help the network convergence. Since the neural network studies created using the MATLAB© Neural Network Toolbox proved to be well behaved and with a good convergence, an Evolutionary Topology Network was the best solution for this particular problem. An Evolutionary Network that changes its topology is far beyond the scope of this study since its complexity is greater than its possible advantages in this particular problem. Because the main problem is to decide the number of neurons on each layer, it was decided to fix the number of layers and let the Genetic Algorithm evolve the number of neurons for each layer. Two configurations were chosen: 2 and 5 neuron layers.

2. Literature review

2.1 Artificial intelligence

Intelligent systems is a term normally used for systems that have a capacity to acquire and apply knowledge in an “intelligent’’ manner and have the capabilities of perception, reasoning, learning and making inferences from incomplete information. This feature is of crucial importance when we have systems too complex to monitor, or with too many different inputs. The flight planning is one of those complex systems that will be well suited for intelligent systems. Throughout the history of mankind we have developed systems to help and assist our existence. The most important feature of those systems is the generation of outputs, based on some inputs and the nature of the system itself. Quite commonly the generation of outputs involves some kind of decision making. This process of decision making may range from numerical computations to complex knowledge-based decision-making using natural and artificial intelligence, as in modern decision support systems. For very complex systems, even when the modelling is possible, the models could be so complex that accurate algorithms or decision-making based on these models, could considerably increase the costs of computing and hardware, and make the process too slow to be feasible. Knowledge-based systems with intelligent decision-making capabilities are very successful on those types of problems. In the near future, it is expected that industrial machinery and decision support systems maintain consistency and repeatability of operation and cope with external performance without visible degradation of performance. Artificial Intelligence (AI) can be defined then as, according to Marvin Minsky of the Massachusetts Institute of Technology (MIT), “artificial intelligence is a science of making machines do things that would require intelligence if done by men”. The field of soft computing is a major contributor to realizing intelligent machines, thereby broadening the meaning of machine intelligence by way of developments through a different path from traditional AI. It wasn’t until the 1960s that significant activity in the field of artificial intelligence began with the objective of developing computers that could think like humans. Just like neurons in the brain, the hardware and software of a computer are themselves not intelligent, yet it has been demonstrated that a computer may be programmed to demonstrate some intelligent characteristics of a human. A complete yet simple definition of intelligence is not available. It is commonly accepted that an intelligent system possesses one or more of the following characteristics and capabilities:

- Pattern Recognition
- Learning and knowledge acquisition
- Inference from qualitative or approximate information
- Ability to deal with unfamiliar situation
- Adaptability to new yet related situations (through expectational knowledge)
- Inductive reasoning
- Sensory perception
- Common sense
- Display of emotions
- Inventiveness

The first six characteristics on the list are the most important in this document since they represent characteristics that are needed to predict the amount of fuel consumed. The last three of the list are the most difficult to implement, and for that reason, rarely seen on intelligent machines. An intelligent machine is a machine that can exhibit one or more intelligent characteristics of a human. In broad terms, an intelligent machine may be viewed as consisting of a knowledge system and a structural system. The knowledge system affects and manages intelligent behaviour of the machine, loosely analogous to the brain, and consists of various knowledge sources and reasoning strategies. The structural system consists of physical hardware and devices that are necessary to perform the machine objectives yet do not necessarily need a knowledge system for their individual functions [1].

2.2 Knowledge-based systems

The method of knowledge-based systems is most likely the most notable paradigm to have found widespread application, particularly in the process and manufacturing industries, medical, business, education and legal fields. This technique can represent a controller or a decision support system according to a number of architectures, and is able to handle heuristic and comprehensive computer programs which solve problems within a limited and specific field, using data on the problem, knowledge related to the problem and incorporating decision-making capabilities. It typically emulates the problem-solving behaviour of a human expert in the field. In this case of fuel prediction it tries to emulate the problem-solving of a flight planner and process analyst. This is different from the conventional algorithmic programming technique where the solution to a problem can be obtained by executing a prescribed set of steps. In industrial plants, for example, there are situations involving a significant number of variable factors such as changing plant characteristics, unexpected disturbances, system wear and tear, and different fault and alarm combination, here the approach of knowledge-based systems has certain advantages and flexibility which can make the method particularly appropriate or attractive in such systems.

2.3 Soft Computing

Soft computing techniques resemble biological processes more closely than traditional techniques, which are largely based on formal logical systems, such as sentential logic and predicate logic, or rely heavily on computer-aided numerical analysis. Unlike hard computing schemes, which strive for exactness and full truth, soft computing techniques exploit the given tolerance of imprecision, partial truth, and uncertainty for a particular problem. Soft computing is a very important branch of study in the area of intelligent and knowledge-based systems. It has effectively complemented conventional AI in the area of machine intelligence. Human reasoning is predominantly approximated, qualitative, and ''soft''. Humans can effectively handle incomplete, imprecise, and fuzzy information in making intelligent decisions. Fuzzy logic, probability theory, neural networks and genetic algorithms are cooperatively used in soft computing for knowledge representation and for mimicking the reasoning and decision-making processes of a human. Quite effective are the mixed or hybrid techniques, which synergistically exploit the advantages of two or more of these areas. On this project the challenge is to create a hybrid system derived from neural networks and genetic algorithms. For that reason the Genetic Algorithms and the Neural Networks will be explained in a more detailed manner.

2.4 Neural networks

2.4.1 Introduction

Neural Networks (NN) are massively connected networks of computational “neurons” and represent parallel-distributed processing structures . The inspiration for NN came from the biological architecture of neurons in the human brain. A key characteristic of neural networks is their ability to approximate arbitrary nonlinear functions. Since machine intelligence involves a special class of highly nonlinear decision-making, neural networks would be effective there. The process of approximation of a nonlinear function by interacting with a system and employing data on its behaviour may be interpreted as “learning”. Through the use of neural networks, an intelligent system would be able to learn and perform high-level cognitive tasks. A neural network consists of a set of nodes, usually organized into layers, and connected through weight elements called synapses. At each node the weighted inputs are summed, thresholded, and subjected to an activation function in order to generate the output of that node. These operations are shown on Figure 1. If the weighted sum of the inputs to a node (neuron) exceeds a threshold value w_0 , then the neuron is fired and an output $y(t)$ is generated according to Equation 2, with x_i being the neural inputs.

$$y(t) = f\left[\sum_{i=1}^{n} w_i x_i - w_0\right] \qquad (2)$$

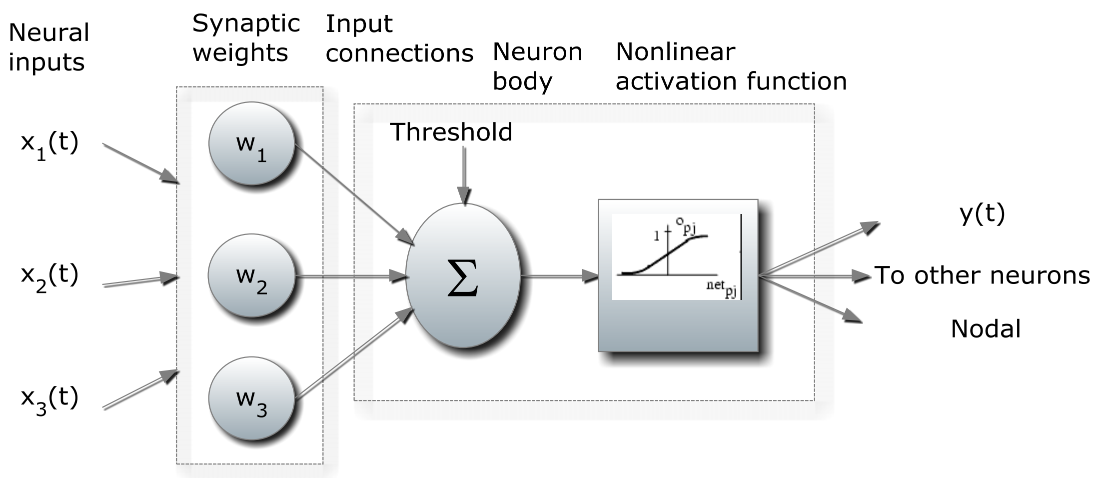

Figure 1 - The operations at a node of a neural network

Where x_i are neuron inputs, w_i are the synaptic weights and $f[.]$ is the activation function.

There are two main classes of neural networks known as feedforward networks and feedback networks. In a feedforward network, an example is a multilayer perceptron where the signal flow from a node to another node takes place in the forward direction only. In a feedforward neural network, learning is achived through examples. This is known as supervised learning. The process works on two basic steps. In the first step a set of input-output data of the actual process is determined and the input data are fed into the NN. In the second step the network output is compared with the desired output and the synaptic weights of the NN are adjusted using a gradient algorithm until the desired output is achieved. In a feedback NN, the outputs of one or more nodes are feedback to one or more nodes in a previous layer or even to the same node. The feedback provides the capability of "memory" to the network. Unlike computers, which are programmed to solve problems using sequential algorithms, the brain uses a massive network of parallel and distributed computation elements called neurons. The large number of connections linking the elements provides humans with the very powerful capability of learning. Artificial neural networks (ANN) were born from the need to have systems that can process information in a similar way. They are composed of a large number of highly interconnected processing elements analogous in functionality to biological neurons and are tied together with weighted connections corresponding to brain synapses.

2.4.2 Learning and acquisition of knowledge

Learning is accomplished in general by developing algorithms that allow the system to learn for itself from a set of input/output training data. Once a learning capability has been implemented within a given system, it continues acquiring knowledge and uses the stored information to draw right decisions for similar future situations. Two possible approaches for designing learning systems employ symbolic-learning and numerical-learning based techniques.

2.4.2.1 Symbolic learning

Symbolic learning pertains to the ability of a given system in formulating and updating its knowledge base form a set of well-structured feedback data related to the performance of the system. Several categories of symbolic learning exist. These include rote learning, explorational learning, inductive learning and explanation-based learning. What distinguishes those learning algorithms from one another is essentially the level of supervision.

2.4.2.2 Numerical learning

Numerical-learning-based algorithms represent a class of very useful learning tools used most often for the design of neural networks. They provide a given network with the capacity of adjusting its parameters in response to training signals. In some cases, learning is acquired by practice through training. In others the system is presented with a number of patterns and it is up to the network, through well-defined guidelines, to group these patterns into categories. This is done by solving an optimization problem involving *m* patterns of desired responses $t^{(1)}, t^{(2)}, \dots, t^{(m)}$ and their corresponding inputs $x^{(1)}, x^{(2)}, \dots, x^{(m)}$ and a set of *a priori* unknown optimization parameters called connection weights $W = [w_1, w_2, \dots, w_p]$, with *m* denoting the number of training patterns presented to the network and the subscript *p* the number of interconnection weights among all nodes of the network. Each training input vector *x* is composed of *i* components x_i, while each training output vector *t* is composed of *q* components t_q. The optimization process seeks the minimization of the cumulative sum of the *error e* between the k-th desired output vector $t^{(k)}$ and the k-th output vector $o^{(k)}$given by $h(W, x^{(k)})$, corresponding to the actual output of the network in response to the excitation $x^{(k)}$. The optimization problem is expressed then as

$$min_w \sum_{k=1}^{m} e\left(o^{(k)}, t^{(k)}\right) = min_w \sum_{k=1}^{m} e\left(h\left(w, x^{(k)}\right), t^{(k)}\right) \qquad (3)$$

The training algorithm keeps adjusting the weights according to well defined learning rules. The basic learning process is shown on Figure 2.

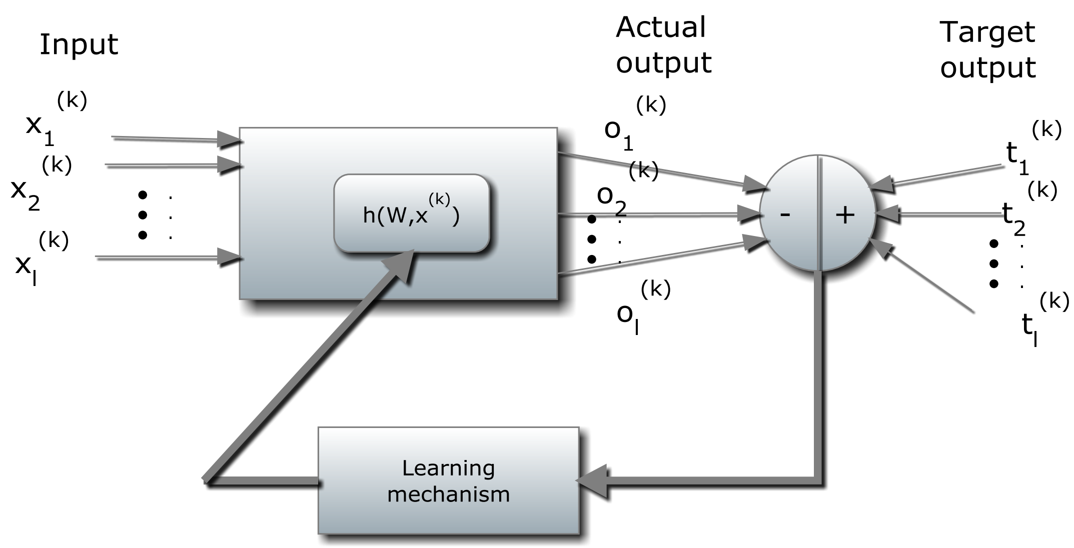

Figure 2 - Schematic representation of a supervised ANN

2.4.3 Neural network topologies

An ANN is typically composed of a set of parallel and distributed processing units, called nodes or neurons. These are usually ordered into layers, appropriately interconnected by means of weighted signal channels. Figure 3 represents a typical feedforward network where the connections represent the connection weights and the internal element represents neurons.

Neural networks gather their knowledge through detection of patterns and relationships found in data provided to them. Three important features generally characterize an artificial neural network: the network topology, the network transfer functions, and the network learning algorithm. Feedforward networks relate directly the input and the output. Recurrent networks relate the output with the input and the previous state of the neuron. The topology chosen was the feedforward due to its speed and the lack of a need to use the previous state of the neurons to predict the output because the problem at hand does not depend on the previous state of training process.

2.4.3.1 The Feed forward topology

A network with this configuration has its nodes hierarchically arranged in layers starting with the input layer and ending with the output layer. The internal layers are called hidden layers and provide most of the network computational power. The nodes in each layer are connected to the next layer through unidirectional paths starting from one layer and ending at the subsequent layer. This means that the outputs of a given layer feed the nodes of the following layer in a forward path. This configuration has been very popular due to its association with a quite powerful and relatively robust called Back Propagation Learning algorithm (BPL).

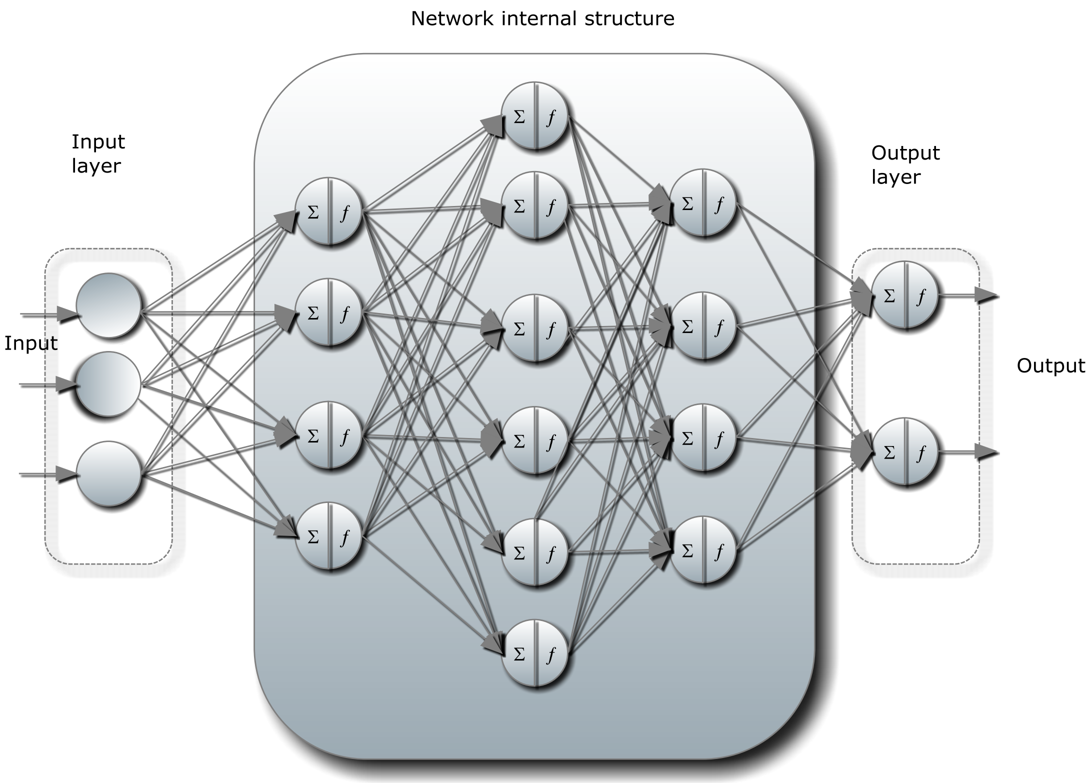

Figure 3-Feedforward artificial neural network representation

2.4.4 Neural network activation functions

The basic elements of the computational engine for a neural network are the neurons. They assume the task of small processors which take the weighted sum of their inputs form other nodes and apply them a nonlinear mapping called an activation function before delivering the output to the next neuron. Depending on the problem at hand and on the location of the node within a given layer, the activation functions can take different forms: sigmoid mapping, signum function, step function or linear correspondence. MATLAB® uses a Tan-sigmoid transfer function for the hidden layers and a linear function for the output layers. There is no consensus on which transfer to use and MATLAB® recommends this disposition.

2.4.5 Neural network learning algorithms

Learning algorithms are used to update the weighing parameters at the interconnection level of the neurons during the training process of the network. The three well-known and most used learning mechanisms are the supervised, the unsupervised and the reinforced. The unsupervised learning does not involve an external teacher and relies instead upon local information and internal control. The reinforced mechanism changes the network connections according to feedback information provided to the network by its environment. The supervised mechanism trains the network by examples. The network receives a set of input for which the output is known. During the training process, the output results are continuously compared with the desired data. An appropriate learning rule uses the error between the actual output and the target data to adjust the connection weights so as to obtain, after a number of iterations, the closest match between the target output and the actual output.
The supervised learning algorithm was the one chosen.

2.5 Genetic algorithms

2.5.1 Introduction

Genetic Algorithms (GA) are derivative-free optimization techniques, which can evolve through procedures analogous to biological evolution. Genetic algorithms belong to the area of evolutionary computing. They represent an optimization approach where a search is made to “evolve” a solution algorithm, which will retain the “fit” components in a procedure that is analogous to biological evolution through natural selection, crossover, and mutation. Evolutionary computing can play an important role in the development of an optimal and self-improving intelligent machine.

Evolutionary computing has the following characteristics:

(1) It is based on multiple searching points or solution candidates
(2) It uses evolutionary operations such as crossover and mutation.
(3) It is based on probabilistic operations

A genetic algorithm works with a population of individuals, each representing a possible solution to a given problem. Each individual is assigned a fitness score according to how good its solution to the problem is. The highly fit individuals are given opportunities to reproduce by crossbreeding with other individuals in the population; this produces new individuals as offspring, who share some features taken from each parent. The least fit members of the population are less likely to get selected for reproduction and will eventually die out. An entirely new population of possible solutions is produced in this manner, by mating the best individuals from the current generation. The new generation will contain a higher proportion of the characteristics possessed by the "best fit" members of the previous generation. In this way, over many generations, desirable characteristics are spread throughout the population, while being mixed and exchanged with other desirable characteristics in the process. By favouring the mating of the individuals who are more fit, the most promising areas of the search space would be exploited. A GA determines the next set of searching points, which are widely distributed throughout the searching space, and uses the mutation operation to escape from a local minimum.

Standard single-point-based optimization techniques have several deficiencies. Gradient-based methods require knowledge of the function's derivative or partial derivatives and there is no guarantee that they provide the global optimal solution. They are only efficient when the problem is well defined and has a relatively simple objective function. Genetic algorithms tackle problems that are ill-modelled or have multi-objective functions.

The basic idea is to represent every individual of the potential solution as an array of sequences of strings, chromosomes. Each string in the chromosome is called a gene and the position of a gene is called its locus. The values that genes might take are called alleles. The initial population of the potential solutions is created randomly and it evolves according to processes that are based on natural evolution, such as selection, recombination or crossover, and mutations. The mutation process is applied to add diversity to the potential solutions. The basic evolution cycle is shown Figure 4 .

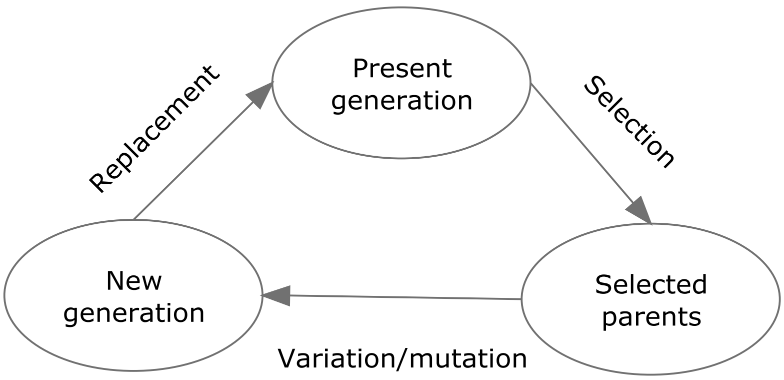

Figure 4-Basic evolution cycle

GAs are non-comprehensive search techniques used to determine, among other things, the global optimum of a given function that may or may not be subject to constraints. The search procedure of GAs is stochastic in nature and doesn't usually provide the exact location of the optima as some other gradient-based optimization techniques do. However, GA-based techniques possess two attractive features putting them at an advantage with respect to their derivative-based counterparts. While GAs may not provide for the mathematically exact solution of a given optimization, they usually outperform gradient-based techniques in getting close to the global optimum and hence avoid being trapped in local ones.

The most important aspect of an evolutionary process pertains to the way a composition of a population changes. Those changes occur due to the selection, mutation and crossover.

2.5.2 Selection

Selection is applied to select the individuals that participate in the reproduction to the next generation. Selection operators usually work on a population, and may serve to remove weaklings, or to select strong individuals for reproduction. Selection operators are stochastic, probabilistically selecting good solutions and removing bad ones based on the evaluation given to them by the objective function. There are several heuristics in this process, including the elitist mode, where the top 10 to 20 individuals of the population are chosen for further processing; the ranking model, where each member of the population is ranked based on its fitness value; and the roulette wheel procedure.

2.5.3 Crossover

Crossover is derived from the natural phenomenon of mating, but refers most specifically to genetic recombination, in which the genes of two parents are combined in some more or less random fashion to form the genotype of a child.

2.5.4 Mutation

It has been widely recognized that mutation is the operator that creates completely new solutions while crossover and selection serve to explore variants of existing solutions while eliminating the bad ones. In genetic algorithms, mutation operators select genes in an individual at random and change the allele.

2.6 Fuel optimization

One of the most common ways to reduce fuel consumption is to use more efficient planes. Planes that can fly longer and faster with little fuel are praised in the aircraft industry. Biggest gains in fuel savings can be achieved through aerodynamic improvements on a certain aircraft. The problem becomes more complex when the airline already has a fleet and has no plans of changing it. This is mostly due to the significant cost of a new, efficient aircraft or engine and the long lead time of the process. Winglets installation has a short lead time but have high capital expenditure intensity. The preferable improvements are those who have short lead time and low capital expenditure intensity.

Those improvements include:

- Weight Reduction
  - Optimized Fuel Loading
  - Optimized Water Loading
  - Aft.CG Position
- Flight Planning Optimization
- Aircraft Maintenance
  - Flight Control and Door rigging
  - Flight Control Seal integrity

The fuel saving per flight in kilos can be up to 20 % of the weight carried per 3000 nautical miles. With that perspective, it is clear that, the lighter the plane the more efficient and less fuel is needed. A 1000 kg of weight reduction would save up to 200kg of fuel per 3000 nm.

The center of gravity should be placed as aft as possible to reduce the drag produced. The aft limit is defined by the flight envelope and as such should be respected. The extra fuel added to the plane is selected by the cockpit crew. This amount varies depending on the crew experience, weather conditions and route chosen. Although those values should continue to be dictated by the commander, an indicative value can be suggested. This value, would then, be assessed by the crew for approval. This way, the algorithm gives the crew extra information on how much fuel the algorithm "thinks" to be necessary.

Removing unnecessary items from the cabin such as old phone equipment, logo lights, magazine racks and razor outlets can save a significant amount of fuel.

HONEYWELL Aerospace is working in a joint venture with SAFRAN, to build an electric taxiing system that will use the aircraft's auxiliary power unit to power electric motors in the main landing gear. The fuel lost on taxiing is significant due to the fact that aeroplanes use their engines during their taxi where they often have to idle for long periods. Short-haul flights could cut their fuel consumption by up to 4% according to Honeywell.

Other techniques can be applied and have shown good fuel saving capabilities. Those techniques are:

1. **Aero drag**

   Roughness on aircraft exterior surfaces increases drag and fuel consumption. The creation of a program to routinely inspect aircraft exterior surfaces to identify and correct defects such as chipped pain, scratches and damaged seals can save up to 0.41% of fuel annually. [2]

2. **Engine water wash**

   Engine water washing consists on regular engine washing to remove accumulated dirt. This cleans the engines inside while they are still on the aircraft wings and allows them to burn fuel more efficiently. Making allowance for the cost of acquiring the washing equipment, biannual engine cleaning can reduce the annual fuel consumption by 0.42 % .[2].

3. **Lighter-weight tires**

   Lighter tires can be installed in the aircraft without a decrease in performance, safety or service life and without additional cost. New tires can be up to 6 kg lighter than the ones they replaced and save up to 48 kg per aircraft. This can represent a saving of 0.02%.[2]

4. **Potable water**

   Many airlines operate their aircraft with full potable water containers even though less is needed for shorter flights. Studying the amount of water spent on the airline's flights and carrying the minimum amount possible to guarantee no shortage of water en route can save up to 100kg per aircraft and reduce annual fuel consumption by up to 0.09 %. [2]

5. **Auxiliary Power Unit usage**

   For an aircraft stationed at an airport that offers ground power the best process is to start its APU 15 minutes before take-off. For arrival, the best process is to connect the aircraft to ground power no more than five minutes after it reaches the gate. APU use should average 60 minutes per flight. Flight Sciences calculated that a 70-minute reduction in APU use saves 73 US gallons of fuel per flight for A310 aircraft and 84 US gallons per flight for A330 aircraft, for a potential annual fuel saving of about 0.20%.[2]

6. **Load Container weight**

   The use of lightweight Unit load devices (ULD) should be encouraged. The difference in weight between the heavy and light versions of containers is 8 kg for the LD3, used for luggage, and 25 kg for the LD6, used for cargo. Airlines can choose to lease the lighter ULD's and that way save up to 0.02% of fuel annually. [2]

7. **Single engine Taxi**

   Consists on single-engine taxi between the gate and the runway where possible - not all airports allow it and some winter conditions make it problematic. Single-engine taxing has reduced annual fuel consumption up to 0.40 %.[2]

8. **Airplane coatings**

   Manufacturers of the ultra-thin nanotechnology as TripleO have created special paints that can save airliners up to 2% of fuel. The paint works by bonding to the aircraft's paint and adds very little weight to the aircraft. It is aimed at reducing the build-up of debris on the aircraft – and therefore drag (friction drag). [2]

3. Experimental Program

3.1 Introduction

In this chapter the experimental program is defined and explained in greater detail. It is here that the choices made are defined, the parameters are chosen and the solutions are created. The software is also defined in this section and its functionality and structure explained.

3.2 Experimental program overview

The neural network training was done using MATLAB® Neural Network Toolbox® version 7.1. The training process is the most sensitive and time consuming part of the algorithm. A well trained network gives good results given any input. A not well trained network on the other hand will give erratic results if the inputs given are different from the input used to train. Because the datasets have a significant size and because the network also needs inputs to validate and test the network, the proper division of the data set would have to be made. Ideally, the bigger the training dataset, the more accurate the network is. In practice a network with a too big training dataset will predict very well the relation between the output and trained input but will give erroneous values for inputs not included in the training dataset. This process is called *over fitting* and it occurs when the network is too trained. Because it is needed that the network predicts fuel consumptions, not only of the year 2010 as those in the foreseeable future, it is of paramount importance that the over fitting problem be avoided.
It is also clear that a too small training dataset will generate a loose neural network, simply because it didn't have enough examples and opportunities to refine its weights and biases. The exact division between training, validation and test datasets is not universally defined. The process is shown on Figure 5.

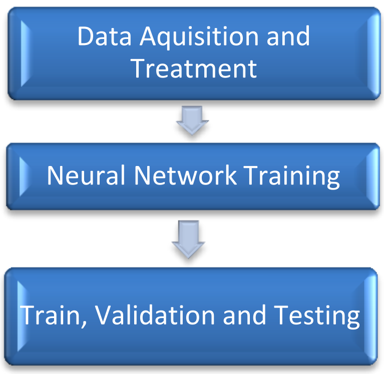

Figure 5-Neural Network Sequence

MATLAB® recommends that the dataset is divided into 70% of training, 15% of validation and 15% of testing. Because the training process is the most time consuming, its percentage of the global data set can be reduced for speed. Doing that, and increasing the percentages of validation and testing, the neural network will have a more restrict validation condition to obey, since its performance has to be good on a bigger data set. For that reason the division of the dataset was chosen depending on the performance of the neural network at study. The test part is used mainly on calculation of the neural network performance. Due to its evolving topology, the network has to have two different performance parameters. One of those parameters is used inside the neural network to help the network evaluate its training performance. The Mean Squared Error (MSE) was selected as the performance function inside the neural network due to its relevance on the neural network performance. Because the MSE is calculated for each neural network iteration, it can't be used as the overall neural network performance parameter. A parameter had to be created that evaluates the performance of the trained and validated neural network. This parameter would have to assess the overall network performance. Because this would be the parameter used to classify a certain network topology it should be calculated the same way on every single network. The parameter chosen is the regression parameter that relates the targets and the actual outputs, where *Targets* are the objective values and the *Outputs* the actual values produced.

$$R = \frac{Targets}{Outputs} \tag{4}$$

.

Equation 1-Regression Parameter

Because genetic algorithms try to minimize its performance function, the inverse of this parameter was used. The genetic algorithm variables are the number of neurons on each neuron layer and its performance function is R. There were created a total of 4 neural networks, 2 for the weather prediction and 2 for the actual fuel prediction. For the weather prediction 2 topologies were chosen: 2 layers and 5 layers. Several validation parameters where set and the overall consistency of the network determined. The training process was made on the hardware specification shown on Table 1.

CPU	Memory
• **8× Intel® Xeon®** • **CPU E5520 @ 2.27 GHz.** • **4 Cores.**	• Size:49562540 kb • Buffer: 144080 kb

Table 1- Hardware specification

3.3 Training parameters

3.3.1 Neural Network Parameters

The neural network, as built, only requires convergence parameters. All the other parameters are either random or calculated automatically by MATLAB®. The dataset parameter was changed for each neural network trained in order to obtain the best performance. For the weather prediction network the division of the dataset was set to 50% for training, 25% for validation and 25 % for testing on the 2 and 5 layers configurations. For the Fuel Prediction Network the configuration was 70% for training, 15% for training and validation. The size of the training set on the final network was bigger due to the fact that the dataset on this network is smaller than the one used on the weather prediction network. Due to the importance of this network on the final results, it was preferable to allow it to be optimized for a longer period of time.

As explained in the introduction, the performance function used was the MSE. MSE measures the average of the squares of the errors. The error is the amount by which the value implied by the estimator differs from the quantity to be estimated. The training function used was the *trainlm* that is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. This function was chosen because it is often the fastest back propagation algorithm in the MATLAB® Neural Network Toolbox, and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms. Since time was a major concern and not memory, the server used had available 48 Gb, this training algorithm suited well the problem in hands.

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix, H, can be approximated as ,

$$H = J^T J \tag{5}$$

And the gradient can be computed as

$$g = J^T e \tag{6}$$

Where *J* is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and *e* is a vector of network errors. The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{7}$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, the function becomes a gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. This way the performance function is always reduced at each iteration of the algorithm.

The other parameters used are the training parameter maximum number of validation failures allowed for a given network, set to 4, minimum performance gradient, set to 1×10^{-4} and the train performance variance set to 0.001.

3.3.2 Genetic Algorithm Parameters

For the genetic algorithm several parameters were set to ensure the evolution of the population and a fast convergence. The population initial range was set between zero and twenty. More than 20 neurons per layer represent a network too complex to be computed easily. The crossover fraction was set to 0.8. This parameter specifies the fraction of each population, other than elite children, that are made up of crossover children. A crossover fraction of 1 means that all children other than elite individuals are crossover children, while a crossover fraction of 0 means that all children are mutation children. The migration interval was set to 20. This parameter specifies how many generation pass between migrations. For example, if one sets migration interval to 10, migration takes place every 20 generations. The migration fraction data was set to 0.2. This parameter specifies how many individuals move between subpopulations. The maximum number of generations was set to 3000. This was a parameter defined to guarantee that the algorithm would be running and would need to be stopped by the user. Since server time was restricted the neural networks would have 48 hours to optimize themselves. After that period the simulation was stopped and the best values saved. The migration direction was set to be on both directions, which allows the subpopulation to migrate in both directions. For the crossover function, the heuristic function was selected. This function returns a child that lies on the line containing the two parents, a small distance away from the parent with the better fitness value in the direction away from the parent with the worst fitness value. The ratio was set to 1.2. Concerning the mutation, the mutation function selected was the Gaussian function. This function adds a random number taken from a Gaussian distribution with mean 0 to each entry of the parent vector. The standard deviation of this distribution is determined by the parameters Scale and Shrink, which were both set to one. The population initial size was also set to three. This means that the initial data is composed of 3 configuration options from which the algorithm evolves. Those configurations were generated randomly. Setting a too big initial population would slow down considerably the simulation. These parameters were used in all the neural networks. The algorithm process is shown on Figure 6.

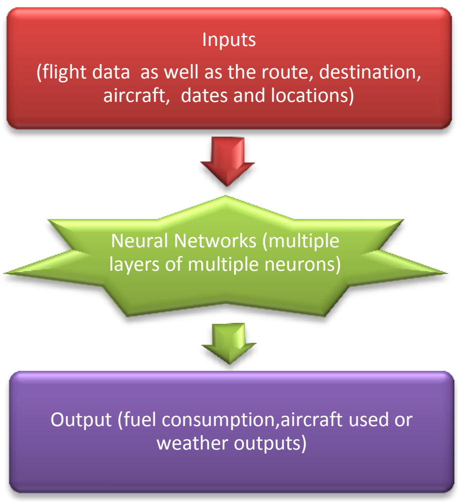

Figure 6-Algorithm Process

3.4 Fuel optimization algorithm

The fuel optimization algorithm was an algorithm designed to select which pair plane/destination, would have the smallest total consumption. Since the optimization does not consist on selecting the plane that has the smallest consumption on a certain route, it is more complex than a mere simple selection problem. For that reason the Genetic Algorithm was used. The problem in hands is a multiple input, multiple output optimization. Due to the fact that genetic algorithms are used with great results, it was the obvious choice for this predicament. The algorithm was designed to have as many variables as flights demanded. On the final optimization that means 71855 variables. Each variable can assume a value between 1 and 55. The variable represents the flight and its value represents the plane. The planes were organized by model and by registry. This attribution means that if the variable "one" which represents the first flight asked has a value of 44, the plane that the genetic algorithm chooses for that flight is the one on position 44 of TAP planes list. This methodology guarantees therefore that the value selected to get best fuel consumption is the plane used. The optimization function calculates the fuel consumption. The main function of the optimization function is to control the values of the variables defined by the Genetic Algorithm. It penalizes the algorithm, giving it a bigger than possible consumption values when the algorithm chooses values that are not possible. One of these limitations is the fleet allocation, the algorithm searches if the genetic algorithm has demanded for the same plane to do 2 flights at the same time. The algorithm checks the time at which the flight should depart and defines when the next one, using the same plane, could start. This fleet management is done adding to the time of departure the duration of flight and an allowance for taxi to take into account the time that the plane needs to be ready for the next flight. This factor was taken as 60 minutes. The algorithm also penalizes the function greatly if it selects a plane for a flight that is out of its range. Using those penalties the algorithm chooses the best value, respecting basic selection rules. The algorithm also penalizes the function if the plane selected is not available. In the end of the optimization the algorithm sends the best pairs to the main software that optimizes. Special population and mutation were created to achieve better results. Crossover scattered that creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child, was chosen as a crossover function due to its ability to evaluate integer populations. The fuel optimization process is shown in Figure 7.

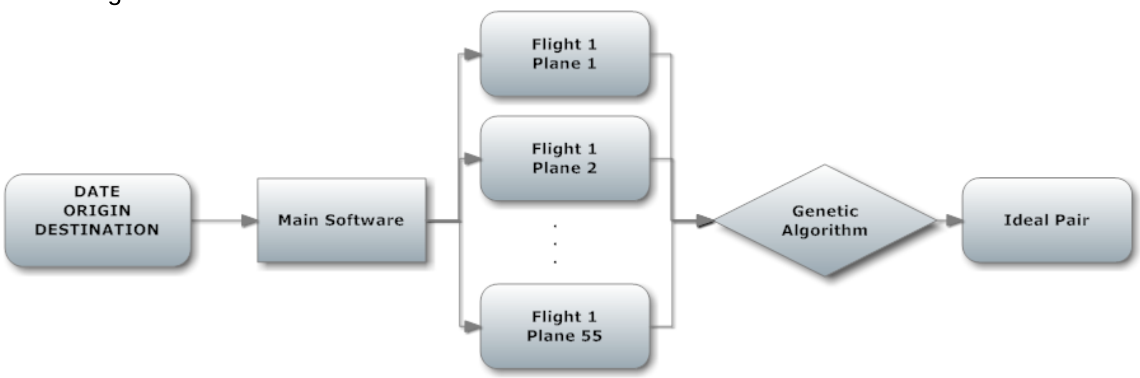

Figure 7-Fuel Optimization Schematic

3.5 Main Algorithm

The algorithm was designed in separate blocks. Each block works alone and interacts with the central algorithm. This allows for an algorithm that can be easily updated and trained, without the need for change in the main algorithm. Because neural networks can be easily trained, the most fundamental update would be to enter more recent datasets and annually train the network to ensure a better performance. The algorithm layout is presented in Figure 8.

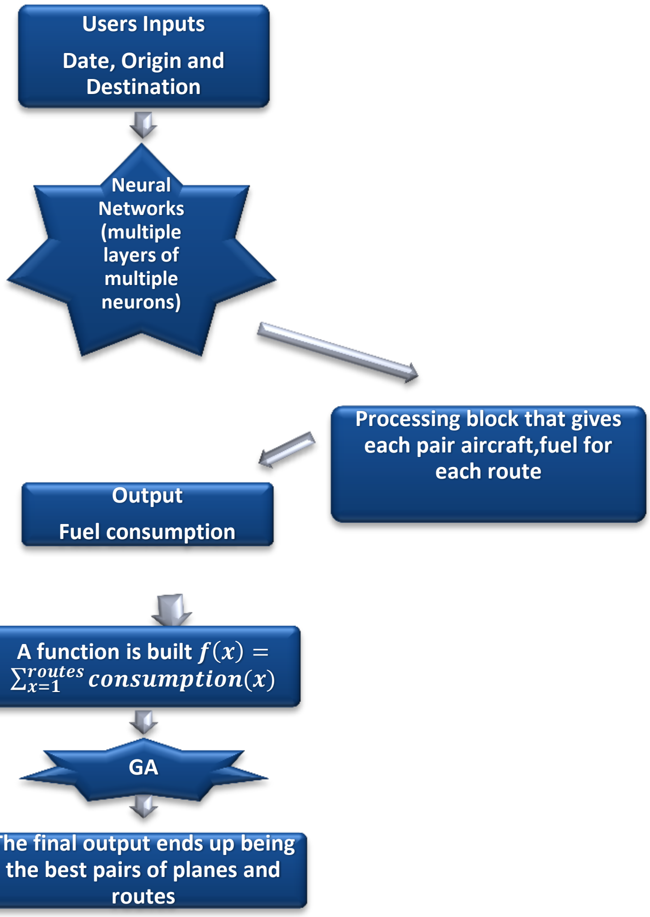

Figure 8-Detailed Algorithm Description

The output presented to the user has the following format:

Route aircraft consumption= $\begin{bmatrix} route\ 1 & aircraft\ 1 & consumption\ 1 \\ route\ 1 & aircraft\ 2 & consumption\ 2 \\ route\ 2 & aircraft\ 1 & consumption\ 3 \end{bmatrix}$

3.6 Main Algorithm Functionality

The main algorithm has two main screens. The first screen allows the user to choose the input method. This feature was designed to allow for mass data entrance and for simple input of a reduced number of flights. It allows the fleet manager to, in a matter of minutes, optimize between the planes available and choose which one should be used for each flight and the predicted fuel consumption. The mass input allows the user to enter the path to a .txt file which contains comma separated values. An example file is shown in the annexes. The user can also input a file that contains planes that are not available for the algorithm. If no file is selected the algorithm assumes that all the planes on the fleet are available. Both the secondary screens have this option. Entering the dates, origin, destination and planes not available, the optimize button will run the fuel prediction algorithm. This algorithm processes those inputs and converts them to a format that the software understands. Primarily the algorithm builds the data needed for the weather prediction software. After obtaining these values, it calculates the state of the weather for the route. That information is added to a matrix containing the distance from the origin to the destination. After that entrance, each route is multiplexed in *n* other routes, *n* being the number of planes on the fleet. In this part of the algorithm the software doesn't take into account which plane is available. Since the state of the weather is constant for a certain flight and does not vary with the plane used, this value is the constant for each block of n routes. By now the system has n times the number of entries demanded by the user. This input is sent to the optimization algorithm. The optimization algorithm selects the best pairs that minimize the fleet consumption and calculates the predicted emissions for each flight. The software also allows the user to specify the number of seats that the flight planner predicts that will be occupied on each one of the entered flights. For the calculation of the final result the user can also enter the maximum time spent with the optimization. The final result is sent to the main software that saves the values in a Microsoft Excel® spreadsheet for easy viewing. The file is saved in the root folder of the computer with the name Fuel_Optimization_Results.xls.

4. Results presentation and discussion

4.1 Introduction

In this section the results of the neural networks will be presented and discussed. The different configurations chosen will be detailed and the differences shown.

4.2 Weather Prediction

4.2.1 Weather prediction- 2 layers

Concerning the weather prediction network, the small network, 2 layers, proved to be the most reliable due to its speed and good results. This is due mostly to the fact that this network was faster and as such could, in the same time interval, train itself more and get for that reason better performance values. The maximum R obtained by this network was 99.3 % which is a very good value if we take into account the amount of data used to validate it and the difficulty in predicting the weather. We should take into account that the R value is an averaged value for all the data tested. The best individual was a network with 10 neurons on each layer and we can see also that each individual had 2 children.

From Figure 9 it is clear that from the 150 epoch the MSE is minimized. The MSE of the network when stopped was of 1433.097 which is a comparatively good value if compared with the other networks. This value varies with the size of the dataset. It is also clear that the validation, training and testing performance is the same. This is the most important factor of the figure. The same behaviour on those three components means that the network behaves well when it is presented with unknown values.

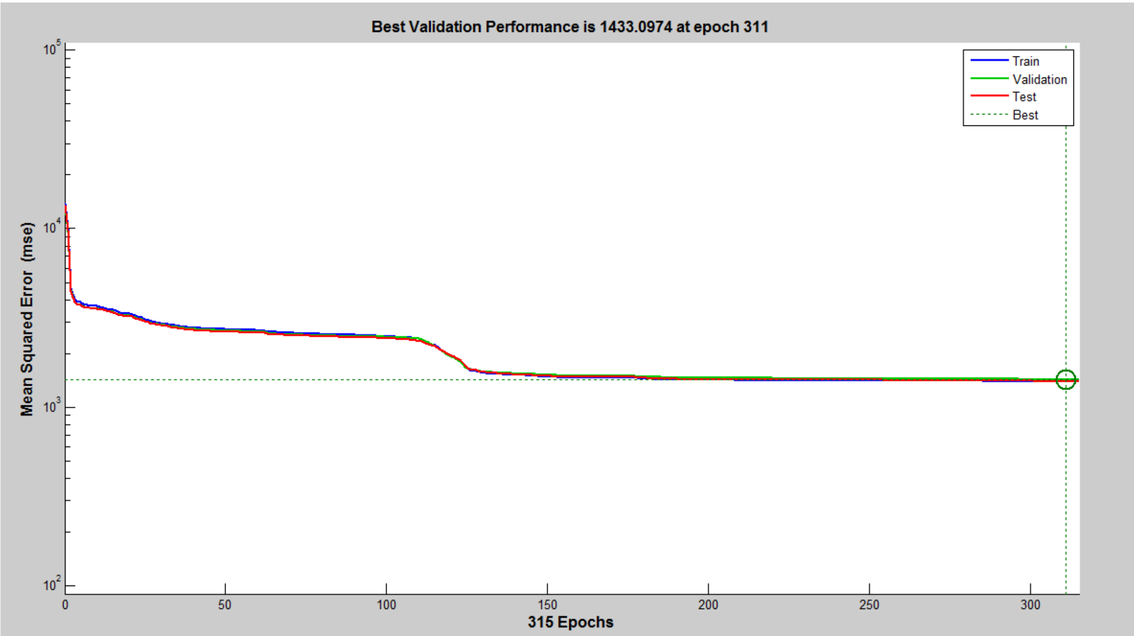

Figure 9 - Weather prediction 2 layer-Performance

Figure 10 represents the error histogram of the network. This graph represents that the majority of the values predicted had close to zero error and the error distribution is close to symmetrical.

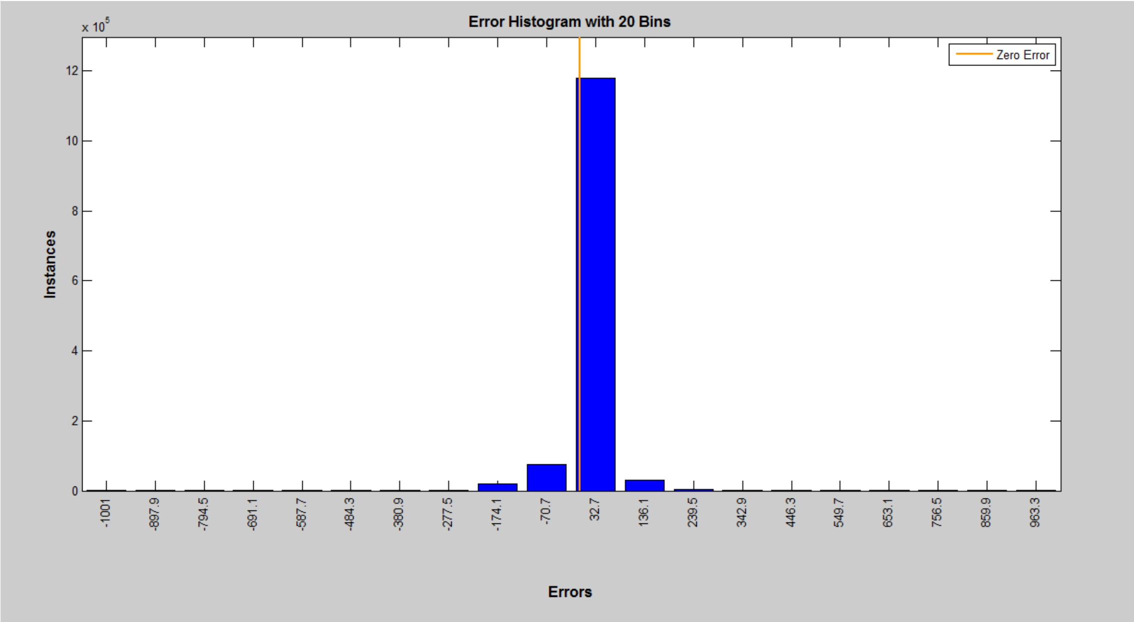

Figure 10 - Weather prediction 2 layers- error histogram

Figure 11 characterizes the training state of the network up to the moment of stop. The gradient was still significant which means that the values of the network were not stable by then. The variance (mu) was stable by the convergence time and on the last part of the figure we can see that the network had very few validation fails.

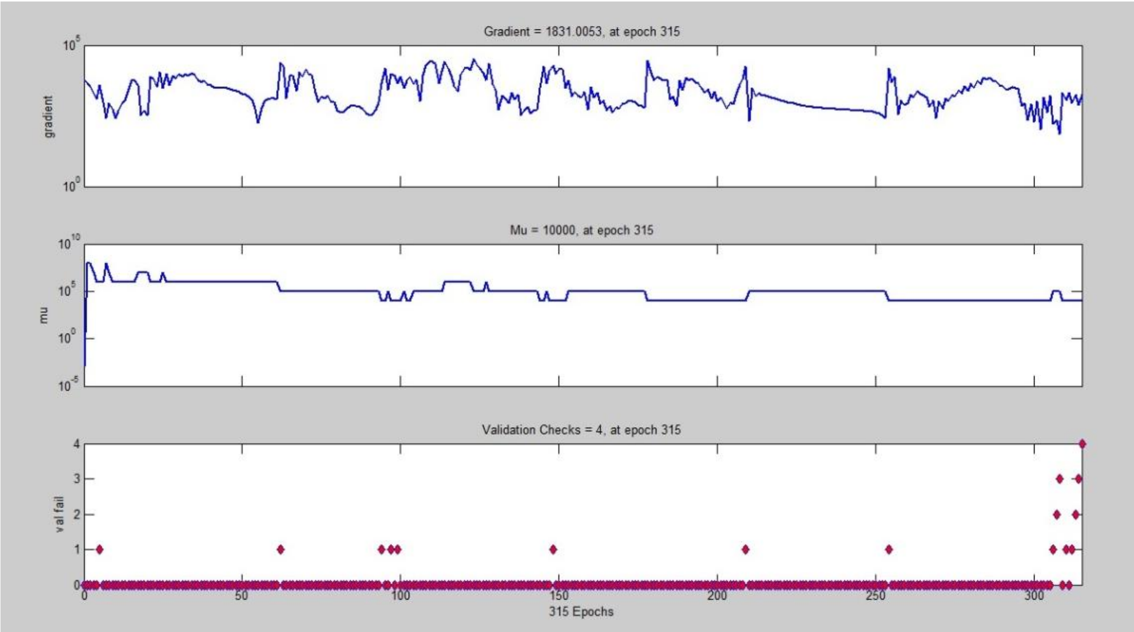

Figure 11 - Weather prediction 2 layers-Train State

Figure 12 represents the most important graph of the network training since it represents the actual accuracy and precision of the simulation. The concentration of the data on the extremes of the graph is due to the fact that pressure (Pa) and temperatures (ºC) have different orders of magnitude. The graph also shows that the algorithm predicted the output following the formula.

$$Output = 0.99 \times Target + 2.1 \quad (8)$$

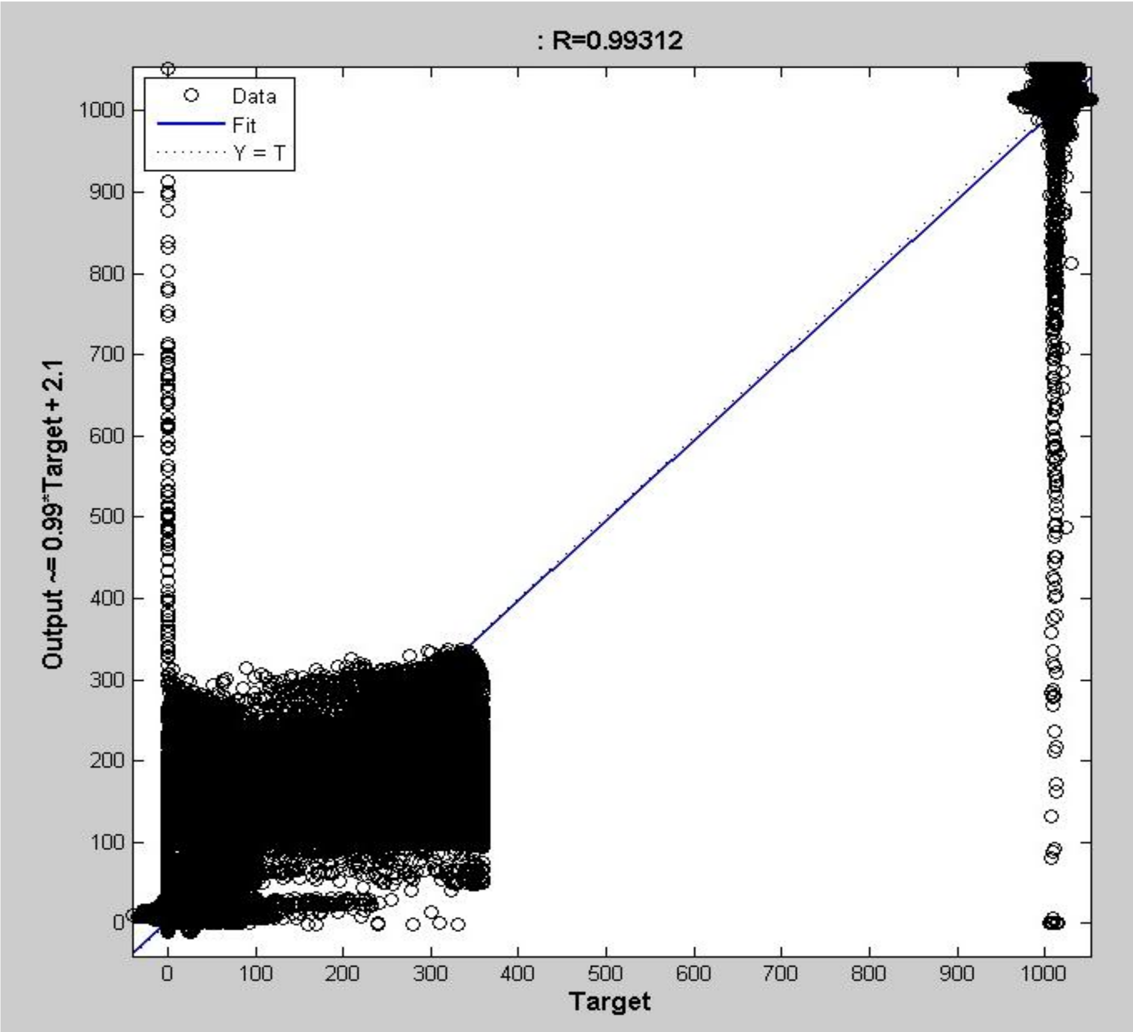

Figure 12-Weahter prediction 2 layers- Regression

4.2.2 Weather prediction- 5 layers

The 5 layers configuration took more time to obtain reasonable results. Networks with more layers have more biases and weights to be calculated and for that reason each network simulation took approximately 20 s more to be calculated. Multiplying this value by the number of simulations calculated it is clear that the processing time is much bigger. Since the processing time was one of the defined stopping criteria, the network had fewer epochs and fewer generations. Its best performance value obtained was of 99.029%.

The performance of the network shown on Figure 13 is very similar to the one on the two layer network.

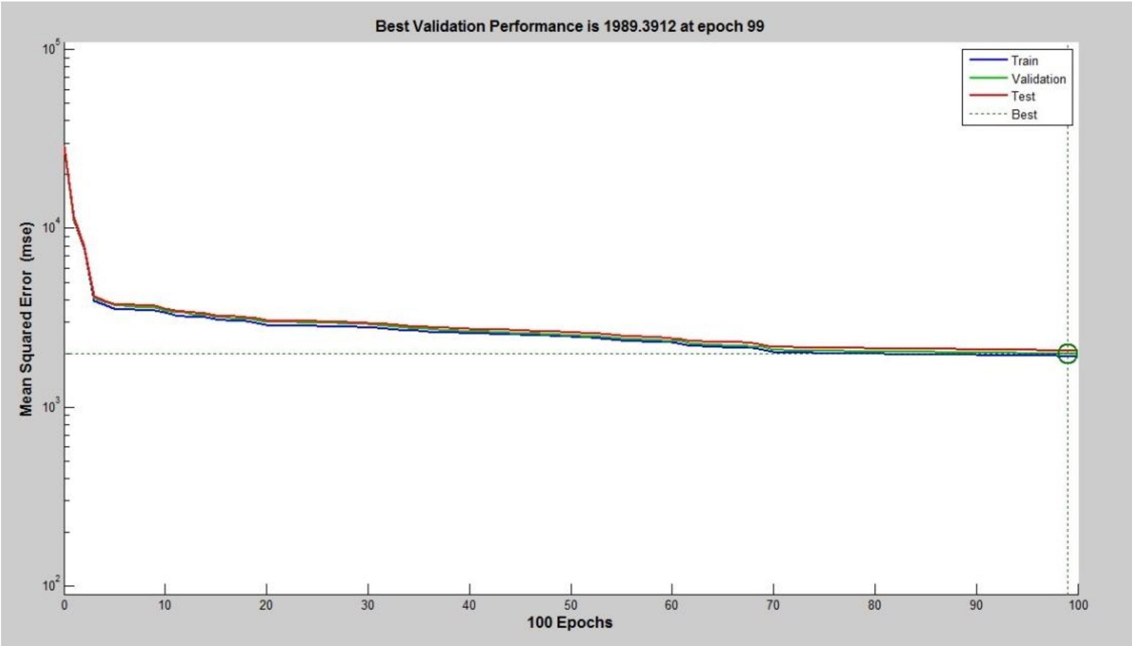

Figure 13 - Weather prediction 5 layers-Performance

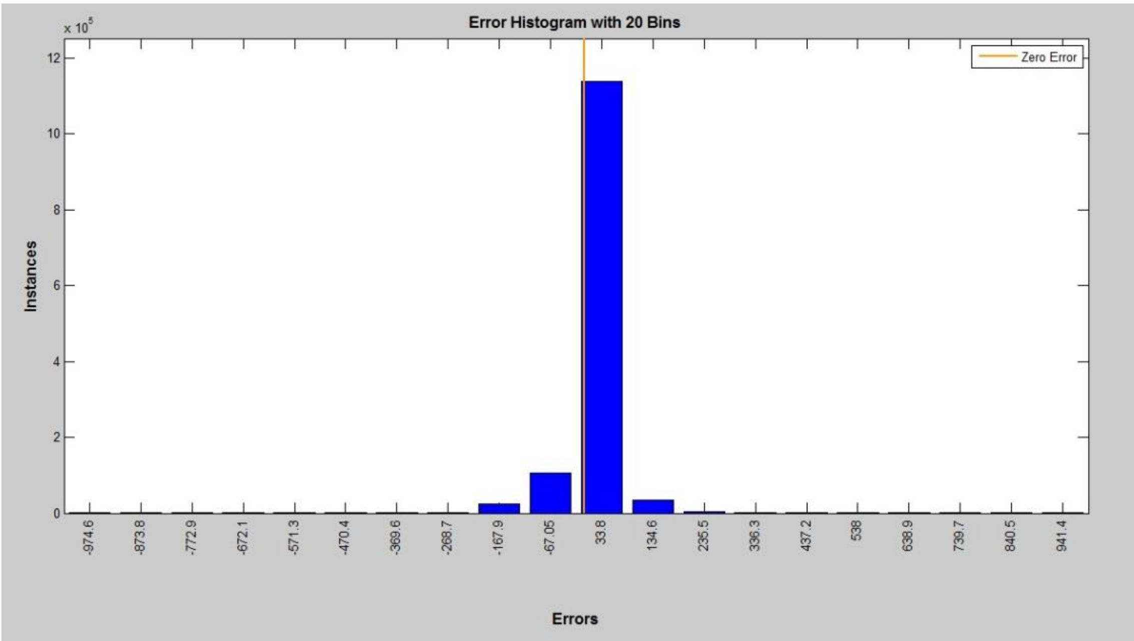

Figure 14 - Weather prediction 5 layers-Error Histogram

On Figure 15 we can see that the gradient is not as stable as the one on the 2 layers network. It is visible that the network had fewer validation fails.

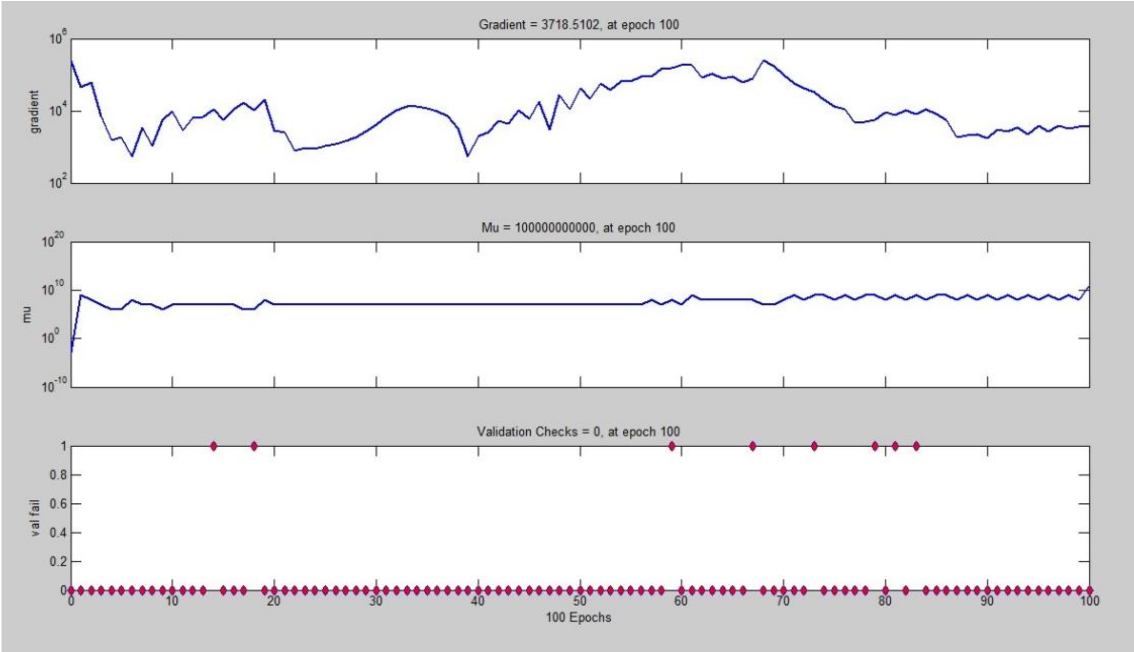

Figure 15 - Weather prediction 5 layers-Train State

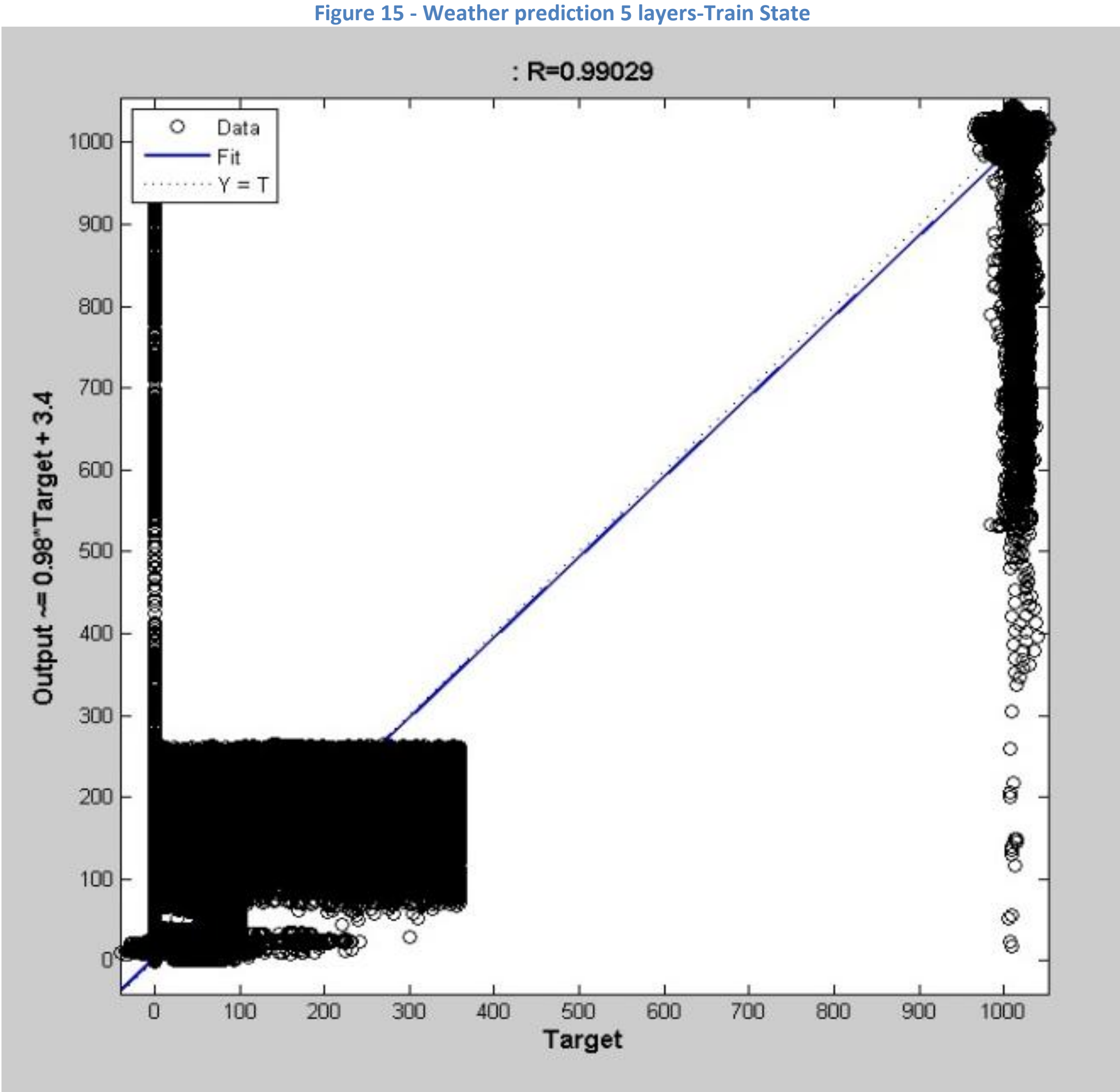

Figure 16 - Weather prediction 5 layers-Regression

Concerning the regression the values are scattered for the same reason as on the 2 layer network. The output relation with the target is:

$$Output = 0.98 \times Target + 3.4 \quad (9)$$

Comparing the two networks, we can see that the 2 layers network has a better performance and a more uniform validation, test and training performance. The network is also faster, less hardware intensive and easier to change due to its reduced number of variables. Allowing the 5 layer network to have more time to train would render a more stable behaviour but a slower processing network. Because speed is essential, mainly when paired with performance, the 2 layers network was chosen as the weather prediction network for the final algorithm. For more accurate results a more complex network would have to be trained and validated.

Due to its current performance, the network was deemed precise enough for its purpose. The training parameters and knowledge acquired during the training process were applied to the fuel optimization network.

4.3 Fuel Prediction

4.3.1 Introduction

The data for training this network was simulated after the weather prediction network was trained. That means that the predicted weather for the flights in TAP database was predicted using the weather prediction software with the best performance. The fuel prediction dataset has a bigger variance with respect to its values. Different flights on different days tend to have very different fuel consumption figures. For that reason, the overall performance expected for this network was lower than the one for the weather prediction network. To avoid this, the maximum number of failed validations was set to 5, the minimum gradient was set to $1e^{-10}$ and the minimum variance was set to 0.001. These tighter tolerances would allow the network to be more robust and have better performance. The training size parameter was also set to 70 % of the dataset.

Firstly a matrix assembler was created to assemble the extra data available and the data initially available from TAP. This assembler matrix task was to assemble a fuel prediction input matrix comprising the flight date, the plane registration, the origin and its corresponding weather variables, the destination and its corresponding weather variables, the plane model, the number of available seats, range, wingspan, length, height, cruising speed, fuel capacity, and the airway distance, great circle and air distance from the origin to the destination. The parameters to be predicted were the trip time, the trip fuel, the contingency fuel, the alternative fuel, the fuel reserve, ETOPS, holding distance fuel, additional fuel, take-off fuel, taxi fuel, fuel on board, actual fuel on board, fuel remaining, zero fuel weight and actual zero fuel weight. Although some of these parameters are a sum of others their prediction was made to allow the operations engineers to have a more complete set of values. Although some of the values predicted may end up being different from the sum of the parameters that comprise them, the difference should not be significant.

4.3.2 Fuel prediction- 2 layers

This network was the simplest and the fastest to train. Due to the more strict converging parameters the network converged on its first generation. For that reason the genetic evolution is not presented.

Figure 17 it is clear that from the 6 epoch the mean square error is minimized. The behaviour of the network on training, validation and testing is very similar. That shows the uniformity of the network response to different inputs.

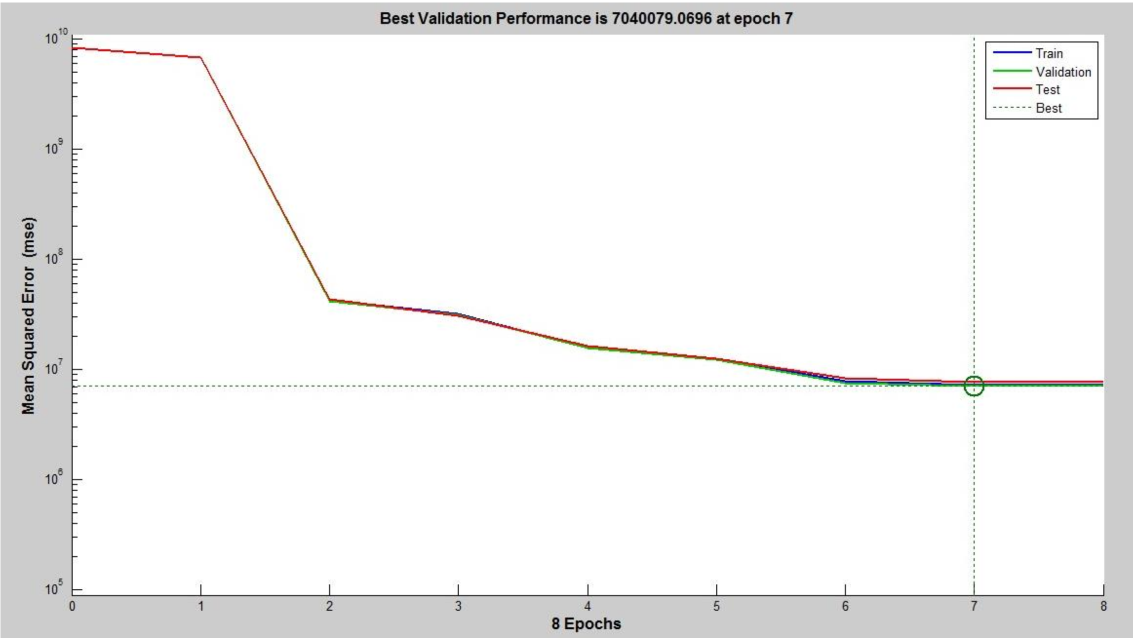

Figure 17 - Fuel prediction 2 layers-Performance

On Figure 18 we can see that the majority of the calculated error is close to 1183. This value is not as big as it might seem since a simple flight from Lisbon to Oporto consumes close to 4600 kg of fuel and error that big are more frequent in longer flights.

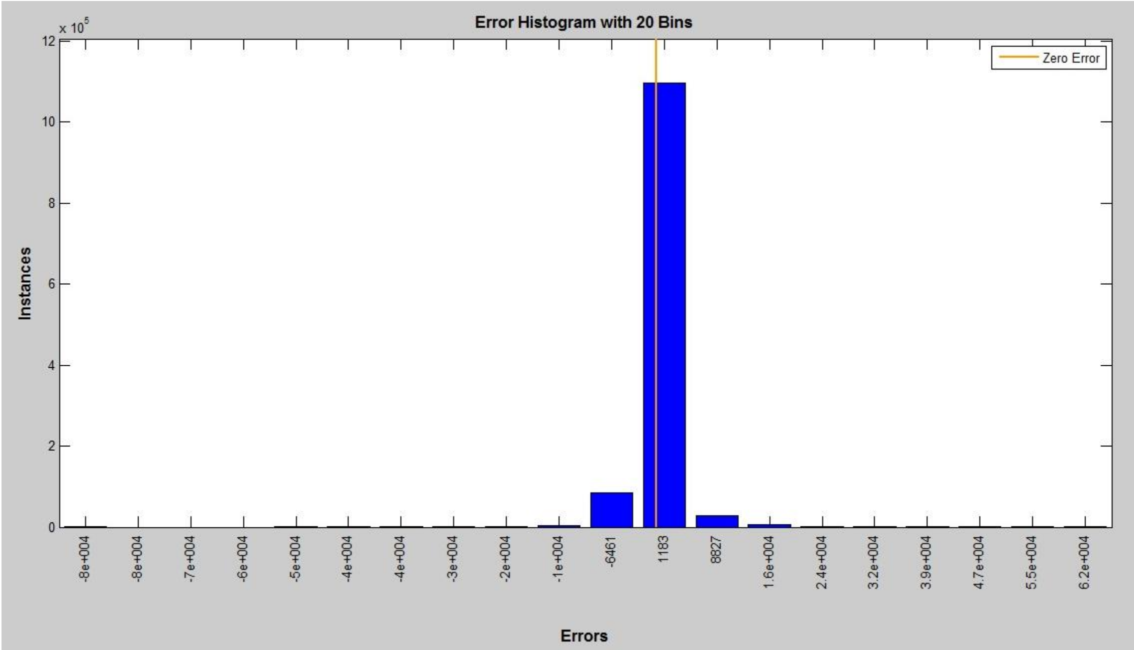

Figure 18 - Fuel prediction 2 layers- Error Histogram

The training state shown on Figure 19 has a gradient that approaches 10^8 in the end. The training gradient should become very small as the performance value becomes smaller. On this network the stopping criteria was the maximum variance. Taking into account the size of the dataset, the value of performance is supposed to be very big.

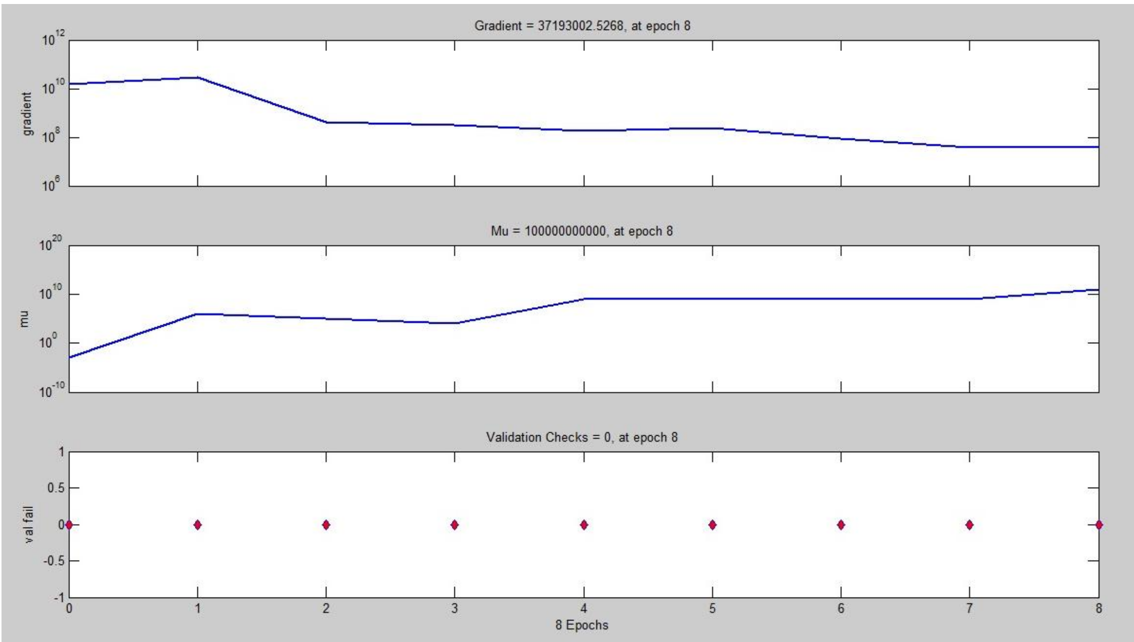

Figure 19-Fuel prediction 2 layers-Train State

The regression plot present on Figure 20 shows a network with an R=0.99449. Several values are from the middle line. That shows that some values are not very well predicted. Due to the vast range of values available this was expected.

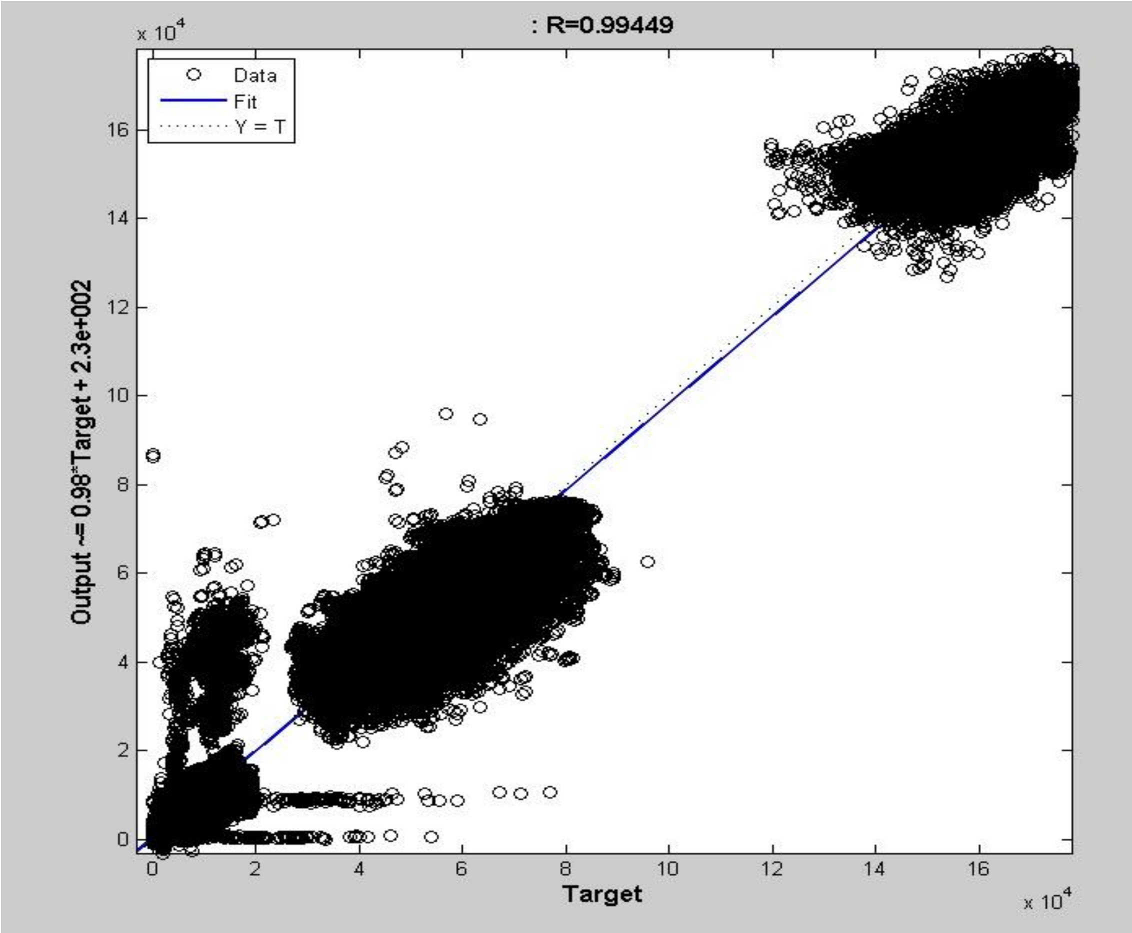

Figure 20 - Fuel prediction 2 layers-Regression

The relation between the output and the target is:

$$Output \cong 0.98 \times Target + 2.3 \times 10^2 \quad (10)$$

4.3.3 Fuel prediction- 5 layers

The data for training this network was calculated after the weather prediction network was calculated. That means that the predicted weather for the flights on TAP database was predicted using the weather prediction software with the best performance.

The performance on Figure 21 has a minimum on the eleventh epoch.

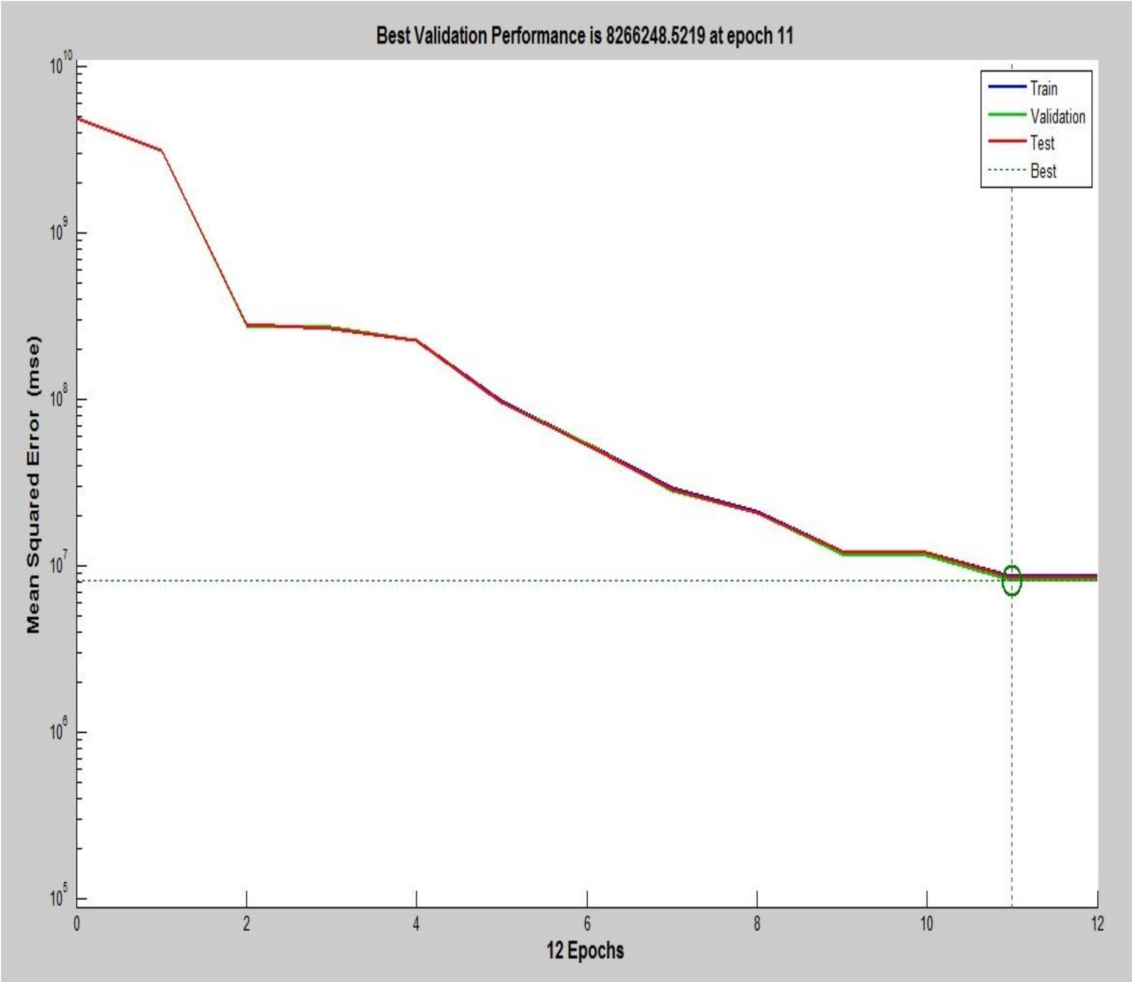

Figure 21 - Fuel prediction 5 layers-Performance

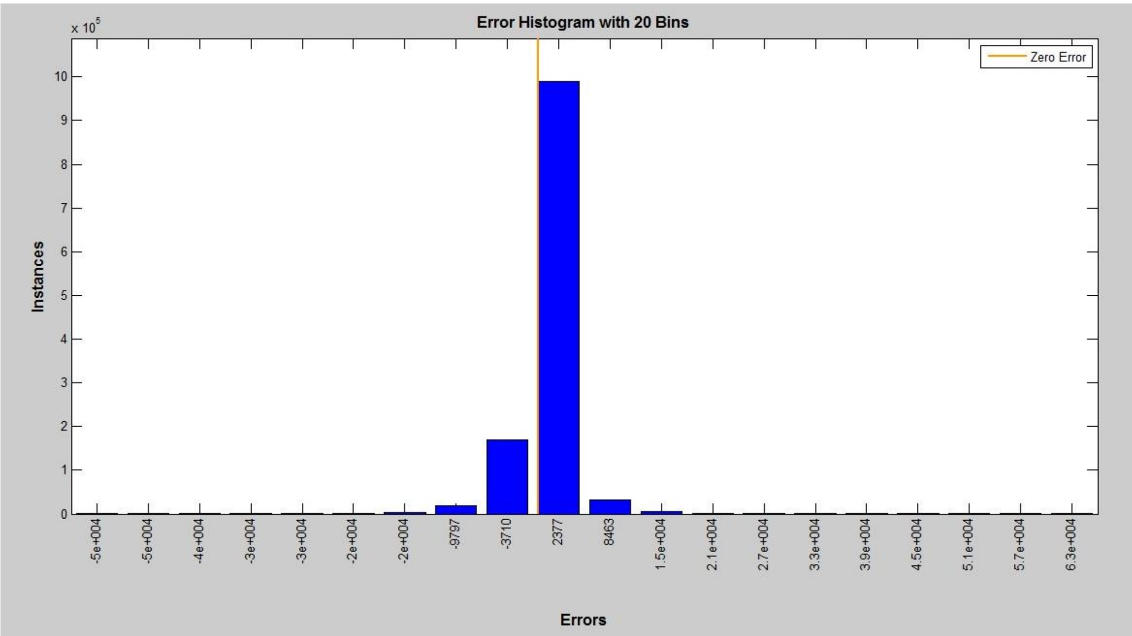

Figure 22 - Fuel prediction 5 layers-Error Histogram

Concerning the error on this network we have more instances with a bigger error than on the 2 layers network.

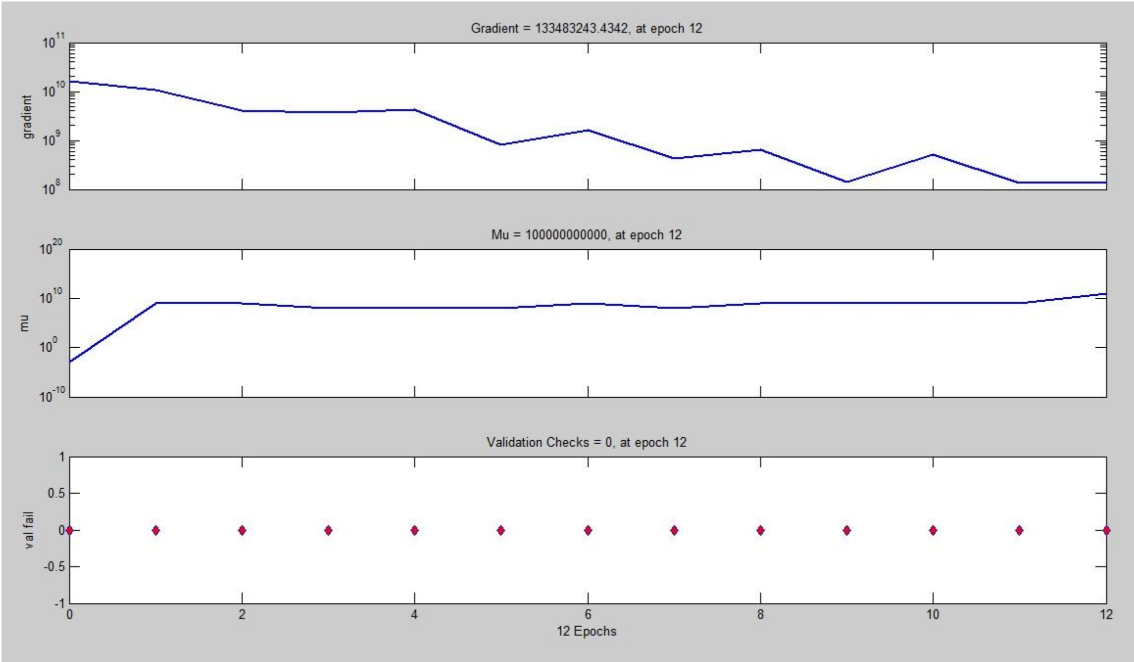

Figure 23-Fuel prediction 5 layers-Train State

The training state is similar to the one on the 2 layers network, but with a more accentuated gradient convergence.

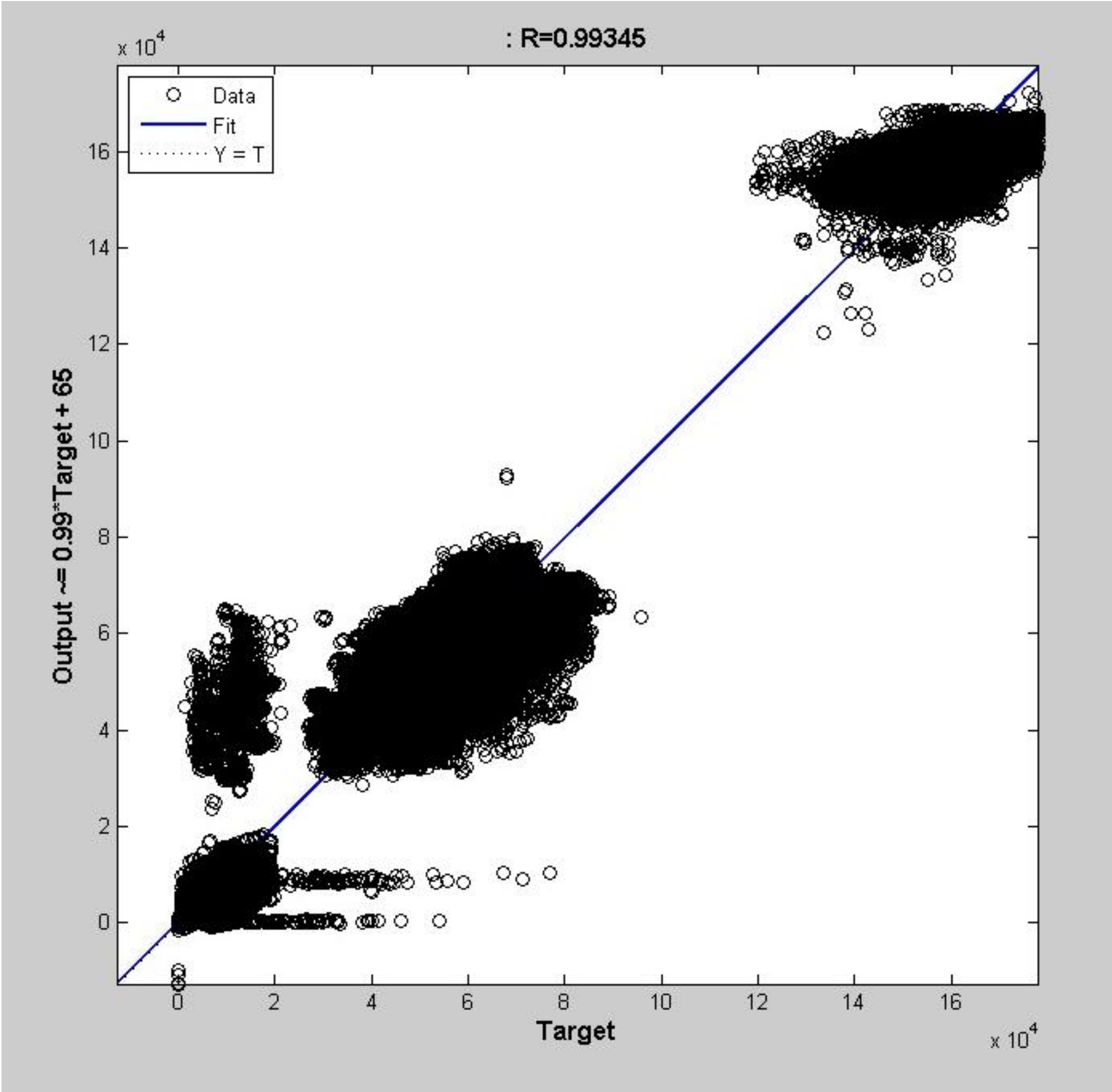

Figure 24-Fuel prediction 5 layers-Regression

The regression on the 5 layer network is very similar to the one on the 2 layers network but with a lower R value. The output is predicted as:

$$Output \cong 0.99 \times target + 65 \qquad (11)$$

Due to its more linear behaviour, a bigger set of zero lag instances and although with bigger error that occurred approximately 1×10^5 less frequently than on the 2 layers network, the five layers network was chosen as the one to be used on the final algorithm.

4.4. Final optimization

4.4.1 Introduction

After the training and the algorithm internal optimization, the algorithm is asked to act as if it were in control in 2010, the year of the data. The objective of the final optimization is to study the impact that this algorithm would have if it was being used during that year. The software doesn't know which planes were used and what was the actual consumption. The simulation time was also recorded to validate the applicability of the algorithm in real world conditions. Additionally the software projects the carbon footprint of the predicted flights. The CO_2 emissions are calculated by the simple rule that states that:

$$CO_2\ produced = 3.15 \times Mass\ of\ fuel\ burned \qquad (12)$$

The factor 3.15 is the one used for Jet A and was taken from the SESAR (Single European Sky ATM Research Programme) and TAP database. Due to the massive amount of data processed and the fact that, if the algorithm was to predict the fuel consumption for the entire year it would have to process an optimization with 71000 variables, it was decided to process for validation purposes a day at a time. This process is similar to the one used in programming flights. The simulation took 20 hours to find the best solution.

4.4.2 Weather Prediction Proof of concept

A weather prediction proof of concept had to exist to justify the existence of the weather prediction neural network. This network's sole purpose was to improve the fuel prediction software precision, by giving it more data and information about the state of the weather. This algorithm tries in no way to act as a substitute to the weather agencies forecasts. In real live conditions it would only be used for weather forecasts if no data was available, either due to the fact that the weather state at study is in a too far future or the data available is not complete. As said in the introduction, this approach has been used for several years by weather prediction agencies to have a more complete and accurate weather forecast. The algorithm developed only predicts 8 weather parameters that were thought to be important to the fuel prediction algorithm.

This proof of concept exists therefore to justify the performance demonstrated by the neural algorithms trained. The network used was the one with two layers configuration. As explained before, this is not the most accurate neural network trained but it is the fastest one.

For this proof of concept, the neural network had to predict the 8 components of the weather from de 1st of January 2011 to the 30th of September (273 days) of the same year for Lisbon, Portugal. The results were then compared to the data found in [30]. The difference between the two values, the real and the predicted was calculated. The difference was then averaged and the maximum difference detected.

The results are the following:

	Average Difference	Max. Difference
Max. Temperature (ºC)	-0.892511	12.10
Mean. Temperature (ºC)	-0.233768	9.98
Min. Temperature (ºC)	-0.211666	8.21
Mean Humidity (%)	-4.558275	25.34
Mean Sea Level Pressure (hPa)	1.596597	18.49
Wind Direction (Degrees)	0.019659	187.00
Max. Wind (Km/h)	0.677381	32.67
Max. Gust (Km/h)	0.646482	68.51

Table 2 - Weather Proof of Concept

From the table above it is clear that during the 273 days of the proof of concept, this algorithm has had, in average, a very good performance. The algorithm proved to be very precise in the prediction of the minimum temperature, wind direction, maximum wind and gust. Due to the stability of the sea level pressure in Lisbon its accuracy in predicting this parameter was high. The software had more difficulty in predicting the maximum temperature and humidity. The very high maximum differences occurred in days where either it was raining or very windy. Those maximum differences serve to prove that the state of the weather is very difficult to predict and has a 100% success rate. Taking into consideration the need to create a very simple neural network to predict the weather due to performance issues the network behaved better than initially predicted and can be easily improved. A more complex and precise network could not be developed due to the associated degradation in speed. The simplest the network, the faster it is. Since the network is called hundreds of times during the fuel prediction estimation, this configuration was considered as the most suitable. The complete results are presented in Annex C.

4.4.3 No Payload Proof of Concept

A proof of concept was made to explain briefly the simulations results for a random day. The day chosen was the first of August. The day in question belongs to the IATA's summer and contained 219 flights. The algorithm took 7 hours to reach an optimum solution. A value of available fuel on board of 1000 times the real value occurs when the solution proposed by the genetic algorithm is not possible, either because the plane selected is not available, or because the range of plane selected is inferior to the distance between origin and destination. All the choices made by the algorithm passed the conditions set.

A thorough analysis of the results yielded the following results.

Model	Number of times changed from
A319	18
A320	59
A321	16
A330	1
A340	5

Table 3-Planes changed no Payload

Concerning the changes from A319, the majority of the changes, 17, were to an A320. On the A320 fleet 58 flights were selected to be done by A319 planes. This change shows a try from the algorithm to attribute the biggest amount of flights possible to the A319 which is understandable due to the high efficiency of the plane. The same happens to the A321 fleet where 15 of the changes were to an A319. All the suggested changes from an A340 were to an A330. Those changes were to flights where the range of the A330 was enough and since the smaller A330 is more efficient it was selected.

The choices made by the algorithm were to select a more efficient plane when possible. For that reason the algorithm on 209 occasions selected a different plane than the original one, mostly choosing a smaller plane than the one used. That way we can conclude that the more efficient plane selection, without seats restrictions, was made by the software hence maximizing the use of the smaller planes. The parameters selected to give the best results on this proof of concept were applied to simulations of the months at study. Allowing the algorithm more time to converge would improve the results obtained. For that reason, and because the definition of the maximum generation when increased can damage the convergence due to the nature of the mutation algorithm it was set to 1000 while the maximum time was an input from the user with a default of 72000 seconds. An overall fuel saving off 4.87% was achieved for the flights that did obey the limitations. This value for that day represents 110312 kilos of less fuel consumed. The savings were mainly obtained using a tighter fleet scheduling and a smaller value of available fuel on board.

It is important to notice that in only 101 flights, 46%, the algorithm was able to save fuel. In the other flights the fuel consumption on the algorithm was bigger than the one in real live. This proves that the algorithm was thriving for an overall fleet consumption reduction and that it was able to decide in which flights fuel should be saved in order to minimize the fuel consumption. The algorithm managed that, using the most fuel efficient planes on the flights where the savings would be more significant.
The complete table of changes is included in annex A.

4.4.4 Payload Proof of Concept

This proof of concept follows the same format as the previous one. This proof was created to validate the payload constraints. The constraints were applied following the payload calculation according to Equation 13.

$$Payload\ carried = AZFW - DOW \tag{13}$$

Where AZFW is the actual zero fuel weight and DOW is the dry operating weight. The value of the payload is then compared to the capacity of the plane which is calculated using Equation14 that considers that each passenger adds 85 kilos to the plane:

$$Plane\ Payload = 85 \times Seats + Cargo\ Payload \tag{14}$$

In this proof of concept the goal is to demonstrate that the algorithm can still save fuel when it considers the payload carried and selects the plane capable of carrying it. This meant that routes that have normally a high payload would have to be done using the bigger planes. With this proof of concept the algorithm always makes the best decision and never chooses an A319 for a flight for which only an A321 can carry the payload. The cargo payload and the DOW where taken from TAP database.

Due to time constraints and the fact that the new conditions made the algorithm slower than on the first proof of concept, instead of studying all the flights that occurred on the 1st of July 2011 it was decided to study the first 100 flights of that day. The algorithm, as previously said , satisfied all the conditions.

A thorough analysis of the results yielded the following results.

Model	Number of times changed from
A319	33
A320	7
A321	7
A330	0
A340	1

Table 4 - Planes changed with Payload

Concerning the changes from A319, the majority of the changes, 30, were to an A320. On the A320 fleet 6 flights were selected to be done by A319 planes, a much smaller value of changes when compared to the previous proof of concept, proving that many flights that were changed previously from an A320 to an A319 would not have been possible if the payload normally carried was taken into account. The same happens with the A321 fleet where 4 of the changes were to an A320 which is a smaller and more efficient plane. The algorithm did not change the flights originally flown with the A330 to another plane. This is due to the fact that the A330 is the airplane with the biggest cargo capacity and that the flights where it operates have normally the biggest payloads. All the suggested changes from an A340 were to an A330. Those changes were to flights where the range of the A330 was enough and since the smaller A330 is more efficient it was selected.

The algorithm on 97 occasions changed the actual plane used. The savings were achieved, mainly, by lowering the AFOB or selecting a smaller plane where possible. As previously the algorithm only managed to save fuel on 38 flights. This result proves that the algorithm tries to minimize the overall fleet consumption. The overall fuel saving for the 100 flights studied was of 15396 kg of fuel, representing a saving of 1.43 %. The saving was smaller on this proof of concept, firstly due to the size of the data studied and due to the cargo limitations that did not allow more flights to be operated by more fuel efficient planes. The complete table of changes is included in the annexes.

5 Conclusions

5.4 Conclusions

The primary goal of building a powerful fuel prediction and optimization algorithm was achieved as proved by the proof of concept. To achieve that goal a small weather prediction neural network was created as was a complete fuel prediction neural network. To achieve good results on a short time frame, an evolutionary shape shifting neural network was created. The implementation of such neural network proved to be more challenging than initially predicted due to the lack of direct support in MATLAB© to genetic algorithm integer evaluations. The same problem demanded the creation of custom mutation and population algorithms. Due to the complexity of flight planning, several conditions were implemented on the final software to assure that the suggested flights obey all the limitations in existence. Custom genetic algorithms had to be created to allow the evaluation of a neural network dependent function. Due to the nature of optimization processes hundreds of possible combinations were tested to achieve the best results in the smallest possible amount of time. The variables were mainly the initial population size and mutation and crossover parameters. The proof of concept proved that, if the predicted payload was not taken into account, the fleet overall consumption could be up to 110313 kg of fuel a day. That saving represents on Jet A prices on September the 30th 2011 according to IATA a saving of $109321 US dollars a day. That saving represents a yearly saving of almost $ 40 million dollars. It is worth noticing that this saving was only achieved not considering the payload and would only be valid if the cargo transported by the airliner was defined by the software and not by previous contracts or the payload prediction. Implementing the payload limitations and taking into account the several dry overall weight of each airplane for each particular trip it was possible to simulate the new flight planning. On this configuration the algorithm was able to achieve a saving of 15396 kg of fuel for 100 flights representing almost $10 million dollars a year. That saving represents a 1.439 % saving for the population studied, hence proving the capabilities of the algorithm to optimize the fuel consumption. The differences between the two savings is mainly due to the difference in flight population size and the fact that the majority of the flights that in section 4.4.3 were operated by A319 could not be operated in section 4.4.4 due to the payload. The software proved that, even in very complex systems (more than 1000 variables), as fuel prediction, on an airline with such rich product offering as TAP, a solution is possible. The software can be used and improved in order to determine, not only the maintenance schedule of the company, as it can detect anomalies in the system and planes even before they become too evident. The results show that if applied in practice and taken into consideration the limitations of the software, it can be used safely as a guiding tool in flight scheduling and a powerful fuel prediction tool. Although not the main objective of this thesis, the fuel prediction software has the potential to allow airlines to predict with very little error the fuel consumption of their fleet without any need of calculation or assumptions. The software can also allow a better fleet management and a cheaper way of testing which plane is better for a given flight.

The algorithm presents the user with a solid user interface and delivers several fuel parcels, as well as estimated flight duration, plane used an expected payloads. The software can still be improved, not only speed wise as in real live limitations wise. Fully functional standalone software is to be created when the software reaches the maturity that can only be achieved after putting it in real live operation.

The main purpose of building an innovative way of predicting the fuel consumption was achieved. For that purpose intelligent systems had to be created and real live limitations simulated. The tool used to predict the weather conditions, although simple and limited, achieved better than expected results. Due to its modular foundation, updates can be made easily.

5.5 Future Work

Due to its modular conception the algorithm can be easily upgraded. For that reason adding extra features such as automatic flight planning document generator could be added in the future. A complete and more complex weather predicting algorithm can also be projected and a decision engine to determine the best period to buy the year´s requested fuel can also be projected. Speed improvements can be implemented to the software to allow the user more possible outputs.

6 References

1.O.Karray, Fakhreddine and De Silva, Clarence. *Soft Computing And Intelligent System Design.* Essex : Pearson Education Limited, 2004. 0 321 116178.

2. **Aircraft Fuel Reduction Initiatives**. *Transport Canada.* [Online] [Cited: 01 03 2011.] http://www.tc.gc.ca/eng/programs/environment-ecofreight-air-airtransat-96.htm.

3. Weather Forecast & Reports - Long Range & Local | Wunderground : Weather Underground. *Weather Underground.* [Online] NETWORK SOLUTIONS, LLC., 30 03 1995. [Cited: 01 03 2011.] http://www.wunderground.com/.

4. **Abraham, Ajith.** *Optimization of Evolutionary Neural Networks Using Hybrid Learning Algorithms.* Clayton, Victoria 3800, Australia, : School of Business Systems, Monash University,.

5. **Hochman, Robert, M.Khosgoftaar, Taghi and B.Allen, Edward.** *Evolutionary Neural Networks: A robust Approach to Software Reliability Problems.* Boca Raton : Department of Computer Science and Engineering Forida Atlantic Univesity.

6. **Nolfi, Stefano and Parisi, Domenico.** *Evolution of Artificial Neural Networks.* Rome : Institute of Psychology National Research Council.

7. **Miikkulainen, Risto.** *Neuroevolution.* Texas : The University of Texas at Austin.

8. **Stanley, Kenneth O. and Miikkulainen, Risto.** *Efficient Reinforcement Learning through Evolving Neural Network Topologies.* Texas : Department of Computer Sciences University of Texas at Austin.

9. —. Evolving Neural Networks through Augmenting Topologies. *The MIT Press Journals.* Vol. 10, 2.

10. **Owen, Bethan.** *Fuel Efficiency Development and Prediction. Main Thematic Area: Climate Change.* s.l. : Manchester Metropolitan University, 2008.

11. **Rypdal, Kristin.** *AIRCRAFT EMISSIONS.*

12. **1, Peeters P.M., J., Middel and A., Hoolhorst.** *Fuel efficiency of commercial aircraft . An overview of historical and future trends.* s.l. : National Aerospace Laboratory NLR , 2005.

13. **Chattopadhyay, Surajit and Chattopadhyay, Goutami.** *Identification of the best hidden layer size for three-layered neural net in predicting monsoon rainfall in India.* s.l. : IWA Publishing, 2008.

14. **Roskam, Jan and Edward Lan, Chuan-Tau.** *Airplane Aerodynamics and Performance.* s.l. : Design,Analysis and Research Corporation , 1997.

15. **Babikian, Raffi, P. Lukachko, Stephen and A. Waitz, Ian.** *The Historical Fuel Efficiency Characteristics of Regional Aircraft from Technological, Operational, and Cost Perspectives.* s.l. : Massachusetts Institute of Technology.

16. **Rojas, Raul.** *Neural Networks, A systematic introduction.* s.l. : Springer, 1996.

17. **Button, Kenneth and Taylor, Samantha.** International air transportation and economic development. *Journal of Air Transport Managment.* 6, 2000.

18. **A. Kalogirou, Soteris.** Optimization of solar systems using artificial neural-networks and genetic algorithms. *Applied Energy.* 77, 2004.

19. **Chow, T.T, et al., et al.** Global optimization of absorption chiller system by genetic algorithm and neural networks. *Energy and Buildings.* 34, 2001.

20. **Jin, Di and L. Kite-Powell, Hauke.** Optimal fleet utilization and replacement. *Transportation Research Part E.* 36 , 2000.

21. **Hall, Tony, E. BROOKS, Harold and A. DOSWELL III, Charles.** Precipitation Forecasting Using a Neural Network. *W E A T H E R A N D F O R E C A S T I N G.* 1998, Vol. 14.

22. **Paras, Sanjay Mathur, Avinash, Kumar and Mahesh, Chandra.** A Feature Based Neural Network Model for Weather Forecasting. *World Academy of Science, Engineering and Technology.* 34, 2007.

23. **Balakrishna, Poornima, Ganesan, Rajesh and Sherry, Lance.** Accuracy of reinforcement learning algorithms for predicting aircraft. *Transportation Research Part C.* 2010, 18.

24. **Hoon Oum, Tae and Yu, Chunyan.** A productiviy comparison of the world's major airlines. *Journal of Air Transport Management.* 1996, Vol. 2, 3/4.

25. **Tsaia, Wen-Hsien and Kuob, Lopin.** Operating costs and capacity in the airline industry. *Journal of Air Transport Management.* 2004, 10.

26. **Capri, Salvatore and Ignaccolo, Matteo.** Genetic algorithms for solving the aircraft-sequencing problem:. *Journal of Air Transport Management.* 2004, 10.

27. **D. Scheelhaase, Janina and G. Grimme, Wolfgang.** Emissions trading for international aviation—an estimation of the. *Journal of Air Transport Management.* 2007, 13.

28. **Miyoshi, C. and Mason, K.J.** The carbon emissions of selected airlines and aircraft types in three. *Journal of Air Transport Management.* 2009, 15.

29. **Givoni, Moshe and Rietveld, Piet.** The environmental implications of airlines' choice of aircraft size. *Journal of Air Transport Management.* 2010, 16.

30. **Abdelghanya, Khaled, Abdelghanyb, Ahmed and Raina, Sidhartha.** A model for the airlines' fuel management strategies. *Journal of Air Transport Management.* 2005, 11.

7 Annex

7.4 Annex A

This annex contains the data produced by the No Payload Proof of Concept.

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
DKR	**LIS**	319	5655	321	11980
DME	**LIS**	319	5206	320	12520
OXB	**LIS**	319	5177	319	9420
SID	**LIS**	319	7786	319	8480
RAI	**LIS**	319	5533	321	10620
GRU	**LIS**	330	39360	330	52900
NAT	**LIS**	330	39713	330	38000
HAM	**LIS**	319	5192	319	7020
MUC	**LIS**	319	6073	319	6480
FCO	**LIS**	319	5715	320	6880
FRA	**LIS**	319	5506	320	6700
BRU	**LIS**	319	6754	320	6300
AMS	**LIS**	319	6011	320	6740
ZRH	**LIS**	319	6130	321	6660
LIN	**LIS**	319	6140	320	6340
ORY	**LIS**	319	6488	319	5280
LHR	**LIS**	320	6318	319	5700
GVA	**LIS**	319	5997	319	5000
FNC	**LIS**	319	5138	319	3300
ORY	**OPO**	319	6144	319	4320
OPO	**LIS**	321	8776	319	1760
WAW	**LIS**	319	6263	320	9260
LGW	**OPO**	320	6456	320	4700
FAO	**LIS**	320	6079	320	1220
ORY	**LIS**	320	7333	319	5220
OPO	**ORY**	320	6082	320	4820
LIS	**OPO**	319	5656	319	1740
MAD	**LIS**	320	3884	319	2860
FNC	**OPO**	320	5271	319	4440
LIS	**ORY**	320	6279	319	5220
LIS	**FCO**	319	6028	320	6560
CPH	**LIS**	319	5486	319	7860
FNC	**LIS**	319	5193	319	3560
LIS	**LHR**	319	5757	320	5920
LAD	**LIS**	330	37127	340	44700
LIS	**FNC**	319	6156	319	3440
LIS	**HAM**	319	6125	319	6900
LIS	**BRU**	319	6630	320	5880

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
LIS	ORY	319	5701	321	6380
LIS	AMS	319	6489	320	6640
LIS	ZRH	319	4958	320	6000
LIS	GVA	319	6156	320	5360
LIS	MXP	319	6096	319	6000
LIS	MUC	319	5914	320	6760
LIS	BLQ	319	6224	319	6240
LHR	LIS	320	6279	320	5320
LIS	OPO	319	5692	319	1940
LIS	FRA	319	6219	321	8000
OPO	LIS	319	5874	319	1760
OPO	FNC	320	6485	319	4100
LIS	TER	319	5530	321	7000
LIS	BCN	319	6582	319	3960
LIS	ARN	319	6478	320	9800
LIS	FNC	319	6037	320	3700
LIS	MAD	319	5141	319	2600
LIS	OPO	319	5518	320	1980
LIS	CPH	319	6393	320	7880
LIS	PIX	319	3517	320	6360
LIS	OSL	319	5801	320	9760
LIS	ORY	319	5594	319	5620
LIS	GRU	330	49405	330	57080
LIS	GIG	330	44195	330	56980
LIS	FAO	319	6217	319	1480
OPO	GVA	320	6962	319	4460
LIS	CNF	330	49650	330	54860
ORY	LIS	319	6189	320	5360
LIS	BSB	330	49992	330	51740
OPO	MXP	319	5953	319	4900
LIS	LHR	319	5653	319	5860
LIS	BUD	319	6293	319	8620
OPO	ZRH	319	4868	320	5860
LIS	FCO	319	6150	319	6620
OPO	GRU	330	46605	330	53800
LIS	VCE	319	5351	319	6940
FNC	LIS	320	5972	319	3620
ORY	OPO	319	5931	319	4280
OPO	LGW	319	5903	320	4380
FCO	LIS	319	5474	320	6880
LIS	REC	330	39476	330	38500
GVA	LIS	320	7328	320	5040
LIS	SSA	330	41729	330	41600

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
ORY	**LIS**	321	8952	321	6160
FAO	**LIS**	319	5226	319	1320
BRU	**LIS**	319	5992	320	6660
MXP	**LIS**	320	7646	319	5440
ZRH	**LIS**	320	6273	320	5640
LHR	**LIS**	319	5290	320	5800
BLQ	**LIS**	319	5272	319	6540
MAD	**LIS**	320	3863	319	2380
LIS	**EWR**	330	36662	330	38840
AMS	**LIS**	319	5858	320	7320
FNC	**OPO**	320	5141	319	3940
LIS	**FNC**	319	5949	320	3560
BCN	**LIS**	319	5154	319	3540
FNC	**LGW**	319	5175	320	7900
HAM	**LIS**	320	5596	319	7080
MUC	**LIS**	319	5774	320	6600
LIS	**BRU**	319	6442	319	5860
LIS	**PRG**	319	5064	319	7360
OPO	**EWR**	319	5373	330	39200
LIS	**LAD**	330	34733	340	46020
FRA	**LIS**	319	5013	321	7760
TER	**LIS**	319	5920	321	6200
ORY	**LIS**	319	6298	319	4980
LIS	**LHR**	319	5655	319	5540
GVA	**OPO**	319	4931	319	4500
LIS	**ORY**	319	5555	320	5540
LIS	**TER**	319	5327	319	5520
MXP	**OPO**	319	6182	319	4500
ZRH	**OPO**	319	5173	320	5240
OPO	**LIS**	319	5844	319	1620
LHR	**LIS**	319	5318	319	5540
FCO	**LIS**	319	5370	319	6480
CPH	**LIS**	319	5234	320	9120
LGW	**FNC**	320	6295	320	8320
LIS	**FNC**	320	6406	320	3560
OSL	**LIS**	319	6341	320	9960
OPO	**FNC**	320	6560	319	3840
ARN	**LIS**	319	5719	320	10340
FNC	**LIS**	320	5880	320	3460
LIS	**FCO**	319	5826	319	6620
VCE	**LIS**	319	7326	319	6000
LIS	**FRA**	320	6573	320	6560
LIS	**AMS**	319	6456	319	6180

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
BUD	LIS	319	5114	319	7960
PIX	TER	320	8303	320	1100
LIS	ZRH	320	5071	320	6200
LIS	LHR	319	5674	321	6660
LIS	BRU	319	6474	320	5860
LIS	PIX	320	3667	320	6420
LIS	GVA	319	5919	319	5020
LIS	MXP	319	5781	320	6000
LIS	FOR	330	48630	330	38660
LIS	GIG	330	46581	330	49060
LIS	MAD	320	5507	319	2760
LIS	FNC	319	5901	319	3620
LIS	ORY	319	5438	321	6160
BRU	LIS	319	5866	319	6080
TER	LIS	319	5737	320	4800
LIS	LHR	319	5711	319	5000
LIS	FAO	320	6602	320	1520
LIS	OPO	319	5348	320	1540
LGW	OPO	319	5466	320	5040
PRG	LIS	319	4682	319	7380
LHR	LIS	319	5141	319	5560
ORY	LIS	319	6047	320	5980
FNC	LIS	319	4593	320	3660
LIS	HOR	320	5064	320	6820
OPO	LIS	319	5661	320	1680
OPO	ORY	319	5459	319	4620
TER	LIS	319	5703	319	4440
OPO	GVA	319	6027	319	4280
FNC	OPO	319	4401	319	4000
LIS	CPH	319	6184	319	7180
MAD	LIS	319	3476	319	2480
FAO	LIS	319	5076	320	1270
LIS	MAD	319	4844	319	2540
OPO	LIS	319	5565	320	1480
LIS	ORY	319	5247	320	4920
FNC	LIS	319	4549	319	3380
FCO	LIS	319	5584	319	6400
GVA	LIS	319	5118	319	5240
AMS	LIS	319	5498	319	6520
LIS	MPM	330	48142	340	70460
FRA	LIS	320	5814	320	6360
LIS	TER	320	5838	321	6820
LIS	HAM	320	6433	319	6320

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
ZRH	LIS	319	5504	320	5780
BRU	LIS	319	5727	320	6280
MXP	LIS	319	6085	320	5780
LHR	LIS	319	5230	321	6840
PIX	TER	319	8183	320	1200
FNC	LIS	319	4597	320	3380
ORY	LIS	319	6041	321	6900
LIS	LHR	319	5704	320	5580
LIS	FNC	319	5932	319	3620
LIS	ORY	319	5510	320	4920
LIS	ZRH	320	5112	319	5980
LIS	LIN	320	6073	320	5600
LIS	FCO	320	6301	320	6280
LIS	MUC	320	6173	319	6260
LIS	FRA	320	6568	320	6500
LIS	AMS	320	7012	320	6020
OPO	LGW	319	5679	320	4155
LHR	LIS	319	5225	319	5480
LIS	DME	319	6137	320	12680
ORY	OPO	319	5734	319	4860
LIS	BRU	319	6439	320	5700
OPO	ORY	319	5354	319	4400
MAD	LIS	319	3468	319	2520
LIS	GVA	319	5954	319	5560
LIS	LHR	319	5337	319	4880
LIS	OPO	319	5090	319	1540
TER	LIS	319	5686	320	4800
GVA	OPO	319	4675	319	4580
HOR	LIS	319	8175	320	5040
LIS	MAD	319	4687	319	2500
LIS	PXO	319	5290	319	3680
CNF	LIS	330	41429	330	50240
BSB	LIS	330	41194	330	47100
GRU	LIS	330	43442	330	56180
FNC	LIS	319	4549	319	3460
GIG	LIS	330	40780	330	51400
OPO	LIS	319	5384	319	1520
TER	LIS	319	5744	321	5520
LIS	FNC	319	5553	321	4620
LAD	LIS	330	38588	340	45140
LIS	DKR	319	6234	321	12260
LIS	SID	320	6815	320	9240
LIS	RAI	320	6626	320	10800

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
REC	**LIS**	330	40889	330	39860
LIS	**HEL**	321	7758	320	10210
LIS	**LAD**	330	33779	340	48880
LIS	**FAO**	319	5881	319	1540
OPO	**FNC**	319	5608	319	4100
GRU	**OPO**	330	43445	330	54100
LIS	**OPO**	319	5295	319	1400
PXO	**LIS**	319	5866	319	3460
FOR	**LIS**	330	43977	330	39160
Fuel Total		Predicted	2151683	Actual	2261995
Difference = 110313 = 4.87% less fuel consumed					

Table 5 – No Payload Proof of Concept

7.5 Annex B

This annex contains data produced by the Payload Proof of Concept.

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
DKR	LIS	319	5636	321	11980
DME	LIS	319	5254	320	12520
OXB	LIS	319	5201	319	9420
SID	LIS	330	7711	319	8480
RAI	LIS	319	5609	321	10620
GRU	LIS	330	39311	330	52900
NAT	LIS	330	39792	330	38000
HAM	LIS	320	5787	319	7020
MUC	LIS	320	7399	319	6480
FCO	LIS	320	6429	320	6880
FRA	LIS	320	6291	320	6700
BRU	LIS	320	7620	320	6300
AMS	LIS	320	7033	320	6740
ZRH	LIS	320	6683	321	6660
LIN	LIS	320	6910	320	6340
ORY	LIS	320	7087	319	5280
LHR	LIS	320	6336	319	5700
GVA	LIS	320	6557	319	5000
FNC	LIS	319	4756	319	3300
ORY	OPO	320	6739	319	4320
OPO	LIS	320	6592	319	1760
WAW	LIS	320	6812	320	9260
LGW	OPO	320	6420	320	4700
FAO	LIS	319	5238	320	1220
ORY	LIS	320	7132	319	5220
OPO	ORY	320	6100	320	4820
LIS	OPO	320	6052	319	1740
MAD	LIS	319	3498	319	2860
FNC	OPO	319	4625	319	4440
LIS	ORY	320	6042	319	5220
LIS	FCO	320	6926	320	6560
CPH	LIS	319	5554	319	7860
FNC	LIS	319	4861	319	3560
LIS	LHR	320	6323	320	5920
LAD	LIS	330	37076	340	44700
LIS	FNC	320	6618	319	3440
LIS	HAM	320	6537	319	6900
LIS	BRU	320	7250	320	5880
LIS	ORY	320	6110	321	6380
LIS	AMS	321	7786	320	6640

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
LIS	**ZRH**	320	5214	320	6000
LIS	**GVA**	320	6818	320	5360
LIS	**MXP**	320	6603	319	6000
LIS	**MUC**	320	6452	320	6760
LIS	**BLQ**	320	6686	319	6240
LHR	**LIS**	320	6261	320	5320
LIS	**OPO**	319	5519	319	1940
LIS	**FRA**	320	6880	321	8000
OPO	**LIS**	320	6341	319	1760
OPO	**FNC**	320	6447	319	4100
LIS	**TER**	320	5911	321	7000
LIS	**BCN**	320	7299	319	3960
LIS	**ARN**	320	7124	320	9800
LIS	**FNC**	320	6580	320	3700
LIS	**MAD**	320	5656	319	2600
LIS	**OPO**	320	5948	320	1980
LIS	**CPH**	320	6977	320	7880
LIS	**PIX**	319	3517	320	6360
LIS	**OSL**	319	5771	320	9760
LIS	**ORY**	320	6077	319	5620
LIS	**GRU**	330	49454	330	57080
LIS	**GIG**	330	44195	330	56980
LIS	**FAO**	320	6729	319	1480
OPO	**GVA**	320	6849	319	4460
LIS	**CNF**	330	49426	330	54860
ORY	**LIS**	320	6792	320	5360
LIS	**BSB**	330	49940	330	51740
OPO	**MXP**	320	6354	319	4900
LIS	**LHR**	320	6182	319	5860
LIS	**BUD**	320	7083	319	8620
OPO	**ZRH**	319	4772	320	5860
LIS	**FCO**	321	8884	319	6620
OPO	**GRU**	330	46605	330	53800
LIS	**VCE**	320	6269	319	6940
FNC	**LIS**	319	4886	319	3620
ORY	**OPO**	320	6486	319	4280
OPO	**LGW**	320	6305	320	4380
FCO	**LIS**	320	6529	320	6880
LIS	**REC**	330	39241	330	38500
GVA	**LIS**	320	6743	320	5040
LIS	**SSA**	330	41611	330	41600
ORY	**LIS**	330	6647	321	6160
FAO	**LIS**	319	5196	319	1320

Origin	Destination	Model	Predicted Fuel Consumption (kg)	Model (Real)	Actual Fuel Consumption (kg)
BRU	**LIS**	320	7380	320	6660
MXP	**LIS**	320	7592	319	5440
ZRH	**LIS**	320	6382	320	5640
LHR	**LIS**	320	5906	320	5800
BLQ	**LIS**	320	6153	319	6540
MAD	**LIS**	319	3575	319	2380
LIS	**EWR**	330	36413	330	38840
AMS	**LIS**	320	6721	320	7320
FNC	**OPO**	319	4289	319	3940
LIS	**FNC**	320	6490	320	3560
BCN	**LIS**	320	5637	319	3540
FNC	**LGW**	319	5068	320	7900
HAM	**LIS**	319	5088	319	7080
MUC	**LIS**	320	7090	320	6600
LIS	**BRU**	321	7403	319	5860
LIS	**PRG**	320	5525	319	7360
OPO	**EWR**	330	32801	330	39200
Fuel Total		Predicted	1054465	Actual	1069860
Difference =15396= 1.43% less fuel consumed					

Table 6 - Payload Proof of Concept

7.6 Annex C

This annex contains data produced by the Weather Proof of Concept.

Predicted								Actual							
Max Temperature °C	Mean Temperature °C	Min Temperature °C	Mean Humidity (%)	Mean Sea Level Pressure (hPa)	WindDir (Degrees)	Max Wind Speed (Km/h)	Max Gust Speed (Km/h)	Max Temperature °C	Mean Temperature °C	Min Temperature °C	Mean Humidity	Mean Sea Level Pressure (hPa)	WindDir (Degrees)	Max Wind Speed (Km/h)	Max Gust Speed (Km/h)
19,52	15,34	11,54	74,25	1014,49	183,23	25,44	13,91	16	13	12	89	1020	293	19	
19,86	15,69	11,89	74,66	1014,52	173,71	25,14	13,05	12	11	10	100	1021	46	8	
20,05	15,89	12,10	74,90	1014,54	168,76	24,99	12,58	13	11	10	99	1020	59	13	
20,16	16,00	12,21	75,04	1014,55	166,13	24,91	12,33	16	13	11	97	1017	176	19	
20,22	16,06	12,28	75,12	1014,55	164,74	24,87	12,19	17	16	15	97	1012	209	27	47
20,25	16,10	12,31	75,16	1014,56	164,02	24,84	12,12	18	16	14	94	1010	215	42	63
20,27	16,11	12,33	75,18	1014,56	163,69	24,83	12,09	18	17	15	83	1013	224	37	55
20,27	16,12	12,34	75,19	1014,56	163,57	24,83	12,07	16	13	12	86	1013	223	45	71
20,27	16,12	12,34	75,19	1014,56	163,59	24,83	12,07	15	12	10	78	1020	292	21	
20,27	16,12	12,34	75,18	1014,56	163,68	24,83	12,08	16	12	9	93	1023	223	23	
20,26	16,11	12,33	75,18	1014,56	163,82	24,84	12,09	18	14	12	89	1027	321	21	
20,26	16,11	12,32	75,17	1014,56	163,99	24,84	12,10	16	14	13	80	1030	351	16	
20,25	16,10	12,32	75,16	1014,56	164,19	24,85	12,12	12	10	8	98	1025	42	13	
20,24	16,09	12,31	75,15	1014,55	164,42	24,86	12,14	10	9	9	100	1024	52	10	
20,23	16,08	12,30	75,14	1014,55	164,65	24,86	12,16	11	10	9	100	1026	66	11	
20,22	16,07	12,29	75,13	1014,55	164,91	24,87	12,18	14	11	8	94	1027	87	13	
20,21	16,06	12,28	75,12	1014,55	165,18	24,88	12,21	16	13	11	90	1028	102	14	
20,20	16,05	12,27	75,10	1014,55	165,47	24,89	12,23	14	12	10	94	1024	49	14	
20,19	16,04	12,26	75,09	1014,55	165,77	24,90	12,26	14	11	9	92	1021	52	14	
20,18	16,03	12,24	75,08	1014,55	166,08	24,91	12,28	16	12	9	85	1019	57	19	
20,17	16,02	12,23	75,06	1014,55	166,41	24,92	12,31	14	10	7	66	1019	59	29	
20,16	16,00	12,22	75,05	1014,55	166,76	24,93	12,34	9	7	4	51	1016	53	35	55
20,14	15,99	12,20	75,03	1014,54	167,11	24,94	12,37	8	6	4	54	1014	48	32	48
20,13	15,98	12,19	75,01	1014,54	167,49	24,95	12,41	11	7	4	75	1013	52	26	
20,12	15,96	12,18	75,00	1014,54	167,87	24,96	12,44	12	9	7	67	1014	53	26	
20,10	15,95	12,16	74,98	1014,54	168,28	24,97	12,47	14	10	7	64	1005	35	26	
20,09	15,93	12,15	74,96	1014,54	168,69	24,99	12,51	13	9	6	87	1000	268	19	
20,07	15,92	12,13	74,94	1014,54	169,12	25,00	12,54	14	11	8	82	1005	275	34	
20,06	15,90	12,12	74,93	1014,54	169,57	25,01	12,58	12	9	7	79	1012	297	27	24
20,04	15,88	12,10	74,91	1014,53	170,02	25,03	12,62	12	8	5	79	1015	334	29	
20,03	15,87	12,08	74,89	1014,53	170,50	25,04	12,66	13	10	7	66	1020	351	29	24
19,33	15,18	11,41	74,09	1014,45	193,78	25,76	14,50	12	8	4	67	1027	15	19	
19,56	15,41	11,63	74,35	1014,48	185,35	25,50	13,84	14	9	5	65	1030	14	19	
19,70	15,54	11,77	74,51	1014,49	181,05	25,37	13,49	18	12	7	70	1031	356	24	
19,77	15,62	11,84	74,60	1014,50	178,91	25,30	13,30	17	12	8	76	1033	26	14	
19,80	15,65	11,87	74,64	1014,51	177,94	25,27	13,22	18	11	5	79	1032	47	10	
19,82	15,67	11,89	74,66	1014,51	177,62	25,26	13,18	18	11	5	75	1028	31	11	
19,82	15,67	11,89	74,66	1014,51	177,67	25,26	13,18	17	12	7	84	1025	231	13	
19,81	15,66	11,89	74,65	1014,51	177,94	25,27	13,20	15	11	8	93	1021	142	23	
19,80	15,65	11,88	74,64	1014,50	178,33	25,28	13,22	18	14	12	89	1019	146	24	
19,79	15,64	11,87	74,63	1014,50	178,80	25,30	13,26	17	13	10	85	1021	124	14	
19,78	15,63	11,85	74,62	1014,50	179,32	25,31	13,29	17	13	9	92	1019	170	32	
19,76	15,61	11,84	74,60	1014,50	179,87	25,33	13,33	16	13	10	89	1021	27	11	27

Predicted								Actual							
Max Temperature °C	Mean Temperature °C	Min Temperature °C	Mean Humidity (%)	Mean Sea Level Pressure (hPa)	WindDir (Degrees)	Max Wind Speed (Km/h)	Max Gust Speed (Km/h)	Max Temperature °C	Mean Temperature °C	Min Temperature °C	Mean Humidity	Mean Sea Level Pressure (hPa)	WindDir (Degrees)	Max Wind Speed (Km/h)	Max Gust Speed (Km/h)
19,75	15,60	11,83	74,59	1014,50	180,45	25,35	13,37	14	12	10	83	1014	261	27	47
19,73	15,59	11,82	74,57	1014,49	181,05	25,37	13,41	14	11	8	89	1015	231	35	58
19,72	15,57	11,80	74,55	1014,49	181,67	25,38	13,45	15	13	11	79	1007	267	37	52
19,70	15,56	11,79	74,54	1014,49	182,31	25,40	13,49	15	12	9	79	1001	260	48	82
19,69	15,55	11,78	74,52	1014,49	182,95	25,42	13,53	14	12	10	69	1011	297	47	71
19,67	15,53	11,77	74,51	1014,48	183,61	25,44	13,57	16	12	9	89	1017	201	26	
19,66	15,52	11,76	74,50	1014,48	184,29	25,46	13,61	17	14	12	96	1015	226	27	48
19,64	15,51	11,74	74,48	1014,48	184,97	25,49	13,66	17	13	11	87	1022	266	27	
19,63	15,49	11,73	74,47	1014,48	185,67	25,51	13,70	17	14	12	82	1028	295	26	
19,62	15,48	11,72	74,46	1014,48	186,38	25,53	13,74	16	13	12	80	1027	332	29	
19,60	15,47	11,72	74,45	1014,47	187,10	25,55	13,78	19	14	10	79	1024	358	26	
19,59	15,46	11,71	74,44	1014,47	187,83	25,57	13,82	21	14	9	74	1025	19	19	
19,58	15,45	11,70	74,43	1014,47	188,56	25,59	13,85	22	16	10	74	1027	343	35	
19,56	15,44	11,69	74,42	1014,46	189,31	25,62	13,89	19	14	11	82	1027	338	35	
19,55	15,43	11,69	74,41	1014,46	190,06	25,64	13,93	18	14	11	69	1026	347	37	
19,54	15,42	11,68	74,40	1014,46	190,82	25,66	13,96	16	12	9	67	1026	348	47	50
19,12	15,10	11,45	74,12	1014,37	216,46	26,47	15,31	15	11	8	59	1025	14	27	
19,25	15,19	11,52	74,20	1014,40	208,42	26,21	14,89	14	10	7	62	1023	16	27	
19,32	15,26	11,57	74,27	1014,41	204,41	26,09	14,64	15	11	7	62	1019	19	24	
19,37	15,30	11,61	74,31	1014,42	202,57	26,03	14,52	13	10	7	63	1014	52	19	
19,39	15,32	11,63	74,34	1014,42	201,91	26,01	14,45	15	11	7	82	1013	103	27	39
19,40	15,34	11,64	74,35	1014,42	201,92	26,01	14,43	17	13	9	85	1014	105	19	
19,40	15,34	11,66	74,37	1014,42	202,31	26,02	14,43	17	13	9	85	1013	106	23	
19,41	15,35	11,66	74,38	1014,42	202,91	26,04	14,43	14	12	11	91	1013	86	19	
19,40	15,35	11,67	74,39	1014,41	203,63	26,06	14,44	17	13	10	83	1016	59	23	
19,40	15,36	11,68	74,40	1014,41	204,41	26,08	14,46	18	14	11	79	1018	54	21	
19,40	15,36	11,69	74,41	1014,41	205,25	26,11	14,47	15	13	12	92	1010	46	14	
19,40	15,37	11,71	74,42	1014,41	206,10	26,13	14,49	16	13	11	85	1003	295	21	
19,40	15,38	11,72	74,43	1014,40	206,96	26,16	14,50	16	13	11	86	1002	142	21	
19,40	15,38	11,73	74,45	1014,40	207,84	26,19	14,51	13	11	10	88	999	334	27	
19,40	15,39	11,75	74,47	1014,40	208,71	26,21	14,52	15	11	8	68	1007	226	23	35
19,41	15,40	11,76	74,48	1014,39	209,59	26,24	14,52	16	11	7	76	1015	351	32	
19,41	15,42	11,78	74,50	1014,39	210,46	26,27	14,53	15	13	11	75	1023	337	34	
19,41	15,43	11,80	74,52	1014,39	211,33	26,30	14,53	20	14	9	72	1026	348	21	
19,42	15,44	11,82	74,55	1014,39	212,19	26,32	14,53	22	16	9	61	1024	19	19	
19,43	15,46	11,85	74,57	1014,38	213,05	26,35	14,53	24	18	12	54	1020	14	16	
19,43	15,47	11,87	74,60	1014,38	213,90	26,37	14,52	20	17	13	65	1019	290	19	
19,44	15,49	11,89	74,62	1014,38	214,74	26,40	14,51	20	16	12	77	1020	67	21	
19,45	15,51	11,92	74,65	1014,38	215,57	26,43	14,50	18	14	10	63	1019	54	23	
19,46	15,53	11,95	74,68	1014,38	216,40	26,45	14,49	20	16	12	74	1018	34	16	
19,47	15,55	11,97	74,71	1014,37	217,21	26,48	14,48	19	17	14	82	1017	197	23	
19,49	15,57	12,00	74,74	1014,37	218,00	26,50	14,46	18	16	13	84	1016	230	32	55
19,50	15,59	12,03	74,77	1014,37	218,79	26,52	14,44	18	16	13	79	1019	279	26	37
19,51	15,62	12,07	74,80	1014,37	219,56	26,55	14,42	16	14	13	88	1021	228	21	
19,53	15,64	12,10	74,83	1014,37	220,31	26,57	14,40	18	16	13	83	1023	316	27	
19,55	15,67	12,13	74,86	1014,37	221,05	26,59	14,37	20	17	13	79	1027	330	23	
19,56	15,69	12,17	74,90	1014,37	221,78	26,62	14,35	25	18	12	74	1025	38	16	
19,55	15,92	12,59	75,39	1014,26	248,35	27,54	14,73	27	20	14	64	1017	107	24	
19,55	15,85	12,46	75,24	1014,29	240,43	27,25	14,61	19	17	14	82	1015	329	26	

Predicted								Actual							
Max Temperature °C	**Mean Temperature °C**	**Min Temperature °C**	**Mean Humidity (%)**	**Mean Sea Level Pressure (hPa)**	**WindDir (Degrees)**	**Max Wind Speed (Km/h)**	**Max Gust Speed (Km/h)**	**Max Temperature °C**	**Mean Temperature °C**	**Min Temperature °C**	**Mean Humidity**	**Mean Sea Level Pressure (hPa)**	**WindDir (Degrees)**	**Max Wind Speed (Km/h)**	**Max Gust Speed (Km/h)**
19,57	15,84	12,42	75,20	1014,31	236,41	27,11	14,51	18	16	13	71	1017	342	42	
19,60	15,85	12,43	75,21	1014,32	234,53	27,04	14,42	21	16	12	70	1020	353	34	48
19,62	15,88	12,45	75,23	1014,32	233,84	27,02	14,36	28	21	14	59	1018	33	24	
19,65	15,91	12,48	75,26	1014,32	233,82	27,01	14,30	26	23	20	54	1016	102	24	40
19,67	15,94	12,52	75,30	1014,32	234,16	27,02	14,25	28	22	16	59	1021	96	16	
19,70	15,98	12,57	75,35	1014,32	234,69	27,04	14,20	30	23	17	57	1019	124	27	
19,73	16,01	12,61	75,39	1014,32	235,33	27,06	14,15	21	18	15	76	1023	310	24	
19,75	16,05	12,66	75,44	1014,33	236,02	27,08	14,11	22	18	15	72	1024	343	45	55
19,78	16,09	12,71	75,48	1014,33	236,73	27,10	14,06	26	20	14	66	1022	352	29	
19,81	16,13	12,76	75,53	1014,33	237,43	27,12	14,01	28	22	17	53	1021	29	23	
19,84	16,17	12,80	75,58	1014,33	238,13	27,14	13,96	29	22	16	51	1017	28	29	
19,87	16,21	12,85	75,62	1014,33	238,81	27,17	13,91	29	22	16	48	1011	355	27	
19,90	16,26	12,90	75,66	1014,33	239,48	27,18	13,86	29	21	14	57	1008	360	19	
19,93	16,30	12,95	75,71	1014,34	240,12	27,20	13,80	26	21	17	57	1010	348	19	40
19,97	16,34	13,00	75,75	1014,34	240,74	27,22	13,75	25	20	16	67	1013	154	27	
20,00	16,38	13,05	75,79	1014,35	241,34	27,24	13,70	23	19	15	77	1011	124	26	
20,03	16,43	13,10	75,83	1014,35	241,92	27,25	13,64	21	18	15	89	1007	172	23	
20,07	16,47	13,15	75,86	1014,36	242,47	27,27	13,59	19	16	13	75	1007	296	26	
20,10	16,51	13,20	75,90	1014,37	243,00	27,28	13,54	18	16	13	81	1004	235	19	
20,14	16,56	13,25	75,93	1014,37	243,50	27,29	13,48	17	14	12	84	1004	299	23	
20,17	16,60	13,30	75,96	1014,38	243,98	27,31	13,43	19	17	14	86	1009	325	42	
20,21	16,64	13,35	75,99	1014,39	244,44	27,32	13,38	24	19	15	71	1012	355	35	
20,24	16,68	13,39	76,01	1014,40	244,87	27,32	13,33	25	20	16	60	1015	16	32	
20,28	16,73	13,44	76,03	1014,41	245,28	27,33	13,28	26	21	16	58	1015	24	19	
20,32	16,77	13,48	76,05	1014,43	245,67	27,34	13,23	26	21	17	61	1012	38	21	
20,36	16,81	13,53	76,07	1014,44	246,03	27,34	13,18	25	21	17	63	1008	316	19	
20,39	16,85	13,57	76,08	1014,45	246,36	27,35	13,13	23	19	15	79	1001	352	32	
20,43	16,89	13,62	76,09	1014,47	246,68	27,35	13,08	18	17	15	88	999	278	16	
20,92	17,81	14,87	77,66	1014,28	277,08	28,59	12,37	21	18	15	79	1001	143	16	
20,81	17,59	14,58	77,32	1014,32	269,69	28,26	12,51	22	19	16	75	1004	337	34	
20,77	17,51	14,45	77,15	1014,35	265,65	28,09	12,53	21	17	14	72	1015	313	23	
20,78	17,49	14,40	77,08	1014,37	263,53	28,00	12,52	20	17	14	73	1019	296	19	
20,80	17,49	14,40	77,05	1014,39	262,49	27,95	12,48	22	18	14	72	1017	319	19	
20,83	17,52	14,42	77,03	1014,40	262,05	27,92	12,44	20	18	15	76	1012	206	24	
20,86	17,56	14,46	77,03	1014,42	261,94	27,91	12,39	19	17	14	73	1014	237	29	45
20,90	17,60	14,50	77,04	1014,44	262,00	27,90	12,34	21	18	16	69	1019	256	24	
20,95	17,64	14,54	77,04	1014,45	262,15	27,90	12,29	25	19	14	63	1019	2	32	
20,99	17,68	14,58	77,04	1014,47	262,33	27,90	12,24	30	22	15	65	1019	6	27	
21,03	17,73	14,62	77,04	1014,49	262,53	27,89	12,19	31	23	17	62	1018	358	21	
21,07	17,77	14,67	77,04	1014,51	262,72	27,89	12,15	30	24	19	62	1015	16	19	
21,12	17,81	14,71	77,03	1014,54	262,89	27,89	12,10	29	24	20	58	1014	17	23	
21,16	17,85	14,74	77,02	1014,56	263,05	27,88	12,06	29	23	18	59	1017	333	26	34
21,20	17,89	14,78	77,00	1014,58	263,19	27,87	12,02	28	23	19	50	1018	44	27	
21,24	17,93	14,82	76,98	1014,61	263,30	27,87	11,98	28	23	18	59	1019	65	19	
21,29	17,97	14,85	76,96	1014,63	263,40	27,86	11,94	25	21	18	81	1017	83	29	52
21,33	18,01	14,89	76,93	1014,66	263,46	27,84	11,90	21	19	18	85	1011	17	23	
21,37	18,05	14,92	76,90	1014,69	263,51	27,83	11,87	26	21	16	75	1012	32	27	
21,41	18,09	14,95	76,87	1014,72	263,53	27,82	11,84	27	22	17	69	1017	1	34	
21,45	18,12	14,98	76,83	1014,75	263,53	27,80	11,81	25	21	17	75	1020	332	32	

Predicted								Actual							
Max Temperature ºC	Mean Temperature ºC	Min Temperature ºC	Mean Humidity (%)	Mean Sea Level Pressure (hPa)	WindDir (Degrees)	Max Wind Speed (Km/h)	Max Gust Speed (Km/h)	Max Temperature ºC	Mean Temperature ºC	Min Temperature ºC	Mean Humidity	Mean Sea Level Pressure (hPa)	WindDir (Degrees)	Max Wind Speed (Km/h)	Max Gust Speed (Km/h)
21,49	18,16	15,01	76,79	1014,78	263,50	27,79	11,78	27	21	15	69	1021	344	23	
21,53	18,19	15,03	76,74	1014,81	263,45	27,77	11,75	29	23	17	60	1021	343	26	
21,58	18,22	15,06	76,69	1014,84	263,38	27,75	11,72	33	24	17	57	1020	17	21	
21,62	18,26	15,08	76,64	1014,88	263,29	27,73	11,70	29	24	21	57	1019	11	21	
21,66	18,29	15,10	76,58	1014,91	263,17	27,71	11,67	23	20	17	72	1019	280	19	
21,70	18,32	15,12	76,52	1014,94	263,03	27,69	11,65	29	23	17	75	1015	25	23	
21,74	18,35	15,14	76,46	1014,98	262,87	27,67	11,63	27	23	20	74	1013	31	21	
21,77	18,37	15,16	76,39	1015,01	262,68	27,64	11,61	24	21	18	80	1014	194	27	
21,81	18,40	15,17	76,32	1015,05	262,48	27,62	11,59	23	20	17	86	1014	25	37	37
22,68	19,78	16,96	78,69	1014,82	293,66	29,07	10,28	23	20	17	73	1020	336	35	
22,50	19,50	16,60	78,28	1014,84	287,07	28,73	10,49	28	22	16	62	1019	20	24	
22,42	19,36	16,42	78,04	1014,87	283,18	28,54	10,59	26	21	17	42	1018	38	29	45
22,40	19,30	16,33	77,88	1014,91	280,88	28,42	10,63	29	24	19	50	1015	13	26	
22,41	19,28	16,29	77,76	1014,94	279,48	28,34	10,64	30	25	20	56	1012	347	23	
22,42	19,28	16,27	77,66	1014,97	278,59	28,28	10,65	28	23	18	62	1008	326	29	
22,45	19,29	16,26	77,58	1015,01	277,98	28,24	10,64	22	19	16	77	1009	329	35	
22,48	19,31	16,26	77,50	1015,05	277,51	28,20	10,63	20	18	15	68	1013	330	39	
22,52	19,32	16,27	77,42	1015,08	277,11	28,17	10,62	21	18	15	62	1018	329	26	42
22,55	19,34	16,27	77,34	1015,12	276,74	28,13	10,61	19	17	15	70	1019	270	16	
22,58	19,36	16,28	77,26	1015,16	276,38	28,10	10,60	24	19	15	68	1019	338	27	
22,62	19,38	16,29	77,17	1015,20	276,01	28,06	10,59	26	20	15	65	1020	345	27	34
22,65	19,40	16,29	77,09	1015,24	275,63	28,03	10,59	27	21	15	64	1019	336	27	
22,69	19,42	16,29	77,00	1015,28	275,24	27,99	10,58	25	21	17	73	1021	324	24	
22,72	19,44	16,30	76,91	1015,31	274,82	27,96	10,57	25	21	17	68	1021	335	29	
22,75	19,45	16,30	76,82	1015,35	274,38	27,92	10,57	26	22	18	72	1022	339	34	
22,78	19,47	16,30	76,73	1015,39	273,92	27,88	10,56	25	21	17	69	1021	340	27	
22,81	19,48	16,30	76,64	1015,43	273,43	27,84	10,56	25	20	16	71	1019	328	27	
22,84	19,50	16,30	76,54	1015,47	272,93	27,80	10,56	23	19	15	64	1020	346	37	
22,87	19,51	16,30	76,45	1015,51	272,40	27,76	10,55	29	22	15	58	1019	351	37	
22,90	19,52	16,29	76,35	1015,55	271,85	27,72	10,55	29	23	18	66	1018	339	27	
22,93	19,53	16,29	76,25	1015,59	271,27	27,68	10,55	30	23	17	67	1018	337	29	
22,96	19,54	16,28	76,16	1015,62	270,68	27,64	10,54	27	22	17	63	1020	355	37	
22,99	19,55	16,28	76,06	1015,66	270,06	27,59	10,54	25	20	16	61	1021	351	40	
23,02	19,56	16,27	75,96	1015,70	269,42	27,55	10,54	34	26	17	53	1018	1	40	
23,04	19,57	16,27	75,87	1015,74	268,76	27,51	10,54	35	28	22	41	1017	320	19	
23,07	19,58	16,26	75,77	1015,77	268,08	27,46	10,54	37	29	22	53	1016	108	29	
23,10	19,59	16,25	75,67	1015,81	267,37	27,42	10,53	28	24	20	78	1016	246	29	
23,12	19,60	16,24	75,57	1015,85	266,65	27,37	10,53	26	22	18	68	1018	339	45	
23,15	19,60	16,23	75,48	1015,88	265,90	27,32	10,53	26	21	16	62	1018	334	37	50
24,19	21,12	18,13	77,95	1015,72	297,47	28,79	9,16	32	24	17	58	1016	338	29	
23,99	20,84	17,80	77,62	1015,72	291,35	28,48	9,33	30	26	22	46	1016	330	21	
23,90	20,70	17,61	77,40	1015,73	287,50	28,29	9,43	26	22	18	63	1013	322	32	37
23,85	20,62	17,50	77,24	1015,75	284,97	28,16	9,48	21	19	18	73	1014	322	37	48
23,84	20,57	17,44	77,10	1015,78	283,21	28,07	9,51	25	21	17	66	1018	331	35	
23,84	20,55	17,39	76,98	1015,81	281,87	27,99	9,53	27	22	17	76	1019	332	32	40
23,85	20,53	17,36	76,87	1015,85	280,76	27,93	9,54	25	21	18	66	1018	341	34	
23,86	20,53	17,33	76,76	1015,88	279,77	27,87	9,55	23	19	16	61	1017	337	42	55
23,87	20,52	17,31	76,65	1015,91	278,85	27,81	9,56	25	20	16	69	1018	336	27	
23,89	20,51	17,29	76,55	1015,95	277,94	27,76	9,57	25	22	20	69	1019	316	26	

Predicted								Actual							
Max Temperature °C	**Mean Temperature °C**	**Min Temperature °C**	**Mean Humidity (%)**	**Mean Sea Level Pressure (hPa)**	**WindDir (Degrees)**	**Max Wind Speed (Km/h)**	**Max Gust Speed (Km/h)**	**Max Temperature °C**	**Mean Temperature °C**	**Min Temperature °C**	**Mean Humidity**	**Mean Sea Level Pressure (hPa)**	**WindDir (Degrees)**	**Max Wind Speed (Km/h)**	**Max Gust Speed (Km/h)**
23,91	20,51	17,27	76,45	1015,98	277,05	27,70	9,57	26	22	19	69	1017	335	29	
23,92	20,51	17,24	76,35	1016,01	276,14	27,65	9,58	26	21	17	66	1015	335	24	
23,94	20,50	17,22	76,25	1016,05	275,22	27,59	9,58	25	21	17	71	1016	333	48	58
23,95	20,50	17,20	76,15	1016,08	274,29	27,54	9,59	25	20	16	62	1020	346	40	52
23,97	20,49	17,18	76,05	1016,11	273,34	27,48	9,59	28	22	17	64	1019	346	40	58
23,98	20,49	17,16	75,95	1016,14	272,36	27,43	9,59	29	23	17	67	1019	337	35	
23,99	20,48	17,14	75,86	1016,17	271,37	27,38	9,60	25	21	18	77	1019	344	37	52
24,01	20,48	17,12	75,77	1016,20	270,37	27,32	9,60	23	20	17	68	1016	346	60	69
24,02	20,47	17,10	75,67	1016,23	269,34	27,27	9,60	25	20	16	70	1017	343	39	60
24,03	20,47	17,08	75,58	1016,25	268,29	27,21	9,60	24	20	17	70	1017	338	50	69
24,05	20,46	17,06	75,49	1016,28	267,23	27,16	9,60	27	21	16	62	1016	348	50	52
24,06	20,46	17,03	75,41	1016,31	266,14	27,10	9,60	25	21	17	70	1013	344	45	66
24,07	20,45	17,01	75,32	1016,34	265,04	27,05	9,60	26	21	16	65	1012	335	42	58
24,08	20,44	16,99	75,24	1016,36	263,92	27,00	9,60	26	21	16	65	1015	327	35	39
24,09	20,44	16,97	75,16	1016,39	262,79	26,94	9,60	32	24	18	60	1016	337	45	
24,11	20,43	16,95	75,08	1016,41	261,64	26,89	9,59	31	24	18	62	1015	339	42	53
24,12	20,43	16,93	75,00	1016,43	260,48	26,83	9,59	29	24	19	66	1013	344	48	
24,13	20,42	16,91	74,93	1016,46	259,29	26,78	9,59	33	26	19	53	1012	335	37	
24,14	20,41	16,89	74,85	1016,48	258,10	26,73	9,58	30	26	21	56	1014	270	23	
24,15	20,41	16,87	74,78	1016,50	256,89	26,67	9,58	27	22	18	74	1016	262	24	
24,16	20,40	16,85	74,71	1016,52	255,67	26,62	9,57	26	22	18	76	1017	328	35	35
25,20	21,83	18,62	76,89	1016,46	289,20	27,98	8,43	28	23	18	66	1014	334	37	
25,02	21,60	18,35	76,70	1016,42	283,21	27,72	8,54	22	20	18	85	1010	332	34	
24,92	21,47	18,19	76,55	1016,41	279,24	27,56	8,60	25	21	18	77	1016	303	26	
24,87	21,39	18,08	76,44	1016,42	276,45	27,44	8,64	26	22	18	69	1020	332	35	
24,84	21,34	18,01	76,33	1016,43	274,33	27,34	8,66	28	23	19	74	1021	338	37	
24,83	21,30	17,96	76,24	1016,44	272,57	27,26	8,67	28	24	20	69	1018	326	26	
24,82	21,28	17,92	76,15	1016,46	271,02	27,19	8,68	25	22	20	85	1015	263	23	
24,82	21,25	17,88	76,07	1016,48	269,58	27,13	8,69	28	22	17	66	1014	345	32	
24,82	21,23	17,84	75,99	1016,50	268,20	27,07	8,70	29	23	18	58	1017	346	35	
24,82	21,22	17,81	75,91	1016,52	266,83	27,00	8,70	36	28	20	42	1017	360	29	
24,82	21,20	17,78	75,83	1016,54	265,48	26,94	8,70	33	27	22	38	1016	348	27	
24,82	21,18	17,75	75,75	1016,56	264,12	26,88	8,70	33	27	22	50	1014	334	35	
24,82	21,17	17,72	75,68	1016,57	262,76	26,82	8,71	30	24	19	63	1013	333	32	
24,82	21,15	17,69	75,61	1016,59	261,38	26,77	8,71	29	24	19	71	1014	335	32	37
24,82	21,13	17,66	75,54	1016,61	260,00	26,71	8,71	28	24	20	75	1014	337	26	
24,82	21,12	17,63	75,47	1016,62	258,60	26,65	8,71	30	24	19	77	1016	325	27	
24,82	21,10	17,60	75,41	1016,64	257,20	26,59	8,70	27	22	18	77	1016	335	27	
24,82	21,09	17,57	75,34	1016,65	255,78	26,53	8,70	29	23	17	66	1014	332	27	29
24,82	21,07	17,55	75,28	1016,67	254,36	26,48	8,70	28	22	16	68	1011	333	26	
24,82	21,06	17,52	75,22	1016,68	252,92	26,42	8,70	33	26	18	59	1013	357	21	
24,83	21,04	17,49	75,17	1016,70	251,48	26,36	8,69	30	27	23	56	1014	333	32	52
24,83	21,03	17,46	75,11	1016,71	250,03	26,31	8,69	25	23	21	82	1014	195	26	
24,83	21,02	17,44	75,06	1016,72	248,57	26,25	8,68	24	22	21	80	1016	255	27	
24,83	21,00	17,41	75,01	1016,73	247,11	26,20	8,68	26	22	19	68	1018	325	29	
24,83	20,99	17,39	74,96	1016,74	245,64	26,15	8,67	28	22	17	65	1018	330	27	
24,83	20,98	17,36	74,91	1016,75	244,16	26,09	8,66	25	21	18	73	1016	270	26	
24,83	20,96	17,34	74,87	1016,76	242,68	26,04	8,65	25	21	17	62	1017	335	34	47
24,83	20,95	17,32	74,82	1016,77	241,20	25,99	8,64	28	22	16	55	1018	3	34	

Predicted								Actual							
Max Temperature °C	**Mean Temperature °C**	**Min Temperature °C**	**Mean Humidity (%)**	**Mean Sea Level Pressure (hPa)**	**WindDir (Degrees)**	**Max Wind Speed (Km/h)**	**Max Gust Speed (Km/h)**	**Max Temperature °C**	**Mean Temperature °C**	**Min Temperature °C**	**Mean Humidity**	**Mean Sea Level Pressure (hPa)**	**WindDir (Degrees)**	**Max Wind Speed (Km/h)**	**Max Gust Speed (Km/h)**
24,83	20,94	17,29	74,78	1016,78	239,72	25,93	8,63	27	21	16	64	1017	330	29	
24,83	20,93	17,27	74,74	1016,79	238,23	25,88	8,62	25	20	16	70	1017	334	27	
24,84	20,92	17,25	74,71	1016,80	236,74	25,83	8,61	24	20	17	80	1015	255	26	
25,79	22,17	18,77	76,41	1016,82	270,15	26,95	7,74	23	21	19	85	1010	217	29	45
25,63	21,98	18,56	76,31	1016,78	264,34	26,76	7,80	23	20	17	86	1008	256	21	39
25,55	21,88	18,43	76,24	1016,75	260,36	26,62	7,83	22	19	16	85	1014	1	19	35
25,50	21,81	18,35	76,17	1016,74	257,43	26,52	7,85	24	20	16	76	1016	288	21	39
25,47	21,76	18,28	76,11	1016,74	255,09	26,44	7,87	23	20	17	72	1019	337	29	
25,45	21,72	18,23	76,05	1016,75	253,07	26,37	7,87	25	20	15	68	1022	355	29	
25,43	21,69	18,19	75,99	1016,75	251,23	26,30	7,88	31	23	17	55	1023	358	24	37
25,42	21,66	18,15	75,94	1016,76	249,50	26,24	7,88	32	24	18	45	1022	347	27	
25,41	21,64	18,11	75,89	1016,77	247,82	26,18	7,88	32	24	18	69	1016	350	23	32
25,40	21,61	18,08	75,84	1016,78	246,17	26,12	7,88	30	23	16	69	1012	288	26	
25,39	21,59	18,05	75,79	1016,78	244,54	26,07	7,88	24	20	17	81	1015	273	23	
25,39	21,57	18,01	75,74	1016,79	242,91	26,01	7,88	26	21	17	74	1020	327	23	
25,38	21,55	17,98	75,70	1016,80	241,29	25,96	7,87	29	23	18	71	1019	339	29	48
25,37	21,53	17,95	75,66	1016,80	239,67	25,90	7,87	30	24	19	65	1014	340	27	
25,37	21,51	17,92	75,61	1016,81	238,06	25,85	7,87	31	24	18	63	1012	308	23	35
25,36	21,49	17,89	75,58	1016,81	236,45	25,79	7,86	29	23	18	68	1013	291	23	39
25,35	21,47	17,86	75,54	1016,82	234,84	25,74	7,86	28	23	18	72	1016	335	26	42
25,35	21,45	17,83	75,50	1016,82	233,23	25,69	7,86	23	20	17	73	1018	344	37	55
25,34	21,44	17,81	75,47	1016,83	231,62	25,64	7,85	22	19	16	64	1018	357	50	69
25,33	21,42	17,78	75,43	1016,83	230,02	25,59	7,84	29	22	15	46	1018	1	32	
25,33	21,40	17,75	75,40	1016,84	228,43	25,54	7,84	31	23	16	32	1017	7	24	35
25,32	21,38	17,73	75,37	1016,84	226,83	25,49	7,83	25	20	16	63	1018	326	24	
25,32	21,37	17,70	75,35	1016,84	225,25	25,44	7,82	25	20	16	73	1019	328	29	
25,31	21,35	17,67	75,32	1016,85	223,67	25,39	7,82	23	19	16	70	1017	338	23	37
25,31	21,34	17,65	75,29	1016,85	222,10	25,34	7,81	23	19	15	80	1016	266	21	
25,30	21,32	17,63	75,27	1016,85	220,54	25,29	7,80	27	21	15	79	1017	319	26	
25,30	21,30	17,60	75,25	1016,85	218,98	25,25	7,79	29	22	16	66	1020	345	24	27
25,29	21,29	17,58	75,23	1016,85	217,43	25,20	7,78	30	23	16	54	1022	308	21	
25,29	21,28	17,56	75,21	1016,86	215,90	25,16	7,77	29	26	22	41	1019	353	23	35
25,28	21,26	17,54	75,19	1016,86	214,37	25,11	7,76	30	24	18	47	1018	354	24	34
26,10	22,29	18,77	76,48	1016,92	243,29	25,94	7,08	30	24	19	56	1017	343	19	29

Table 7 - Weather Proof of Concept Data

7.7 Annex D

In this annex the program interface will be showed and the available options explained.

The main screen of the software presents the user with two options. ***Direct Input*** where the user inserts the origin, destination and dates on a table and the ***File Read*** option were the user selects the files containing that information.

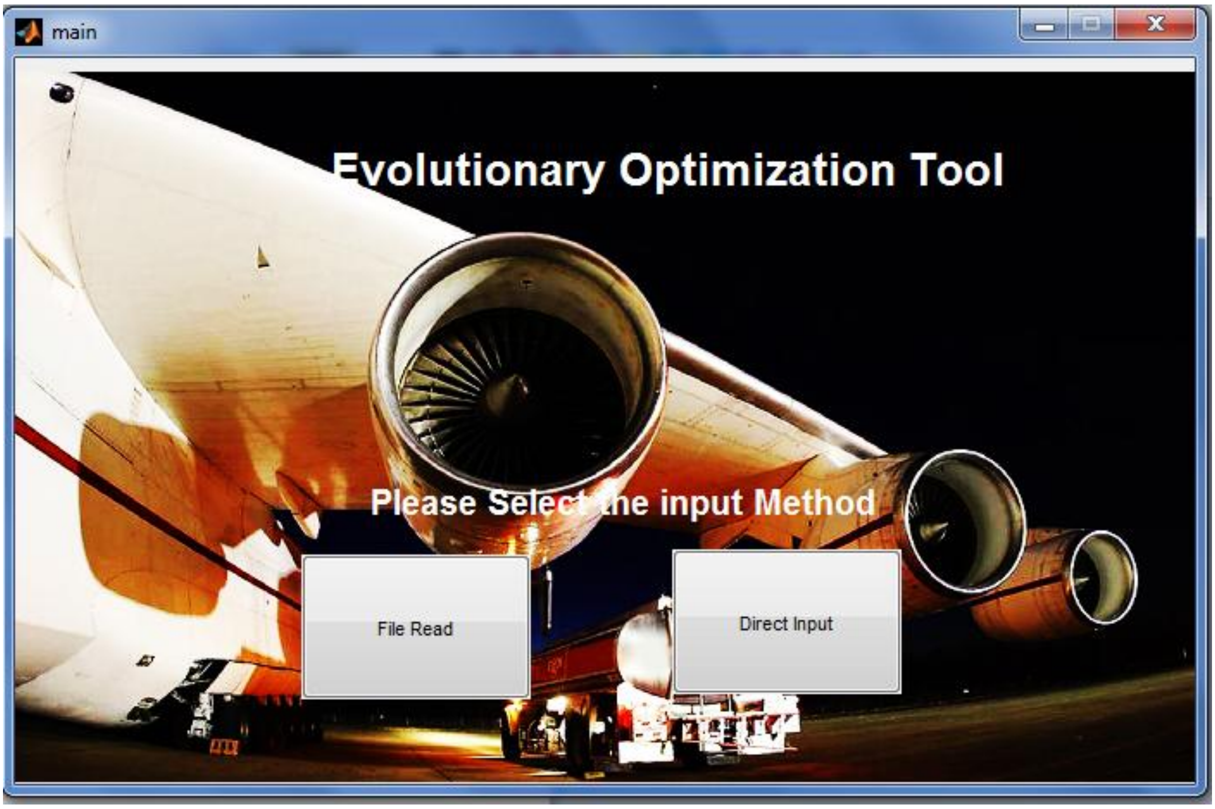

Figure 25 - Main screen

The ***File Read*** option allows for the input of several flights at the same time. The number of flights that can be processed on this mode is only limited to the time that the user has available. The screen has two separated blocks: One to select the File containing the flight data (Origin, Destination, Date) and predicted flight occupation and the other to load a file containing the list of the excluded planes registration.

The process of selecting the file starts with the ***Add File*** option, where the user is prompt to a browser screen where it can select the input file.

The file can be in the ''.csv'' or ''.txt'' formats. After adding the file the user should select ***Read Data*** to allow to program to read the file contents. The ***Delete File*** option allows the user to eliminate files from the list. Those files won't be deleted from the hard drive.

The file with the excluded planes is not mandatory unlike the one with the flight data. After reading the contents of both files the user should change the simulation time in order to control the duration of the simulation.

If the software reaches the best solution before that time it will stop and save the results. The ***Optimize*** button allows the user to run the optimization and the ***Cancel*** and ***Clear*** buttons allow the user to cancel and clear the simulation respectively.

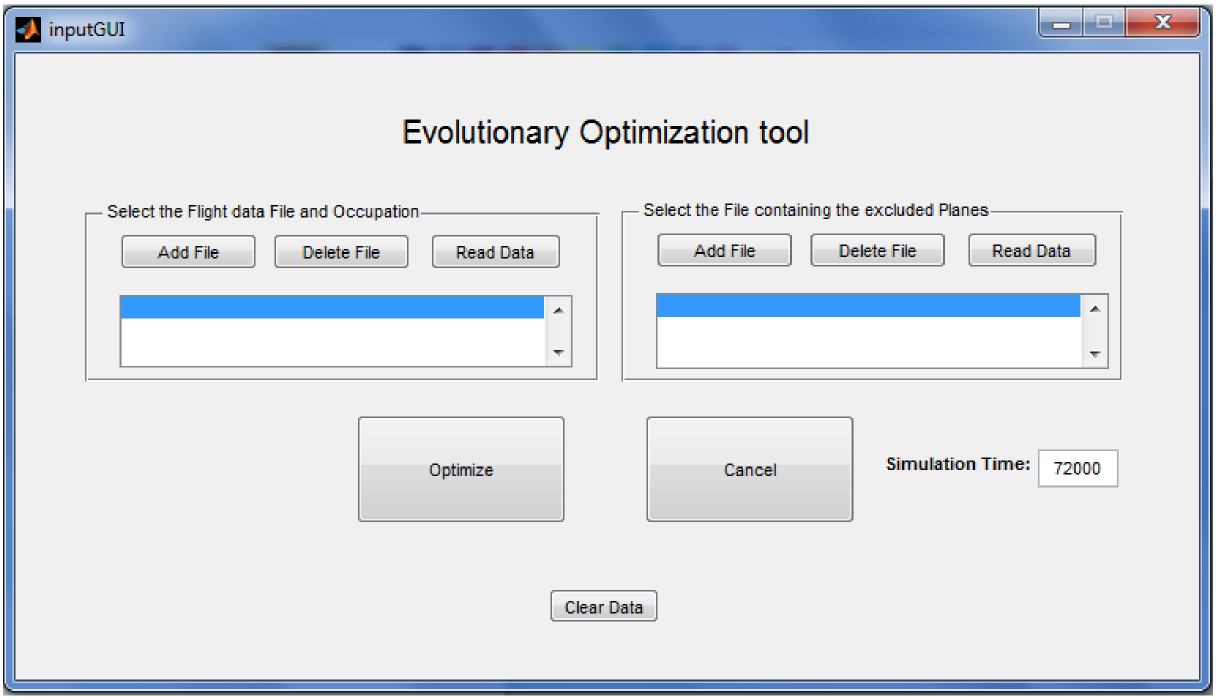

Figure 26 - File Read option

The ***Direct Input*** option is similar in functionality to the ***File Read*** option. The only difference is the input method. On this mode the user can fill the table with the desired values and change the default simulation time. The ***Optimize*** and ***Clear*** buttons have the same functionality as the one on the ***File Read*** option.

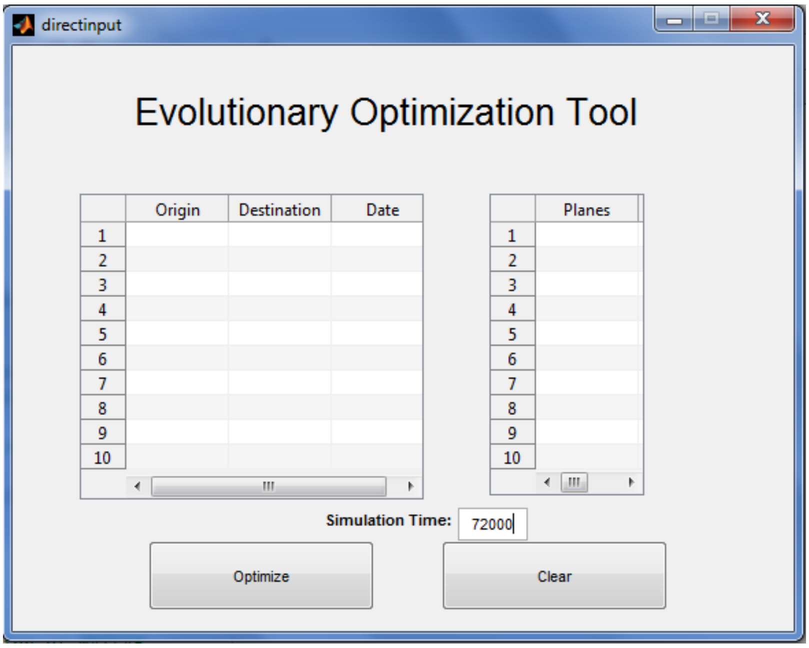

Figure 27 - Direct Input method

After selecting the ***Optimize*** button the algorithm starts the optimization and presents the user with the genetic algorithm plot. This plot is updated on each generation and allows the user to follow the evolution of the algorithm. From the upper left clockwise we have: the plot of the fitness value versus the generation where the fitness value is total fuel consumption; the Current Best Individual where we have the flights and the number of the plane selected; the genealogy which is only presented on the final generation; the simulation status containing the Stall Time, Stall Generation, Time elapsed and Generation; the score of each individual; the Expectation versus the Raw scores; the Individual versus The Number of children.

The user can either stop or pause the simulation selecting their respective buttons. When the simulation is stopped either by the user or by its convergence factors the Fuel_Optimization_results_v_0.1 and the Fuel_Optimization_results_v_0.2 are created. The first one is formatted to be read on Microsoft Excel© versions prior to 2000 and the second one is formatted to be read on more recent versions of Microsoft Excel ©.

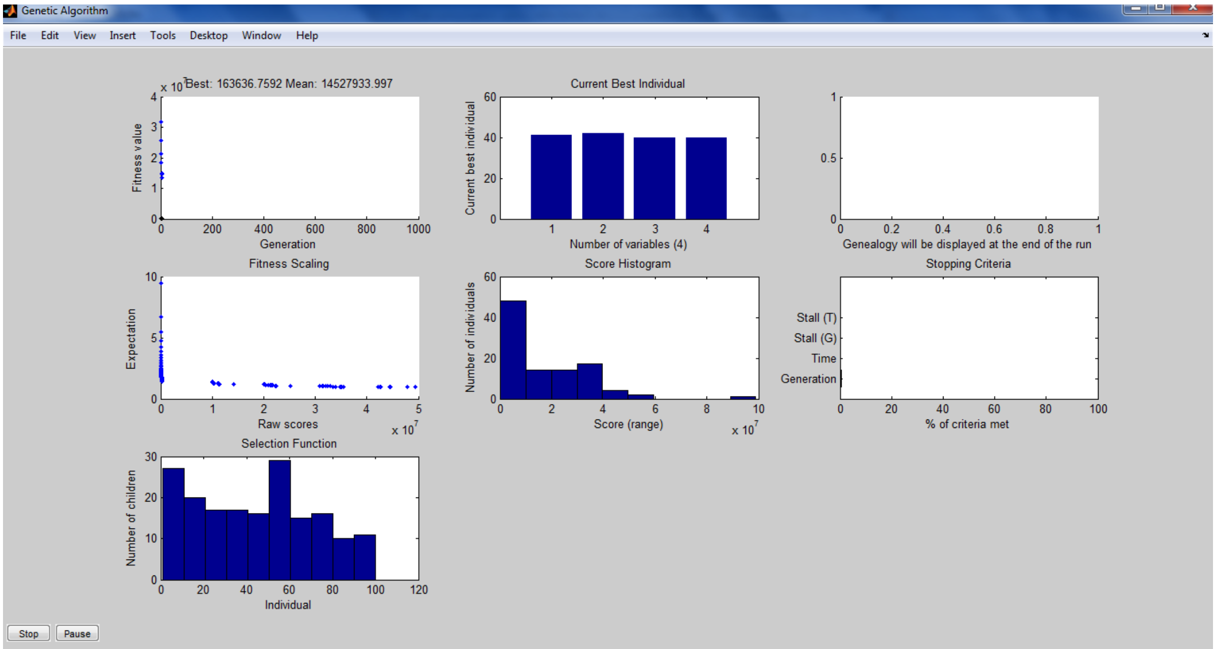

Figure 28 - Genetic Algorithm evolution

7.8 Annex E

This annex has an example of the input file, as well as an example of the output file. These examples were made using 5 flights selected randomly.

The Flight Data input file can be created as a txt or in excel and saved as a CSV file:

FOR	LIS	01-08-2010 00:44
NAT	LIS	01-08-2010 00:40
VCP	LIS	01-08-2010 01:50
GIG	LIS	01-08-2010 01:46

Table 8-Flight Data

The planes excluded file contains the registration code of the planes that should not be taken into account:

CSTTA
CSTTB
CSTTC
CSTTD
CSTTE

Table 9 - Planes Excluded

The output file can be opened on Excel and has the following format:

Origin	Destination	Date and time of departure	Registration of the plane used	Plane Model	Duration of flight in hours	Trip Fuel	Contingency Fuel	Alternative Fuel	Fuel Reserve	ETOPS	Holding distance Fuel	Additional Fuel	Take off Fuel	Taxi Fuel	Fuel on Board	Actual Fuel on Board	Fuel Remaining	Zero fuel Weight	Actual Zero Fuel Weight	CO_2
FOR	LIS	01-08-2010 00:44	CSTOF	330	6,2525	40808,03	1408,735	3455,382	2166,076	287,8835	292,1174	419,5657	48886,44	448,6643	49131,93	49344,76	8500,46	152412,2	151440,2	128659,5
NAT	LIS	01-08-2010 00:40	CSTOE	330	6,1469	38935,47	1320,575	3146,451	2190,593	243,8043	124,1927	415,9129	46663,14	448,6643	49131,93	49344,76	8500,46	152412,2	151440,2	128659,5
VCP	LIS	01-08-2010 01:50	CSTOP	330	8,5938	34466,47	1308,779	2923,897	2055,964	328,8048	184,1535	531,3296	41890,66	447,8792	49222,18	49435,07	8504,949	152488,5	151513,5	128929,9
GIG	LIS	01-08-2010 01:46	CSTOG	330	8,3927	38646,61	1344,393	3213,856	2157,32	311,774	169,0217	515,8099	46553,16	447,8792	49222,18	49435,07	8504,949	152488,5	151513,5	128929,9

Table 10 - Fuel Output