FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Evolutionary Computation methods applied to Operational Control Centers

António José Ferreira de Castro Moura

WORKING VERSION

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Ana Paula Rocha

Second Supervisor: António Castro

June 22, 2015

Evolutionary Computation methods applied to Operational Control Centers

António José Ferreira de Castro Moura

Mestrado Integrado em Engenharia Informática e Computação

June 22, 2015

Abstract

During the execution of an operational plan, there is the likelihood of this plan being affected by some disruptions caused by unexpected events. The disruptions affect at least three dimensions that airline companies and the operational control centers must take into account which are passenger, crew and aircraft. Usually, a disruption is a state during which the current operation being executed is affected by a deviation (which is large enough to cause a change) from the original plan, and sometimes unfortunately it leads to an unfeasible plan. Examples of events that might cause disruptions are bad weather, threats or terrorist attacks and aircraft malfunctions.

Disruption Management, can be defined as the process that starts after the deviation from the original plan is detected. After the disruption, the plan is changed and it will no longer be as close as it was from an optimal plan or it can even turn into an unfeasible plan. Either way there is a need to review the plan and try to minimize the impact caused by the disruption.

MASDIMA is useful to help Airline Operation Control Centers finding a solution to disruptions during an operational plan, and in order to improve both computing time and the quality of solutions, there is a need to improve the system. This will translate in minimising the impact both in terms of costs or delays.

To deal with that problem three agents will be implemented that will reflect, each one, different evolutionary computation algorithms (Particle Swarm Optimisation, Ant Colony Optimisation and Genetic Algorithms) and are related to the aircraft dimension of the problem. These agents will be implemented on a Multi-Agent System named MASDIMA that represents an Operation Control Center.

Resumo

Durante a execução de um plano operacional, existe a possibilidade do mesmo sofrer rupturas causadas por eventos não esperados. As rupturas afetam pelo menos três dimensões sobre as quais as companhias aéreas e os centros de controlo operacional devem ter em conta, nomeadamente os passageiros, a tripulação e os aviões. Normalmente, um ruptura é um estado durante o qual uma operação que esteja a ser executada é afetada por um desvio (que é grande o suficiente para causar uma mudança) do plano original e, por vezes, levando a que o plano não seja execuível. Exemplos de eventos que podem causar rupturas são condições meteorológicas, ameaças ou ataques terroristas e avarias nos aviões.

Disruption Management, pode então ser definido como o processo que começa após detectar o desvio do plano original. Depois da ruptura, o plano é mudado e nunca mais vai estar tão perto da solução ótima quanto estava antes da ruptura, sendo que pode mesmo vir a ser impossível a continuação do plano. De qualquer maneira existe a necessidade de rever o plano e de minimizar o impacto causado pela ruptura.

O MASDIMA é uma grande ajuda para os Centros de Controlo Operacionais das companhias aéreas encontrarem soluções para rupturas durante a execução de um plano operacional, e para melhorar quer o tempo de computação quer a qualidade das soluções, existe a necessidade de melhorar o sistema. Isto traduz-se em minimizar o impacto quer a ao nível dos custos quer ao nível dos atrasos.

Para lidar com este problema serão implementados três agentes, sendo que cada um representa um algoritmo evolutivo diferente (Particle Swarm Optimisation, Ant Colony Optimisation e Genetic Algorithms) e estarão relacionados com a dimensão avião relativa ao problema. Estes três agentes serão implementados num Sistema Multi-Agente chamado de MASDIMA que representará o Centro de Controlo Operacional.

Acknowledgements

I would like to express my special appreciation and thanks to my supervisors Professor Ana Paula Rocha and Professor António Castro, for guiding me through this new academic stage, all the motivation towards the subject and last but not least for the trust they placed in me.

To BEST Porto and all the friends I have made there, which made me improve my personal and soft skills and for the fantastic support.

To my friends which without them I would not be who I am today, and these last years would not be the same.

A special thanks to my family, for all they taught me and for all the sacrifices that they have made on my behalf.

António José Ferreira de Castro Moura

"I can't change the direction of the wind,
but I can adjust my sails to always reach my destination."

Jimmy Dean

Contents

CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

ARO	Aircraft Recovery
AOCC	Airline Operations Control Center
AI	Artificial Inteligence
DM	Disruption Management
EA	Evolutionary Algorithm
IROP	Irregular Operations
MASDIMA	Multi-Agent System for Disruption Management
OCC	Operations Control Center

Chapter 1

Introduction

This chapter will present the context on which the work is going to be performed including the motivation and objectives of the work. The chapter ends with a summary on each of the chapters presented further on this report.

1.1 Context

Airline companies are doing a great effort in order to maximize their revenues while keeping their costs at a minimum. This is no easy task, and due to that, they are investing in tools that optimize their operational schedules. In spite of having an optimal plan, even this ones have a strong probability of being affected during the operation, by disruptions as weather changes, aircraft malfunctions or extra maintenance and crew absenteeism, which may lead into delayed flights causing an irregular operation (IROPS). In order to manage disruptions the Airline Operations Control Centre (AOCC) try to find a solution that will result in a minimum impact either for the flight schedule as well as for the cost. Usually AOCC are managing to solve disruptions using a sequential process, i.e., the process used to solve the disruption is executed in an order, in which the three dimensions of the problem - aircraft, crew and passenger - are present in this same order. [MAS12]

While using this sequential approach, different importances are given to the dimensions of the problem, and will restrict too much the last dimensions to be solved, making it harder to obtain the best integrated solution. MASDIMA however is capable of monitoring the operation plan and deciding what actions to do if an event requires so. [MAS15]

“MASDIMA is capable of monitoring the operational plan and deciding if an event requires or not an action. It is autonomous with decision making capabilities and automates the repetitive tasks; it is adaptive to changes on the environment (includes learning capabilities); it is dynamic and provides solutions in almost real-time and allows the inclusion of a human-in-the-loop to

improve the user acceptance of the solutions found automatically by reacting and learning the preferences of this user." [MAS12]

1.2 Motivation and Objectives

Nowadays, one of the problems that the airline industry is facing happens during the day of operations and refers to unexpected events which can disrupt and jeopardise the operational plan, leading to an IROPS. Examples of these events can be an adverse change in the weather conditions, unexpected aircraft breakdown leading to a longer maintenance, sick crew and so on. If a proper recovery plan is not taken in a short time, the disruptions can propagate in a large scale over time, yielding new disruptions. One example of this can be if the crew is meant to do another flight and they are stuck at the previous airport, delaying the flight were they were supposed to be or the need to assign a new crew to that flight. Airlines try to minimize the impact of these events by, for example, using the same crew to perform a set of flights instead of a different crew each flight. Since the use of expensive recovery actions like ferrying an aircraft, rebook passengers on other airlines and hire a new crew for another flight, increasing the operational cost of the recovery plan and reduces the expected revenues the airline company had with the flight, a proper plan must be taken into account. [AENA04]

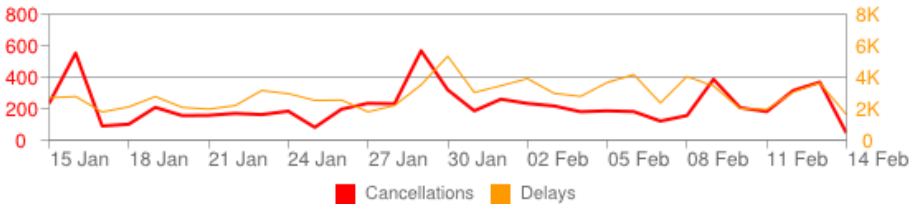

Figure 1.1: European Canceled and Delayed Flights - 2015
[CDF15]

According to the data on the figure 1.1, during a period of thirty days there was a total of 6.861 flights canceled and 88.399 flights delayed only in Europe. The y-axis represents the number for flights the left y-axis presents the number of canceled flights and the right y-axis the number of delayed flights (this last in a scale of 1 to 1.000).

To get some awareness about the dimension of this problem, consider for instance the American Airlines company, which schedules about 510 aircraft of 14 types to 140 cities covering a total of 2.700 flights and assigns 25.000 crew members to these 2.700 flights. In order to operate a system with such a magnitude and complexity, airlines rely on optimisation techniques for their planning, attempting to get an optimal initial plan, making an efficient use of resources, leading to better revenues. [CCZ10]

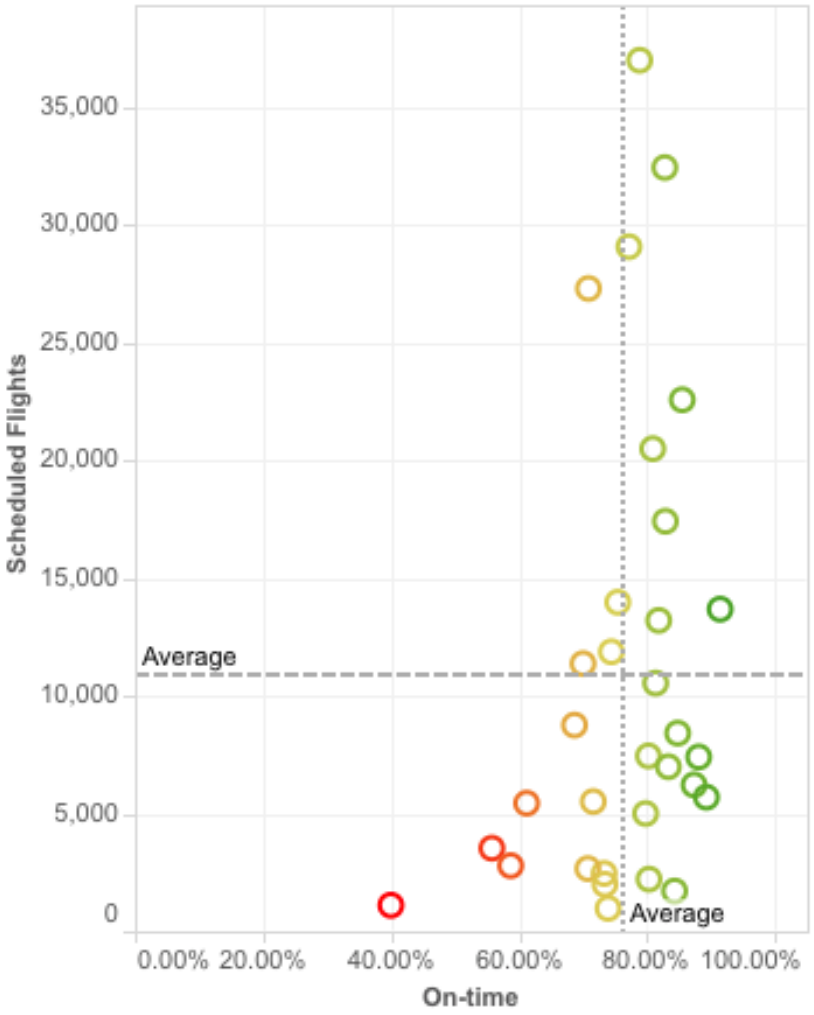

Figure 1.2: Major Airlines from Europe - Arrival Performance, December 2014 [AFS15]

Figure 1.2 represents the percentage of delayed flights in relation to the number of scheduled flights by the major airlines in Europe.

Despite of the average delay among the major Airlines from Europe being 42.05 minutes, only 16% of the flights arrive with delay. According to a study from [CTA04] for each minute of delay at-gate after the first 15 minutes, there is a value of € 72 per minute of delay.

Studies have been made through several airline companies and they indicate that the cost with IROPS can cost up to 3% of the airline annual revenue, [CCZ10]. It was estimated that a better recovery process could result in cost reductions with IROPS of at least 20%, [Irr96]. According to experiments with TAP Portugal data, show that in the worst case scenario, with a better recovery process it is possible to improve the cost reduction of IROPS between 13% to 17%, [MAS15].

Since there is a way to reduce the costs with IROPS, by improving the recovery process, in this dissertation there will be presented a study regarding the implementation of methods of

evolutionary computation for the recovery process in OCCs. Within this study there will also be the implementation of the methods on MASDIMA and a benchmarking between the methods to be implemented and the ones implemented on the system before by defining metrics to do so.

1.3 Thesis Structure

Beyond the introduction, this dissertation is divided into four more chapters. Chapter 2 reviews the state of the art on the field of Disruption Management (DM) and the methods to be implemented are described. Chapter 3 presents the approach taken in the implementation of each method. Chapter 4 presents the results and the benchmarking of all methods. And the chapter 5 closes the dissertation, presenting a conclusion and some additional ideas to be taken into account regarding future work.

Chapter 2

Background

In this chapter background information will be presented, related with Disruption Management in airline operations control as well as with evolutionary algorithms.

2.1 Introduction

This section aims to provide some background information regarding disruption management shortly before or at the day of operations and evolutionary algorithms a subset of evolutionary computation.

During the operational plan, the schedule often has to be revised due to disruptions with distinct sources, as nature related, technical problems or crew related. In order to provide some details regarding Disruption Management, in the following section presents an introduction to Disruption Management in airline industry and a description of the planning process as well as some methods airlines use in order to create a more flexible schedule.

Since we are going to use three classes of evolutionary algorithms during the disruption management task, namely *Particle Swarm Optimisation*, *Ant Colony Optimisation* and *Genetic Algorithm*, a detailed study of each of them is presented in section 2.3.

2.2 Disruption Management

Throughout a given plan, it is normal the existence of disruptions in the very same, caused by internal and external factors, leading to a certain deviation from the original plan and potentially affecting its execution. As examples, the addition of new restrictions, the change in system parameters and unpredictable events such as weather changes and terrorist attacks. Thus, any change made in the original plan is called a disruption.

Disruption Management according to [YQ04], can be defined as after a plane (lying close to an optimal solution for the plan, or even being the optimal solution) have been created, either by

optimisation models or using schemes, and during its execution a disruptions occurs. It is possible that the plan moves away from the optimal or even became unfeasible. Leading to a need to revise the original plan, reflecting on changes and restrictions implemented by the subsequent disruption, thus minimising the overall impact originated.

The definition of Disruption Management goals passes through three essential points, being these to carry out the operational plan, minimize costs and return to the plan as soon as possible. In spite of the first two objectives being part of the Disruption Management goals, they are also clearly present in the construction of the original plan, in an attempt to create an optimal plan. Although airlines must manage to balance two types of dimensions (crew and aircraft), it is also imperative that when in the presence of a disruption, it is considered a global view - now considering also the passengers' dimension -, and to fulfilled the operational plan leading them to their final destination at the agreed time, also paying attention to aircraft and tail assignment reducing travel costs. The third objective is common in problem solving and also somehow connected to the other two goals and to all airline resources (as it may compromise the plan outlined for each). [KLL+07]

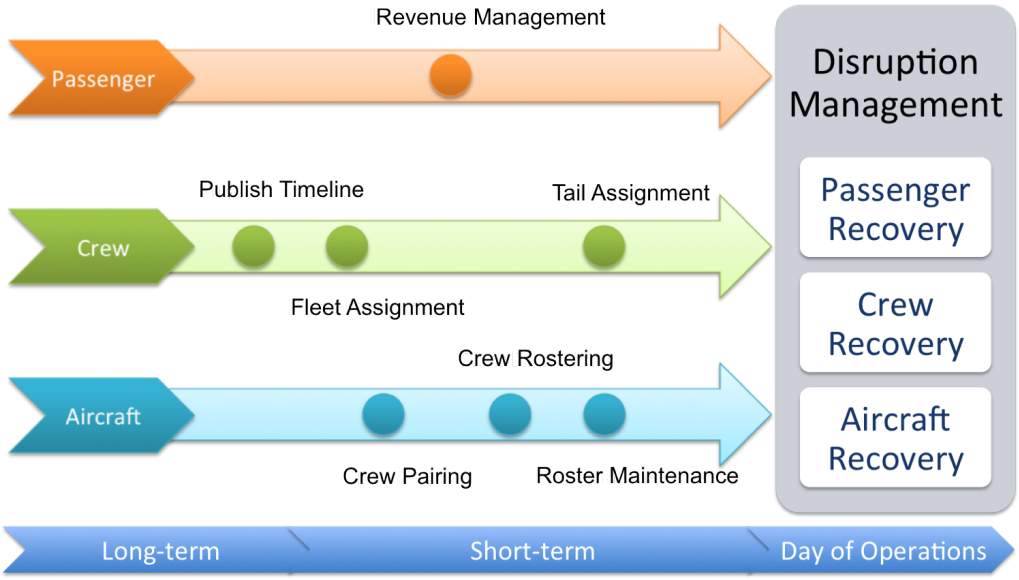

Figure 2.1: Plan creating schema

Disruption management systems face a number of challenges:

- **Timing**, solutions must be generated in a short time when comparing to the time spent at the planning stage.
- **Data**, it is often spread over several databases.
- **Feasibility**, for a solution to be considered as a feasible solution, this one must obey a set of rules, sometimes quite complex.

Alongside with a number of benefits:

- **QoS (Quality of Service)**, improvement of quality of service offered to customers.
- **Resource utilisation**, better use of resources.

However, airlines companies, as they are already aware of possible disruptions they try to anticipate these same events, adding some flexibility in their schedules. This can, incorporate the following methods:

- **Add slack in the plans**, instead of planning everything with close limits of the ideal, there is added extra time to the actions.
- **Crew follow each other and the aircraft**, this method preserves some of the various properties of the original plan resulting in an easier recovery and making monitoring an easier operation.
- **Out and back**, If an aircraft does the flight from a hub to a spoke and back to the same hub, these two flights can be canceled without affecting the rest of the aircraft schedule, the same would happen to the crew if they would do both flights.
- **Stand by crew and aircraft**, an extra airplane and its crew can be valuable but are quite expensive resources that can be used in case of disruption.
- **Increase cruise speed**, airplanes have several cruise speeds depending on the altitude at which they travel. Since cruise travel is defined by the speed at which the aircraft has the best efficiency concerning fuel consumption. Usually planes travel at cruise speed, but this one can be increased, since it is usually lower than the top speed from the same plane, thereby saving time that can lead into higher cost brought by disruptions in the three dimensions planes, crew and passengers.

2.3 Evolutionary Algorithms

Evolutionary algorithms are part of a subset of artificial intelligence, evolutionary computation, where there is a demand for an optimisation of a problem. An evolutionary algorithm can be defined according to the dictionary as "an algorithm which incorporates aspects of natural selection or survival of the fittest. An evolutionary algorithm maintains a population of structures (usually randomly generated initially), that evolves according to rules of selection, recombination, mutation and survival (...)". [ead15]

Fogel, that introduced the concept of evolutionary computation, defined intelligence as "the capability of a system to adapt its behaviour to meet its goals in a range of environment". [Fog06]

The following sections present three classes of evolutionary algorithms, namely *Particle Swarm Optimisation*, *Ant Colony Optimisation* and *Genetic Algorithm*.

2.3.1 Particle Swarm Optimisation

Particle Swarm Optimisation is an evolutionary computing technique introduced by Kennedy (Social Psychologist) and Eberhart (Electrical Engineer) for the first time in 1995 based on a metaphor of social behaviour. The main objective is to create artificial intelligence through the study of social interaction, instead of observing the individual skills. The first simulations performed by Kennedy and Eberhart were influenced by the work of Heppner and Grenander, [ES01].

The Particle Swarm Optimisation was developed by observing shoals and flocks of birds in search of food. It is a search algorithm, which aims to find the global solution of the system and not just the local maximum, despite of recording all the local maxima of each and every of its particles, also the algorithm registers the global maximum. Each particle moves in a certain direction based on their own experience, as well as the experience of the entire group. Comparing to other evolutionary algorithms, the main advantages of Particle Swarm Optimisation are its robustness towards the control parameters and its computational efficiency, [Che09].

Particle Swarm Optimisation can be used across a wide area of problems and the applications are numerous and diverse. Examples of applications are video analysis and image, restructuring and design of electrical networks and cargo shipping; electronics and electromagnetism; power generation systems and power; scheduling; architecture and optimisation of communication networks; biological, medical and pharmaceutical; signal processing; robotics; neural networks; military and security. [PKB07]

Particle Swarm Optimisation is based on a number of particles named entities, which are on the search space of the problem, and each of these is responsible for evaluating its own fitness value and the knowledge of its current position. Through the analysis of some data, such as their best and current fitness, the positions they occupied in each of the previous iterations, as well as facts of other particles and some random perturbations, the particle then calculates the speed with which will go through the search space. With this data, the algorithm proceeds to the next iteration after all the particles have been moved to their next position. It is expected that in the next iteration the set of particles - *swarm* - approaches the optimal solution according to defined fitness function.

Each individual particle is composed of three N-dimensional vectors, where N is the size of the search space of the problem. The three vectors are the current position, the best previous position and the current speed. The current position is defined by a set of coordinates which represent specific details of the problem, over which the search is based on, and in each iteration the set of coordinates makes a solution which is evaluated as a whole. If the new position is better than any found so far, its coordinates and the value generated by the fitness function are stored according to the position, in order to compare with future iterations.

But the algorithm does not depend on only one particle, but from a set of particles. Once a particle alone would not bring any advantages in solving the problem, since the interaction between particles would not exist and thus there was no cooperation among particles. Solving the problem happens with the interaction between the swarm and the analysis of the various individual behaviours, introducing the advantages of group/team work to the problem, mostly known as

cooperation, [PKB07].

However, the interaction between particles follows a particular organisation, which is designated by neighbourhood, that defines the way that two or more particles communicate among themselves. The neighbourhood helps in most cases the algorithm not to get stuck in a local minimum of the fitness function. Examples of neighbourhoods are: single-sighted where each particle communicates only with the following one, ring topology where particles can communicate with the previous and the following one, fully connected topology where each particle has the possibility to communicate with any other particle of the problem and in isolated environments where only a specific number of particles communicates with each other but following the ring topology, [pso15]. Every particle communicates with other particles and they get affected by the best position of a particle that is in its neighbourhood, [PKB07].

```
For each particle {
   Initialise particle
}

While stopping condition is not met {
   For each particle {
      Calculate particle fitness value
      If the new fitness value is better than the personal Best {
         Update personal Best with the new fitness value
      }
      If the personal Best is better than the global Best {
         Update global Best with the personal Best
      }
   }

   For each particle {
      Evaluate particle Velocity
      Use global Best and Velocity to update the particle Data
   }
}
```

Listing 2.1: Particle Swarm Optimisation Pseudocode (Adapted from [pso15])

In the pseudocode listing 2.1, some parameters that are important to its development have not been mentioned. One of them is the swarm size that can vary greatly with the context of a problem, or even with the complexity of its particles. Another parameter with his own importance is the maximum speed that a particle can get, preventing that particles change a high number of parameters in a single iteration, [pso15]. The opposite can also co-exist, i.e., the minimum speed which can help the problem, forcing the particles to change their parameters on each iteration.

2.3.2 Ant Colony Optimisation

Ant Colony Optimisation is a metaheuristic nature inspired, with the main objective of solving Combinatorial Optimisation Problems with a high degree of difficulty. This algorithm was introduced in 1990 by M. Dorigo (Research Director for the FNRS and co-director of IRIDIA), [DB05]. The source of inspiration of Ant Colony Optimisation was the colonies of ants found in nature, specifically by their foraging, i.e., the search and exploration undertaken by animals in search of food resources, [Blu05].

Ants have a very specific way of performing this foraging, leaving a pheromone trail on the ground through which other ants can know what track they must follow in a certain way, this track is a way of communication between them. In analogy with what has been presented so far, the Ant Colony Optimisation algorithm is therefore based on indirect communication within a colony of agents (ants) simple motivated by the pheromone trails. Thus ants use the pheromone trails to build probabilistic solutions of the problem and they adapt it, while the algorithm is running, into something that reflects their experience in finding a solution, [DS10].

The Ant Colony Optimisation algorithm has many possible applications in a lot of areas and can even be considered the top algorithm for various applications. Examples of areas where applications of Ant Colony Optimisation algorithm have good results are: sequential ordering, planning, probabilistic Travel Salesman Problem, DNA sequencing, [DS10].

There are two different methods used to get solutions, one is based on creating the solution itself, starting from a partial solution and constructing in the best way possible in order to create the final solution, the other is a typical local search algorithm, moving through the search space and works on solutions already completed, [DS10].

The construction algorithms defines the solution of the problem, step by step as in an incremental way starting from an initial solution (empty) and going through an iterative process by adding components to the solution without using backtracking until a complete solution is made. As a first case, quite simple, the components to be added to the solution are created using a stochastic process. However, there is a way to find better solutions if a heuristic (greedy construction heuristic) is used to estimate the benefit of adding the component to be added to the same solution.

Greedy construction heuristic goes through a step by step algorithm which adds components that reached an ideal and adds some benefit to the solution calculated using the heuristic, where basically starting from an empty solution, and until the same be a complete solution, is being added components calculated using the same heuristic, and in the end the complete solution is returned. To calculate the component to be added to the solution it is used a heuristic which returns the component with the best heuristic according to the solution which was at least until then a partially solution.

However, a disadvantage of a heuristic running with greedy basis is that often the component which is selected is one out of a relatively small closed set, to the state of the solution at the time the heuristic was performed. Thus and at an early stage, the solution search space starts to be

restricted, as other possible results are taken from the search space, leading to a reduced number of possibilities when the solution is considered to be close to a complete solution.

Moreover, there are local search based algorithms, which depart from the initial (complete) solution, and try to find a better solution in a neighbourhood of the current solution. The algorithm follows an iterative process and demands for a better solution in the neighbourhood. In the event that a better solution is found, the current solution is replaced by the newly found, and the search continues. This process goes on until no new solution is found and thus the algorithm ends at the local maximum (or perhaps the global maximum), [DS10].

The neighbourhood must be defined based on problem structure and what is desired of it. It is also through the defined structure of the neighbourhood that we can access a given set of other solutions when the problem lies over a particular solution through one step in the algorithm. This is an important step in the search for local solutions, it is necessary to define a good neighbourhood and so that the neighbouring solution is to replace the current one.

```
1  Create the heuristic solution
2  Evaluate cost of the solution
3  Initiate pheromones
4
5  While stopping condition is not met {
6     For each ant {
7        Construct the solution
8        Calculate particle fitness value
9        If the new cost value is better than the personal Best {
10          Update personal Best with the new cost value
11       }
12       Update local solution and the pheromones trail
13    }
14    Update global solution and the pheromones trail
15 }
```

Listing 2.2: Ant Colony Optimisation Pseudocode (Adapted from [**?**])

One other option which can be included in the listing 2.2, are the Deamon Actions. These are centralized actions which cannot be implemented be performed by ants themselves. Examples of these are the deposit of additional pheromone in paths, either by adding it to the path of an ant (when in local search) or to another path of the problem which can somehow benefit the solution (when in global solution), [201].

The update of pheromones is meant to share the good solutions components so that they serve as a possible model for future iterations. In order to do this, we can use two mechanisms. The first one involves increasing the level of pheromone of a certain component of a solution, which is associated with a set of good solutions, where the goal is to convert a certain component in a choice that will be given as more certain for ants to choose (it is not imperative that ants will follow, since the algorithm continues to follow stochastic patterns). The second mechanism prevents an overly

conversion to a sub-region of the search space, so the mechanism is implemented in a region where a certain amount of pheromone will be decreasing in the course of time (iterations) the pheromone deposits left by other ants.

2.3.3 Genetic Algorithms

Genetic Algorithms were formally introduced by John Holland in the 70s in the University of Michigan, United States, [**?**]. These are considered meta-heuristic algorithm, a top-level general strategy which guides other heuristics to search for a solution, which are used efficiently in problems of optimisation and search (Goldberg, 1989; Gen and Cheng, 1997; Parmee, 1999), it is also based in nature, specifically with the process of natural selection, involving concepts such as mutation, recombination and selection. Examples of areas of application of the Genetic Algorithms are: benchmark problems, magnetically leviltated vehicles, optimisation of object shaping and circuit layout [**?**].

In the original Holland's algorithm, a parent is selected according to the fitness of each actual being (these are selected through a stochastic decision, and the better the fitness for each being, the greater the chance of being chosen), and in case of recombination, the other parent will be chosen randomly.

However, there are some variations of the algorithm, such as both parents can be chosen based on the fitness value, different probabilities for the existence of mutation or recombination or population size, since the way the initial population is chosen can have a significant impact on results, [**?**].

In order to find the optimal solution in the context of a large Combinatorial Optimisation Problem, genetic algorithms work on a population of N solutions.

```
1  Initiate the first generation
2  Evaluate the population of the first generation
3
4  While stopping condition is not met {
5     Create the next generation
6     Selection of individuals
7     Crossover
8     Mutation
9     Evaluate the population
10 }
```

Listing 2.3: Genetic Algorithms Pseudocode (Adapted from [**?**])

Listing 2.3 presents the pseudocode for Genetic Algorithms, the phases contained here are explained in next paragraphs, particularly the selection of individuals, crossover and mutation.

Selection of individuals, is based on selecting certain chromosomes, through the fitness value assigned by the fitness function and then these will be part of creating the offspring for the next generation. The connection between the algorithm and natural selection takes place here, since the

best are the ones who are more likely to participate in this process. There are several methods of selection of chromosomes, but the most popular is the "roulette-wheel", which is the analogy of the game itself. Once the chromosomes who will take part in the creation of a new generation are chosen, the process can be repeated.

Regarding the use of genetic operations, in classical genetic algorithms usually two are commonly used: crossover and mutation operators, wherein each has a different probability of occurrence, as in nature, in which the probability of each operator is different.

Crossover, the first stage involves selecting pairs of chromosomes that will be the parents of the next generation. This process is done stochastically according to a probability set to crossover. Then, for each pair of parents, it is necessary to decide the crossover point, i.e., the point at which discontinuity exists from the parents information and passes to the other parent to provide the remaining information. To complete this process, two new sprouts must be created, representing the two possible combinations: the first part of the first parent, and the second part of the second parent, leaving with the first part of the second parent and the second portion of the first parent.

Mutation, there is a probability that one or more genes (part of the same chromosome) change its value.

2.4 Summary

Looking into the classical, unorthodox and stochastic ways of both search and optimisation algorithms, there are two different methodologies. In one hand, the classical way which goes as a point-by-point approach through the problem and in each iteration the solution is modified into a different one, hopefully better. In the other hand, using both the unorthodox and stochastic ways, particularly the evolutionary algorithms as stated above are motivated by nature evolutionary principles, which leads the search towards an optimal solution. The difference here is that evolutionary algorithms make use of a population of solutions, and not just one single solution like in the classical algorithms. If there is just one solution, then it is expected to the rest to converge into that solution. If there are multiple solutions, then the algorithm can use multiple optimal solutions as its final solution, [Deb01].

In the next chapter, for both Particle Swarm Optimisation and Ant Colony Optimisation an approach to the problem as this same will be described.

Chapter 3

Aircraft Recovery Problem

In this chapter the problem will be described and the approach to the same will be presented.

3.1 Problem Description

Originally, upon the creation of an initial plan, an aircraft is scheduled to fly a set of routes. A route is a sequence of pairs of airports which are served by an airline, and both departure and arrival times. However, during the operational plan, disruptions may occur, leading to infeasible routes and either an increased cost related to that route. But to produce recovery plans is a complex task, since not only one dimension must be taken into account, and be re-planned. The aircraft recovery process cooperates with the remaining dimension so a feasible solution is found and able to be implemented, also the quality of this solution must be estimated so it can be compared to other possible solution and to the disrupted.

Aircraft Recovery (ARO) needs to provide a solution for this disruption, either by providing a new aircraft to perform the route or delaying the flight until the disruption is settled, [**?**]. The ARO arises when a disruption occurs and its main goal is set to restore or recover the initial plan as much as possible in order to minimize the cost and the delay of both the disrupted flight as the cost of the operation plan, [**?**].

3.2 Approach

In this section, and for both Particle Swarm Optimisation and Ant Colony Optimisation a description of the approach to be used will be provided. For each method or algorithm, the details on how they were implemented and nature mataphors will be complemented with information from the problem itself.

3.2.1 Particle Swarm Optimisation

3.2.2 Ant Colony Optimisation

3.3 Summary

Chapter 4

Experiments and Result

In this chapter there will be presented data collected from the implemented methods, the how it was collected and comparison between methods.

4.1 Experimentation Scenarios

In this section the environment on which the tests were performed will be described, the generation of tests itself and how data will be handled.

The system used to host this tests is MASDIMA, which according to [**?**] is holding data from an operational plan from September 2009 of TAP Portugal, since it has properties similar to the average of one year of operation, and not only it has the data related to the activity presented on the operational schedule as it also includes the data related with operational costs. Furthermore, and still according to [**?**], MASDIMA has data from 49 disruption events, which were randomly selected from the operation plan. Also, 49 flights, 31 aircrafts, 286 crew members and 4.760 passengers were affected by this 49 disruption events.

Although, in particular for this experimentation scenario, only 5 static events will be used, since any of this 5 events was representing a disruption in every dimension.

As for the metrics to measure the results, there will be taken into account three main indicators, the algorithm running time, the number of solutions the specialist sends to its manager and the cost of the solution for the disruption.

4.2 Results

Table 4.1: My caption

Algorithm Running Time (ms)						
	928	1917	864	1614	839	Average SD
HC	770,8	532,1	865,2	87,9	38,5	65,0
SA	77,8	69,8	76,7	72,8	69,9	7,6
PSO	168,9	151,7	157,6	87,7	59,7	17,1
ACO	1555,4	884,4	1883,0	227,0	327,3	80,5

Table 4.2: My caption

Number of Solutions					
	928	1917	864	1614	839
HC	111,0	81,0	133,3	12,6	6,0
SA	10,0	10,0	10,1	10,2	10,0
PSO	7,7	7,0	7,9	4,4	3,6
ACO	4,3	4,0	4,6	2,8	4,0

Table 4.3: My caption

Aircraft Cost						
	928	1917	864	1614	839	Average SD
HC	3160,2	3151,0	3122,7	6612,0	9127,3	1,5
SA	3728,8	3650,8	4052,6	8235,0	9127,2	990,2
PSO	3437,7	3696,7	3778,6	7771,3	9126,0	853,9
ACO	3560,8	3792,5	4054,8	9627,8	9130,0	1156,1

4.3 Summary

Proin vehicula pharetra urna. Aliquam egestas nunc quis nisl. Donec ullamcorper. Nulla purus. Ut suscipit lacus vitae dui. Mauris semper. Ut eget sem. Integer orci. Nam vitae dui eget nisi placerat convallis.

Chapter 5

Conclusion

Deve ser apresentado um resumo do trabalho realizado e apreciada a satisfação dos objetivos do trabalho, uma lista de contribuições principais do trabalho e as direções para trabalho futuro.

A escrita deste capítulo deve ser orientada para a total compreensão do trabalho, tendo em atenção que, depois de ler o Resumo e a Introdução, a maioria dos leitores passará à leitura deste capítulo de conclusões e recomendações para trabalho futuro.

5.1 Objective Fulfilment

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam non felis sed odio rutrum ultrices. Donec tempor dolor. Vivamus justo neque, tempus id, ullamcorper in, pharetra non, tellus. Praesent eu orci eu dolor congue gravida. Sed eu est. Donec pulvinar, lectus et eleifend volutpat, diam sapien sollicitudin arcu, a sagittis libero neque et dolor. Nam ligula. Cras tincidunt lectus quis nunc. Cras tincidunt congue turpis. Nulla pede velit, sagittis a, faucibus vitae, porttitor nec, ante. Nulla ut arcu. Cras eu augue at ipsum feugiat hendrerit. Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus quam lacus, pharetra vel, aliquam vel, volutpat sed, nisl.

5.2 Future Work

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam tempor tristique risus. Suspendisse potenti. Fusce id eros. In eu enim. Praesent commodo leo. Nullam augue. Pellentesque tellus. Integer pulvinar purus a dui convallis consectetuer. In adipiscing, orci vitae lacinia semper, sapien elit posuere sem, ac euismod ipsum elit tempus urna. Aliquam erat volutpat. Nullam suscipit augue sed felis. Phasellus faucibus accumsan est.

Conclusion

Appendix A

Loren Ipsum

Depois das conclusões e antes das referências bibliográficas, apresenta-se neste anexo numerado o texto usado para preencher a dissertação.

A.1 O que é o *Loren Ipsum*?

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum [**?**].

A.2 De onde Vem o Loren?

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet...", comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.

A.3 Porque se usa o Loren?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using "Content here, content here", making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for "lorem ipsum" will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

A.4 Onde se Podem Encontrar Exemplos?

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.

References

[201] 2015. Metaheuristics network. http://www.metaheuristics.net/index.php?main=3. Accessed: 2015-02-09.

[AENA04] Ahmed Abdelghany, Goutham Ekollu, Ram Narasimhan, and Khaled Abdelghany. A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, 127(1-4):309—310, 2004.

[AFS15] Airline performance reports. http://www.flightstats.com/go/Stats/airlinePerformanceReports.do, 2015. Accessed: 2015-02-06.

[Blu05] Christian Blum. Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, 2(4):353–373, 2005.

[CCZ10] Xindu Chen, Xin Chen, and Xinhui Zhang. Crew scheduling models in airline disruption management. In *Industrial Engineering and Engineering Management (IE&EM), 2010 IEEE 17Th International Conference on*, pages 1032–1037. IEEE, 2010.

[CDF15] Flightstats global cancellation and delays. http://www.flightstats.com/go/Media/stats.do, 2015. Accessed: 2015-02-06.

[Che09] Po-Hung Chen. *Particle swarm optimization for power dispatch with pumped hydro*. INTECH Open Access Publisher, 2009.

[CTA04] Andrew J Cook, Graham Tanner, and Stephen Anderson. Evaluating the true cost to airlines of one minute of airborne or ground delay: final report. page 100, 2004.

[DB05] Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2):243–278, 2005.

[Deb01] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[DS10] Marco Dorigo and Thomas Stützle. Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics*, pages 227–263. Springer, 2010.

[ead15] Evolutionary algorithm | define evolutionary algorithm at dictionary.com. http://dictionary.reference.com/browse/evolutionary+algorithm, 2015. Accessed: 2015-02-08.

[ES01] Russell C Eberhart and Yuhui Shi. Particle swarm optimization: developments, applications and resources. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 81–86. IEEE, 2001.

REFERENCES

[Fog06] David B Fogel. *Evolutionary computation: toward a new philosophy of machine intelligence*, volume 1. John Wiley & Sons, 2006.

[Irr96] M. E. Irrang. Crew scheduling models in airline disruption management. In *Airline Irregular Operations*, pages 349–365, 1996.

[KLL+07] Niklas Kohl, Allan Larsen, Jesper Larsen, Alex Ross, and Sergey Tiourine. Airline disruption management—perspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3):149–162, 2007.

[MAS12] Masdima - multi-agent system for disruption management. http://www.disruptionmanagement.com/MASDIMA_Project.html, 2012. Accessed: 2015-02-12.

[MAS15] Agifors airline operations 2015 - abu dhabi, uae. http://www.agifors.org/studygrp/opsctl/2015/program.html, 2015. Accessed: 2015-06-02.

[PKB07] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[pso15] Introduction to particle swarm optimization. http://mnemstudio.org/particle-swarm-introduction.htm, 2015. Accessed: 2015-02-09.

[YQ04] Gang Yu and Xiangtong Qi. *Disruption management: framework, models and applications*. World Scientific Publishing Company Incorporated, 2004.