FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Evolutionary Computation methods applied to Operational Control Centers

António José Ferreira de Castro Moura

PREPARAÇÃO DA DISSERTAÇÃO

U. PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Ana Paula Rocha

Second Supervisor: António Castro

February 15, 2015

Evolutionary Computation methods applied to Operational Control Centers

António José Ferreira de Castro Moura

Mestrado Integrado em Engenharia Informática e Computação

February 15, 2015

Abstract

During the execution of any operational plan, there is the likelihood of plan being affected by some disruptions caused by some internal or external factors, also disruptions may affect three dimensions that Airline companies and the Operational Control Centers must take into account during planning witch 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. There are some good and simple examples of disruption that may occur like changes in weather, threats or even terrorist attacks, and delays in aircraft maintenance.

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. [YQ04]

Future work will be focused on implementing three agents 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. Also, these agents will be implemented on a Multi-Agent System named MASDIMA that represents an Operation Control Center. [Cas13] [CRO14]

Resumo

Durante a execução de qualquer plano operacional, existe a possibilidade do mesmo sofrer rupturas causadas por fatores quer internos ou externos, além do mais as rupturas podem afetar três dimensões sobre as quais as companhias Aéreas e os Centros de Controlo Operacional devem ter em conta aquando do planeamento sendo estas os passageiros, a tripulação e aviões. Normalmente, um ruptura é um estado no qual durante o qual uma operação que esteja a ser executada sobre um desvio (a qual é grande o suficiente para causar uma mudança) do plano original, e por vezes leva mesmo a que o plano não seja de possível execução. Existem alguns exemplos bons e simples de rupturas como mudanças meteorológicas, ameaças ou mesmo ataques terroristas e atrasos na manutenção dos 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. [YQ04]

O trabalho futuro passa por implementar 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. Além do mais, esses três agentes serão implementados num Sistema Multi-Agente chamado de MASDIMA que representará o Centro de Controlo Operacional. [Cas13] [CRO14]

Acknowledgements

Aliquam id dui. Nulla facilisi. Nullam ligula nunc, viverra a, iaculis at, faucibus quis, sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Curabitur magna ligula, ornare luctus, aliquam non, aliquet at, tortor. Donec iaculis nulla sed eros. Sed felis. Nam lobortis libero. Pellentesque odio. Suspendisse potenti. Morbi imperdiet rhoncus magna. Morbi vestibulum interdum turpis. Pellentesque varius. Morbi nulla urna, euismod in, molestie ac, placerat in, orci.

Ut convallis. Suspendisse luctus pharetra sem. Sed sit amet mi in diam luctus suscipit. Nulla facilisi. Integer commodo, turpis et semper auctor, nisl ligula vestibulum erat, sed tempor lacus nibh at turpis. Quisque vestibulum pulvinar justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam sed tellus vel tortor hendrerit pulvinar. Phasellus eleifend, augue at mattis tincidunt, lorem lorem sodales arcu, id volutpat risus est id neque. Phasellus egestas ante. Nam porttitor justo sit amet urna. Suspendisse ligula nunc, mollis ac, elementum non, venenatis ut, mauris. Mauris augue risus, tempus scelerisque, rutrum quis, hendrerit at, nunc. Nulla posuere porta orci. Nulla dui.

Fusce gravida placerat sem. Aenean ipsum diam, pharetra vitae, ornare et, semper sit amet, nibh. Nam id tellus. Etiam ultrices. Praesent gravida. Aliquam nec sapien. Morbi sagittis vulputate dolor. Donec sapien lorem, laoreet egestas, pellentesque euismod, porta at, sapien. Integer vitae lacus id dui convallis blandit. Mauris non sem. Integer in velit eget lorem scelerisque vehicula. Etiam tincidunt turpis ac nunc. Pellentesque a justo. Mauris faucibus quam id eros. Cras pharetra. Fusce rutrum vulputate lorem. Cras pretium magna in nisl. Integer ornare dui non pede.

The Name of the Author

“You should be glad that bridge fell down.
I was planning to build thirteen more to that same design”

Isambard Kingdom Brunel

Contents

CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

ADT	Abstract Data Type
ANDF	Architecture-Neutral Distribution Format
API	Application Programming Interface
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CORBA	Common Object Request Broker Architecture
UNCOL	UNiversal COmpiler-oriented Language
Loren	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Sed vehicula lorem commodo dui
WWW	*World Wide Web*

Chapter 1

Introduction

O primeiro capítulo da dissertação deve servir para apresentar o enquadramento e a motivação do trabalho e para identificar e definir os problemas que a dissertação aborda. Deve resumir as metodologias utilizadas no trabalho e termina apresentando um breve resumo de cada um dos capítulos posteriores.

Este documento ilustra o formato a usar em dissertações na Faculdade de Engenharia da Universidade do Porto, não servindo de exemplo sobre os conteúdos a usar. São dados exemplos de margens, cabeçalhos, títulos, paginação, estilos de índices, etc. São ainda dados exemplos de formatação de citações, figuras e tabelas, equações, referências cruzadas, lista de referências e índices.

Uma recolha de normas existentes sobre este assunto pode ser encontrada em [**?**].

> "Like the Abstract, the Introduction should be written to engage the interest of the reader. It should also give the reader an idea of how the dissertation is structured, and in doing so, define the thread of the contents." [**?**, chap. Introduction]

Neste primeiro capítulo ilustra-se a utilização de citações e de referências bibliográficas. Para além de dar um exemplo de utilização de uma citação, a citação anterior, introduz uma referência que pode ser consultada, entre muitas outras referências bibliográficas interessantes [**?**, **?**].

1.1 Context

Esta secção descreve a área em que o trabalho se insere, podendo referir um eventual projeto de que faz parte e apresentar uma breve descrição da empresa onde o trabalho decorreu.

1.1.1 MASDIMA

1.2 Motivation and Objectives

Nowadays, one of the problems that airline industries are facing happens during the day of operations and refers to irregular operations sources which can erupt and jeopardise the operational plan. Examples of this source can be an adverse change in the weather conditions, unexpected aircraft breakdown leading to a longer maintenance, sick crew and so on. Also if a proper recovery plan is not taken in a short time, this very same 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 same flight. Some measures are being taken into account to prevent this events, before it was usual for different crew members to proceed to a different flight, now the same crew may do the same flights. So a proper recovery plan must be taken into account, since the use of expensive recovery actions like ferrying an aircraft, rebook passengers on other airlines, hire a new crew for another flight increase the operation cost of the plan and reduces the expected revenues the airline company had with the flight or plan.[AENA04]

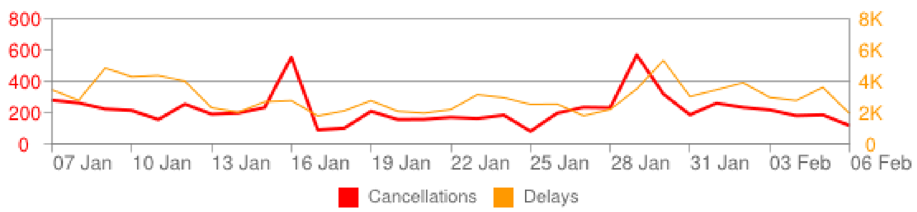

Figure 1.1: European Canceled and Delayed Flights
[flia]

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 and efficient use of resources, leading to better revenues.[CCZ10]

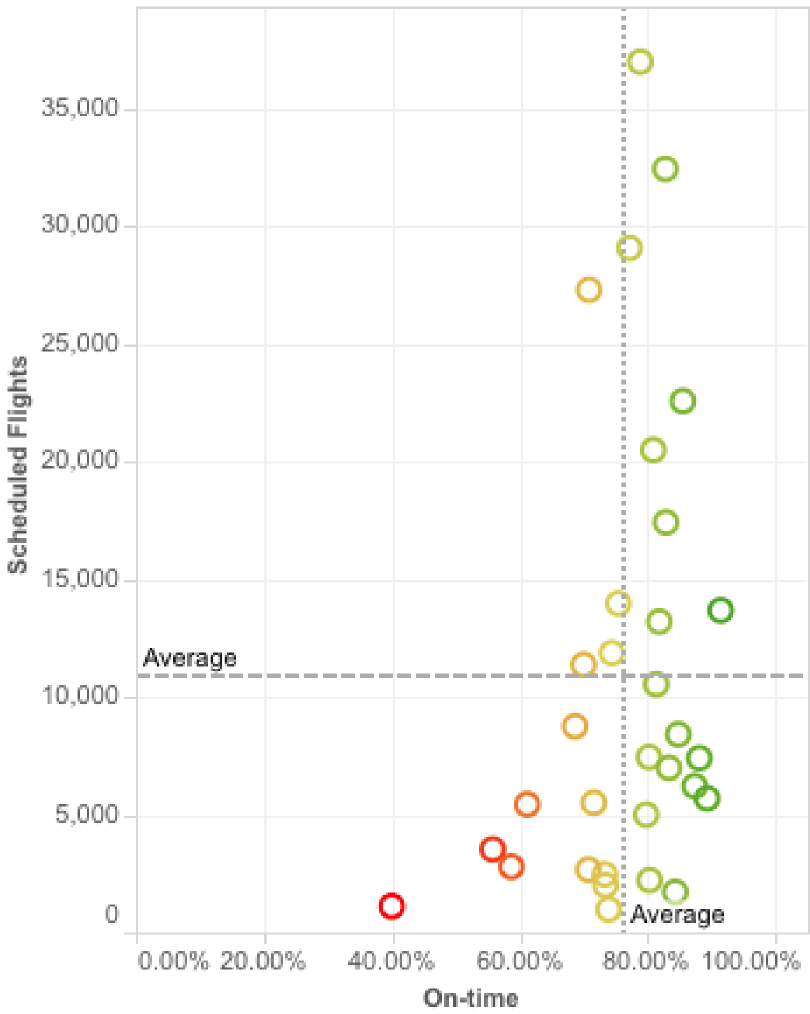

Figure 1.2: Major Airlines from Europe - Arrival Performance, December 2014 [flib]

Despite of, the average delay among the major Airlines from Europe being 42.05 minutes, only 16% of the flights arrive with delay. Also, and 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 irregular operations can cost up to 3% of the airline annual revenue [CCZ10]. Also it was estimated that a better recovery process could result in cost reductions of at least 20% with the cost of irregular operations [Irr96].

1.3 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais x capítulos. No capítulo 2, é descrito o estado da arte e são apresentados trabalhos relacionados. No capítulo **??**, ipsum dolor sit amet, consectetuer adipiscing elit. No capítulo 3 praesent sit amet sem. No capítulo 4 posuere, ante non tristique consectetuer, dui elit scelerisque augue, eu vehicula nibh nisi ac est.

Chapter 2

Revisão Bibliográfica

Neste capítulo é descrito o estado da arte e são apresentados trabalhos relacionados para mostrar o que existe no mesmo domínio e quais os problemas em aberto. Deve deixar claro que existe uma oportunidade de desenvolvimento que cobre alguma falha concreta .

O capítulo deve também efetuar uma revisão tecnológica às principais ferramentas utilizáveis no âmbito do projeto, justificando futuras escolhas.

2.1 Introdução

Neste capítulo é ilustrada a utilização de macros LaTeX para definir entradas no índice remissivo e são feitas diversas referências bibliográficas, usando-se texto de um artigo apresentado na Conferência XATA2006 [**?**].

Nos últimos tempos têm surgido diversas soluções, apresentadas por empresas do sector Automação de Sistemas para a disponibilização de sistemas *SCADA/DMS* na *Web*.

Aliquam sollicitudin facilisis sapien. Mauris tincidunt tristique diam. Mauris sollicitudin pede at tellus varius volutpat. Integer vel leo. Nunc massa diam, egestas eu, venenatis at, porttitor ac, sapien. Sed magna elit, vulputate in, lacinia sed, lobortis ac, urna. Proin cursus massa id risus. Vestibulum libero. Curabitur venenatis augue. Mauris eu libero eget lectus tempus tempor. In tincidunt, justo in varius adipiscing, ipsum enim gravida massa, eget ornare ante lacus id est. Praesent vitae est ut elit convallis convallis. Aenean tincidunt, purus id consectetur volutpat, sem leo pulvinar libero, nec semper sem purus ultricies nibh [**?**].

Fusce risus mi, tristique eu, consectetuer id, auctor sed, elit. Donec laoreet. Duis consectetuer interdum libero. Etiam eu orci. In eu arcu. Fusce luctus diam eget lectus. Duis interdum lacus sed ligula. Proin vestibulum felis eget lacus. Vivamus vestibulum, tellus ut congue viverra, mauris lacus tempor turpis, eu congue nisi magna at dolor. Ut molestie vehicula libero. Praesent in neque sed risus tempus ornare. Donec hendrerit, erat eu semper aliquam, pede nulla dapibus risus, ut pretium orci pede et neque. Etiam eget tortor a metus convallis viverra. Quisque eget nisi sed orci facilisis interdum. Aliquam non felis.

2.2 Secção Exemplo

Scalable Vector Graphics é uma linguagem em formato XML que descreve gráficos de duas dimensões. Este formato padronizado pela W3C (*World Wide Web Consortium*) é livre de patentes ou direitos de autor e está totalmente documentado, à semelhança de outros W3C standards [**?**].

Sendo uma linguagem XML, o *SVG* herda uma série de vantagens: a possibilidade de transformar *SVG* usando técnicas como XSLT, de embeber *SVG* em qualquer documento XML usando *namespaces* ou até de estilizar *SVG* recorrendo a CSS (*Cascade Style Sheets*). De uma forma geral, pode dizer-se que *SVG*s interagem bem com as atuais tecnologias ligadas ao XML e à Web, tal como referido em [**?**, **?**].

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Donec a eros. Phasellus non nulla non massa venenatis convallis. In porta. Mauris quis magna. Proin mauris eros, aliquet id, eleifend vitae, semper quis, erat. Aliquam id lectus non odio dignissim blandit. Vestibulum porttitor arcu ut ligula. Nunc quis erat. Curabitur ipsum tortor, ornare vitae, dapibus pretium, hendrerit sed, urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Phasellus bibendum, nulla eget varius aliquam, tortor nulla sollicitudin quam, vel vestibulum nisl magna at sem. Aliquam velit sapien, ultrices viverra, tempus quis, ultrices at, dui. Aliquam sit amet justo.

Quisque tristique, metus eu iaculis sagittis, urna leo bibendum diam, a ultricies sem diam a augue. Mauris consectetuer, libero vel euismod tincidunt, nisi metus viverra ante, quis pretium sapien odio nec risus. Nunc semper auctor nulla[1].

2.2.1 Subsecção Exemplo

Batik é um conjunto de bibliotecas baseadas em *Java* que permitem o uso de imagens *SVG* (visualização, geração ou manipulação) em aplicações ou *applets* [**?**]. O projeto Batik destina-se a fornecer ao programador alguns módulos que permitem desenvolver soluções especificas usando *SVG* [**?**].

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc eu nulla. Pellentesque vitae nibh ultrices quam iaculis convallis. Aliquam purus eros, varius eget, volutpat sodales, imperdiet nec, lacus. Curabitur in elit sed sem rutrum posuere. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Duis sem. Praesent ultricies odio vel sapien. Integer faucibus malesuada libero. Cras semper, dolor id ullamcorper varius, magna risus volutpat felis, id pellentesque nulla ante at erat. Integer sodales.

Quisque sit amet odio. In at risus sit amet turpis interdum posuere. Maecenas iaculis vehicula sem. Ut leo arcu, malesuada vel, imperdiet id, dignissim a, purus. Duis eleifend, lectus non venenatis dignissim, risus libero imperdiet mi, nec gravida massa libero sed mauris. Nullam lobortis

[1] Exemplo de nota de rodapé.

libero non sapien. Integer convallis iaculis erat. Morbi dictum. Ut ultrices pellentesque velit. Cras ac ante. Etiam in neque tincidunt lacus gravida vehicula. Proin et nisi.

2.2.2 Subsecção Exemplo

Loren ipsum dolor sit amet, consectetuer adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetuer, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

Loren ipsum dolor sit amet, consectetuer adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetuer, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed, luctus in.

2.3 Resumo ou Conclusões

No final do capítulo deverá ser apresentado um resumo com as principais conclusões que se podem tirar.

Vivamus non nunc nec risus tempor varius. Quisque bibendum mi at dolor. Aliquam consectetuer condimentum risus. Aliquam luctus pulvinar sem. Duis aliquam, urna et vulputate tristique, dui elit aliquet nibh, vel dignissim magna turpis id sapien. Duis commodo sem id quam. Phasellus dolor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Chapter 3

Implementation

In this chapter, there will be presented evolutionary algorithms a subset of evolutionary computations, as the algorithms to be implemented an their pseudocodes. Evolutionary computation takes advantage of the optimisation while seeking for solutions. This one also uses a pattern of common procedures between different classes, through the creation of a population of beings and enabling combinations and/or mutations between them in order to create somehow different beings that might be closer to ideal one, also and in order to have a chain of thought, the best of each generation is selected and will act as their leader, leading other to approach him in a certain way, this happens often motivated by evolutions seen in nature.

In order to understand how classes *Particle Swarm Optimisation*, *Ant Colony Optimisation* and *Genetic Algorithm* work, we performed a detailed study of each of them.

3.1 Evolutionary Algorithms

Evolutionary algorithms are part of a subset of artificial intelligence, evolutionary computation, where there is an optimisation to the demand in one problem of artificial intelligence. Evolutionary algorithms represents a term which needs a lot of context, but 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 (...)". [ead] As part of this project, the field of study will be artificial intelligence, specifically the evolutionary computation.

Fogel, the man who made the introduction to evolutionary programming, defined intelligence as "the capability of a system to adapt its behaviour to meet its goals in a range of environment". [Fog06]

3.1.1 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 would be to create artificial intelligence through the study of social interaction analogies, 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 the global solution of the system, not just settling to the local maximum, despite of recording all the local maxima of each and every one of its particles, also has the global maximum registration. Thus, each particle moves in a certain direction based on their own experience, 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]

Also it can be used across a wide area of problems and the applications are numerous and diverse. Among which we can associate applications such as video analysis and image, restructuring and design of electrical networks and cargo shipping; control; 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]

3.1.2 Ant Colony Optimisation

Ant Colony Optimisation is a metaheuristic nature inspired, with the main objective the resolution of Combinatorial Optimisation Problems with a high degree of difficulty. This very same 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]

However, 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 somehow the same track is a way of communication between them. In analogy with what has been presented so far, the Ant Colony Optimisation is therefore based on indirect communication within a colony 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 your experience in finding a solution. [DS10] There are two different methods of construction and local search, which are somewhat different since the construction work on partial algorithms try to extend these solutions and the best possible way to complete solution to the problem, since otherwise the local search algorithm, moves in the search space and works on already complete solutions. [DS10]

The Ant Colony Optimisation has many possible applications in a lot of areas and can even be considered the top algorithm for various applications. Next are some examples of areas where applications have good results such as sequential ordering, planning, probabilistic TSP, DNA sequencing. [DS10]

3.1.3 Genetic Algorithms

The Genetic Algorithms were formally introduced by John Holland in the 70s in the University of Michigan, United States. [gai] These are considered a meta-heuristic, which are used efficiently to 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.

3.2 Pseudocode

In this section, for each of this evolutionary algorithms there will be presented its pseudocode and also a short analysis of the algorithm itself.

“The goal of writing pseudocode, then, is to provide a high-level description of an algorithm which facilitates analysis and eventual coding (...) but at the same time suppresses many of the details that vanish with asymptotic notation.” [pse]

3.2.1 Particle Swarm Optimisation

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, 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 these data, the algorithm proceeds to the next iteration after all the particles have been moved to their next position. Further on 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. And these 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 be eliminated of the problem equation. 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. [PKB07]

However, the interaction between particles follows a particular organisation, the same on which the swarm is designated by neighbourhood, which 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 for instance, 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. [pso] To sum up, 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 3.1: Particle Swarm Optimisation Pseudocode (Adapted from [pso])

However, in the pseudocode Listing 3.1, some parameters have not been mentioned that are important to its development. 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 the particles changes a high number of parameters in a single iteration. [pso] Since 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.

3.2.2 Ant Colony Optimisation

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 you use a heuristic (greedy construction heuristic) to estimate the benefit of adding the component to be added to the same solution.

The construction of the 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. Again, in the first case, 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.

```
Create the heuristic solution
Evaluate cost of the solution
Initiate pheromones

While stopping condition is not met {
   For each ant {
      Construct the solution
      Calculate particle fitness value
      If the new cost value is better than the personal Best {
         Update personal Best with the new cost value
```

```
      }
      Update local solution and the pheromones trail
   }
   Update global solution and the pheromones trail
}
```

Listing 3.2: Ant Colony Optimisation Pseudocode (Adapted from [aco])

The use of local search is a common example of Daemon Actions, which are used to implement isolated actions which can not be carried out individually by ants. Another example Daemon Actions is the use of global information, which is useful for adding extra pheromone to paths, and thus do not look at the problem in a local way. [dea]

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 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, for such a mechanism is implemented where a certain amount of pheromone decreases in the course of time (iterations) the pheromone deposits left by other ants.

3.2.3 Genetic Algorithms

In the original Holland's algorithm, a parent is selected according to the fitness of each (selecting through a system of probabilities, and the better your fitness greater the chance of being chosen), and in case of a recombination, the other parent is 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. [Ree95]

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, for which the fitness value must be calculated.

```
Initiate the first generation
Evaluate the population of the first generation

While stopping condition is not met {
   Create the next generation
   Selection of individuals
   Crossover
   Mutation
```

```
    Evaluate the population
}
```

Listing 3.3: Genetic Algorithms Pseudocode (Adapted from [gap])

Up next, some of new concepts presented in the Listing 3.3 will be explained, 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 there are used as a rule two of them, 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, and 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 same (remaining information). To complete this process, there must be created two new sprouts, 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.

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

3.3 Conclusion

Looking into the classical and the unorthodox and stochastic ways of both search and optimisation algorithms, there are two different methodologies. In one hand, the classical way goes as point-by-point approach through the problem in which iteration a solution is modified to a different one and hopefully better and the outcome would be a single solution. In the other hand, with the unorthodox and stochastic way, particularly the evolutionary algorithms as stated above are motivated by nature's 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 that the rest to converge into that solution, else if there are multiple solutions, then the algorithm can use multiple optimal solutions as its final solution. [Deb01]

Chapter 4

Conclusões e Trabalho Futuro

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.

4.1 Satisfação dos Objetivos

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.

Nullam erat est, vehicula id, tempor non, scelerisque at, tellus. Pellentesque tincidunt, ante vehicula bibendum adipiscing, lorem augue tempor felis, in dictum massa justo sed metus. Suspendisse placerat, mi eget molestie sodales, tortor ante interdum dui, ac sagittis est pede et lacus. Duis sapien. Nam ornare turpis et magna. Etiam adipiscing adipiscing ipsum. Fusce sodales nisl a arcu. Cras massa leo, vehicula facilisis, commodo a, molestie faucibus, metus. Suspendisse potenti. Duis sagittis. Donec porta. Sed urna. Maecenas eros. Vivamus erat ligula, pharetra sit amet, bibendum et, fermentum sed, dolor. Nullam eleifend condimentum nibh. Integer leo nibh, consequat eget, mollis et, sagittis ac, felis. Duis viverra pede in pede. Phasellus molestie placerat leo. Praesent at tellus a augue congue molestie. Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

4.2 Trabalho Futuro

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.

Aliquam felis justo, facilisis sit amet, bibendum ut, tempus ac, dolor. Sed malesuada. Nunc non massa. In erat. Nulla facilisi. Phasellus blandit, est in accumsan cursus, libero augue elementum leo, vitae auctor mauris nisl ac tortor. Cras porttitor ornare elit. Fusce at lorem. Sed lectus tortor, vestibulum id, varius a, condimentum nec, lectus. Maecenas in nisi et magna pretium aliquam. Pellentesque justo elit, feugiat nec, tincidunt a, dignissim vel, ipsum. Sed nunc. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam tempus rhoncus leo. Donec neque quam, cursus sit amet, ultricies varius, semper non, pede. Donec porttitor. Sed aliquet feugiat elit.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus tellus pede, auctor ut, tincidunt a, consectetuer in, felis. Mauris quis dolor et neque accumsan pellentesque. Donec dui magna, scelerisque mattis, sagittis nec, porta quis, nulla. Vivamus quis nisl. Etiam vitae nisl in diam vehicula viverra. Sed sollicitudin scelerisque est. Nunc dapibus. Sed urna. Nulla gravida. Praesent faucibus, risus ac lobortis dignissim, est tortor laoreet mauris, dictum pellentesque nunc orci tincidunt tellus. Nullam pulvinar, leo sed vestibulum euismod, ante ligula elementum pede, sit amet dapibus lacus tortor ac nisl. Morbi libero. Integer sed dolor ac lectus commodo iaculis. Donec ut odio.

References

[aco]

[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.

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

[Cas13] António JM Castro. A distributed approach to integrated and dynamic disruption management in airline operations control. 2013.

[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.

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

[CRO14] Antonio JM Castro, Ana Paula Rocha, and Eugénio Oliveira. *A new approach for disruption management in airline operations control*. Springer, 2014.

[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.

[dea]

[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.

[ead]

[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.

[flia]

[flib]

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

[gai]

[gap]

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

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

[pse]

[pso]

[Ree95] Colin R Reeves. A genetic algorithm for flowshop sequencing. *Computers & operations research*, 22(1):5–13, 1995.

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

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.