



**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**Área Departamental de Engenharia de  
Electrónica e Telecomunicações e de Computadores**



## **GuideMe - Service for Consulting, Publication and Recommendation of Touristic Locations**

**ARTEM UMANETS**

(Licenciado em Engenharia Informática e de Computadores)

Trabalho de projeto realizado para obtenção do grau  
de Mestre em Engenharia Informática e de Computadores

Orientadores:

Mestre Artur Jorge Ferreira

Mestre Nuno Miguel da Costa de Sousa Leite

**Novembro de 2014**





**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**Área Departamental de Engenharia de  
Electrónica e Telecomunicações e de Computadores**



## **GuideMe - Service for Consulting, Publication and Recommendation of Touristic Locations**

**ARTEM UMANETS**

(Licenciado em Engenharia Informática e de Computadores)

Trabalho de projeto realizado para obtenção do grau  
de Mestre em Engenharia Informática e de Computadores

Orientadores:

Mestre Artur Jorge Ferreira

Mestre Nuno Miguel da Costa de Sousa Leite

**Novembro de 2014**



---

## **Acknowledgments**



---

## Resumo

## Palavras Chave





---

## **Abstract**

## **Keywords**



---

## Contents

<b>Acknowledgments</b> .....	v
<b>Resumo</b> .....	vii
<b>Abstract</b> .....	xi
<b>List of Figures</b> .....	xv
<b>List of Tables</b> .....	xvii
<b>List of Code Listings</b> .....	xix
<b>List of Acronyms</b> .....	xix
<b>1 Introduction</b> .....	1
1.1 Examination timetable: State of the Art .....	2
1.1.1 Timetabling Problem .....	2
1.1.2 Solution approaches .....	2



---

## List of Figures



---

## List of Tables





---

## List of Code Listings



## Introduction

In this world, most of the people think that Artificial Intelligence was created to simulate human behavior and the way we humans think. Even though people are not wrong, Artificial Intelligence was also created to solve problems that humans are unable to solve, or to solve it in a shorter amount of time, with a better solution. For example, any exhaustive search requires pretty complex algorithms (heuristics, meta-heuristics etc) with complex mathematical functions in order to find a feasible solution. Problems humans may take days to find a feasible solution, or may not find a solution at all that fits its needs, algorithms like these may deliver a very good solution in minutes, hours or days, depending on how much time the human is willing to use in order to get a better solution and not just a feasible one.

A concrete example is the creation of timetables for scheduling. Timetables can be used for educational purposes, sports scheduling, transportation timetabling etc. It may look like it's simple to create a timetable, but the truth is, this process normally requires following some rules in order to fit its purpose. These rules are called constraints in the timetabling subject, in which the more it has the harder it gets to create a feasible solution for the problem. In some cases, there are so many constraints in which some can be contradictory to each other, that it is needed to "relax the constraints" in order to solve the problem.

The process of creating a timetable requires that the final solution follows a set of constraints. These can be divided in two groups: hard constraints and soft constraints. Hard constraints are a set of rules which must be followed in order to get a feasible solution. In the other hand, soft constraints may be called "optional" since there is no need to follow these to get a solution, because they are not mandatory. The quality of a solution is calculated by using these soft constraints. In another words: the more soft constraints a solution follows, the better it is. This quality is based on the points (weight) of the soft constraints that that solution didn't follow, so the more points, the worse the solution. The weight of each soft constraint is set by the one that created the constraints.

Now that I talked a little bit about timetabling, let me specify the aim of this project. Its main is to create an examination timetable generator using ITC-2007 specifications. The solutions will be validated using a validator also created by me which specifies the quality of the solution and may do some corrections on the timetable in order to get an even better solution. It is also required that the final product may work with two test seasons which

can be considered an extension to ITC-2007 formulation. In the end an (optional) Graphical User Interface will be created in order to allow the user to edit the current solution to fit the users needs and allow optimization to the edited solution. The generator will be tested using data from ITC-2007 and some actual data from six different programs presented in my university ISEL.

## **1.1 Examination timetable: State of the Art**

In this topic we'll be writing about the state of art concerning examination timetable. Concretely, we'll be writing about why timetabling is a rather complex problem, some possible approaches on trying to solve a problem this type and some of the solutions already taken, specifically in ITC 2007.

### **1.1.1 Timetabling Problem**

Timetabling is a subject that has been under investigation for more than ten years. There is no optimal solution for creating perfect timetables. Creating timetables is a process that requires complex algorithms in which search for solutions following a set of constraints, as mentioned above. These constraints can be divided into five main classes named Unary, Binary, Capacity, Event Spread and Agent constraints (REF: 2008 Rhydian Lewis)

Timetabling problem may be formulated as a search or optimization problem (REF: 1999 Schaerf). Search problems consists about finding a solution that satisfies all the hard constraints, which soft constraints aren't the main goal, on contrary optimization problems are problems which tries to satisfy as most soft constraints as possible after satisfying all hard constraints. Optimization problems though are more commonly used to optimize feasible solutions obtained by using a search algorithm. The main goal consists in searching for a feasible solution which satisfies all hard constraints using a search algorithm. If the user's goal is to find the best solution possible in given time, a optimization solution may be applied to the solution given by the search algorithm.

Search problems are quoted as NP-Complete (REF: NP-Complete wiki) and optimization problems are quoted as NP-Hard (REF: NP-Hard wiki) problem. Optimization problems are labeled NP-Hard because any optimization problem can be reduced to a graph coloring problem which is also NP-Hard. Graph coloring will be briefly explained later.

### **1.1.2 Solution approaches**