UNIVERSITAT POLITÈCNICA DE CATALUNYA

DELIVERABLE 5: FINAL MODULE OF THE SPECIALITY
COMPUTER SCIENCE

Design of an environment for solving pseudo-boolean optimization problems

Author:
Marc BENEDÍ

Supervisor:
Dr. Jordi CORTADELLA

GEP

April 9, 2018
Edinburgh, UK

Contents

Chapter 1

Objectives

This project is an extension of a previous C++ library. The main goal of this project is to improve the time required to solve *minimization* problems. To achieve this goal, the following objectives have been established.

1.1 General objectives

1.1.1 Pseudo-Boolean minimization

Given some pseudo-boolean constraints ($c_1x_1 + c_2x_2 + \ldots + c_nx_n \leq k$) where c_i and k are integers and x_i boolean variables, and a cost function, the goal is minimize it $min(a_1x1 + ... + a_nx_n)$ in a way that all the restrictions are satisfied.

This project will try to reduce the time to solve these problems through two approaches:

- Binary search:
  Implement the well known *Binary Search*[1] algorithm to find the minimum value for the cost function.

- Linear search:
  Some *SAT Solvers* can learn and derive new restrictions from previous problems. To take advantage of this ability it is necessary to implement a *Linear Search* algorithm.

1.1.2 Timeout

For some problems, it is more important to find a solution before a deadline than finding the best possible solution. For instance, a delivery company must have all the route planned for all trucks before the journey starts, therefore, they care more about having a solution than finding the best one.
For this, a *Timeout strategy* will be implemented in case that a good enough solution has been found or the problem does not seem to have one.

1.1.3 Multi-threading (optional)

This tool will take advantage of multi-core processors trying to split the problem and solving each part separately.

[1] Binary search is a search algorithm that finds the position of a target value within a sorted array. (more)

Chapter 2

Scope

- GEP
- Initial Stage
  In this stage, the previous code will be tested and refactored[1] to assure a solid base and the requirements and architecture of the system will be defined.
- Iteration 1: Pseudo-boolean minimization
  This iteration will be for the first objective: Pseudo-boolean minimization. In this stage we will work with PBLib, first adding a layer between the user and PBLib to make it more user-firendly and implementing search strategies to find the minimum value.
- Iteration 2: Timeout
  This iteration will be for the second objective: Timeout
- Iteration 3: Multithreading (Optional)
  This iteration will be for the third objective: Multi-threading. Here, some existing SAT solver will be modified to support multithreading techniques.
- Final stage
  In this stage, all possible bugs will be solved and feedback from the supervisor will be taken. Also, the final document and presentation will be finished.

[1] Code refactoring is the process of restructuring existing computer code—changing the factoring—without changing its external behaviour. (more)

Chapter 3

Computer science courses

The courses which will influence more this project are the following ones.

3.1 Informatics Logic

The main topic of this project, Pseudo-boolean minimization, is very related to Informatics Logic.

Informatics Logic taught me the tools and the knowledge to be able to do this project. What is SAT, a logic, a CNF, a BDD, . . .

Without this course, this project could not be done.

3.2 Algorithms

This course taught me different types of programming such as Dynamic programming and some advanced data structures. With this tools, more efficient algorithms will be developed in this project.

Another course will influence this TFG. It is not from CS speciality but it is worth to mention:

3.3 Parallelism

In this course, I learned to identify which parts of an algorithm are parallelisable and how to find the best performance. This course will be useful for the last iteration, apply multithreading on the program.

Chapter 4

Computer Science Adequacy

This project allows me to apply a lot of things learned in the speciality.
The topic of this project, SAT, is a well known NP-Complete problem. One of the requirements to work on it is understand how to evaluate hardness of the problems and, for example, what NP-Complete means. Also, know how the problem is solved nowadays and how the solutions are implemented. This knowledge will allow the modification of this solutions and apply parallelism to it.

As the main goal of this project is trying to reduce solving time it is very important to know how to develop efficient algorithms with critical requirements in reliability and efficiency.

Also, the topic of the project is hardly related to Computer Science. It is widely used in other subjects such as artificial intelligence, circuit design, . . .

Chapter 5

Technical Competences

Comment: In the inscription of the TFG, the competence CCO2.1 is selected. This was an error during the inscription process. The correct one is the competence CCO3.1 and it is the one which is explained in this deliverable.

5.1 CCO1.1 Evaluate the complexity of a problem, know algorithm strategies which could solve them, and recommend, develop and implement the solution which assures the best performance according to the established requirements. [Enough]

As previously explained, the topic of this project, pseudo-boolean minimization, is NP-Complete. The second and the third objectives of this project require modifying existing software, for example, a SAT solver. For this, it is a must to understand the algorithms implemented and modify them in the correct way.

5.2 CCO3.1 Implement code with critical requirements in execution time, efficiency and security. [Enough]

Because the goal of this project is to reduce the execution time for this problem, execution time and efficiency are critical requirements of the solution which will be implemented.