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 5, 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

For the problems of the form $min(c_1x_1 + c_2x_2 + \ldots + c_nx_n \leq k)$, the goal is to find an assignment for $\{x_1, x_2, \ldots, x_n\}$ so that k is minimum.

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

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

2.1 What and how?

To achieve all the general objectives1 of the project, the following stages have been established:

- Analyze, refactor[1] and test the existing code to have a solid base.
- Add the functionality of representing *PBF*.
- Study PBLib library to see which functionalities it has available to work with *minimization*.
- Implement *minimization* strategies.
- Study timeout strategies and implement them.
- Study and implement multithreading.

2.2 Possible obstacles

TODO: Possible obstacles that may hinder the execution of the project are briefly stated.

In this section, the possible obstacles and its solutions are exposed.

2.2.1 Base project

The existing project could be a source of bugs and other problems caused by not following an adequate methodology. For this reason and to solve possible issues, the first stage of the project will be focused on solving them.

2.2.2 Schedule

Due to the circumstances in which this project will be developed (Erasmus) possible delays could appear. To fix these circumstances, a realistic schedule with weekly communication will be planned. This will support a continuous development and detect as soon as possible delays.

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

2.2.3 PBLib

One of the main requirements of this project, *Pseudo-Boolean minimization*, is planned to be done with *PBLib* library. It may be this library does not fit as expected with the project forcing to find a substitute.

2.2.4 Correctness

As explained in *Rigor and Validation*4, correctness in this project is very important because of the context it is in.
Guarantee correctness could be hard and take more time than expected. If this happens, formal correctness could be delayed or reduced.

Chapter 3

Computer science courses

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

3.1 Informatics Logic

This project is very focused on the topics taught in Informatics Logic. Informatics Logic put in my disposition 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.

Chapter 4

Computer Science Adequacy

https://www.fib.upc.edu/ca/estudis/graus/grau-en-enginyeria-informatica/pla-destudis/especialitats/

Un graduat especialitzat en Computació haurà adquirit els coneixements necessàries per dissenyar sistemes informàtics complexos i crítics en termes d'eficiència, fiabilitat i seguretat. Des de la planificació dels vols d'un aeroport, o la verificació del funcionament d'un sistema de frenada ABS, fins al disseny de la interfície persona-màquina dels mòbils del futur. La coresponsabilitat social que obliga a exigir solucions cada cop més eficients, energèticament o econòmica per exemple, fa de l'informàtic amb aquestes habilitats un professional altament valorat en àmbits molt diversos. Per exemple, en àrees com la robòtica i l'optimització de processos a la indústria, els productes financers i la predicció a la banca, la planificació d'infraestructures a l'administració pública, l'experimentació científica i el tractament d'imatges en centres de recerca biomèdica, o la programació de jocs i aplicacions del web a la industria pròpiament informàtica.

La creixent exigència d'innovació front als nous reptes requereix de professionals entrenats per treballar amb rigor científic i que puguin integrar-se en equips multidisciplinars d'enginyers, científics o economistes. La vàlua de l'especialista en computació radica en la seva habilitat per innovar, i per detectar i garantir els requeriments crítics d'un sistema informàtic complex. Aquesta tendència en la nova indústria informàtica ve liderada per les firmes de més prestigi d'àmbit global.

Chapter 5

Technical Competences

CCO1
Tenir un coneixement profund dels principis fonamentals i dels models de la computació i saber-los aplicar per a interpretar, seleccionar, valorar, modelar i crear nous conceptes, teories, usos i desenvolupaments tecnològics, relacionats amb la informàtica.
CCO1.1
Avaluar la complexitat computacional d'un problema, conèixer estratègies algorísmiques que puguin dur a la seva resolució, i recomanar, desenvolupar i implementar la que garanteixi el millor rendiment d'acord amb els requisits establerts.

CCO3

Desenvolupar les solucions informàtiques que, considerant l'entorn d'execució i l'arquitectura del computador sobre el qual s'executen, aconsegueixin el millor rendiment.

CCO3.1

Implementar codi crític seguint criteris de temps d'execució, eficiència i seguretat.