UNIVERSITAT POLITÈCNICA DE CATALUNYA

DELIVERABLE 6: FINAL DOCUMENT

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

Introduction and Context

1.1 Introduction

1.2 Context

Before explaining the main problem which this project is about, *Pseudo-Boolean Minimization*, it necessary to do a quick introduction to a much wider topic.

Boolean satisfiability problems *(SAT from now on)* is the problem of finding a model[1] for a *Boolean Formula* (BF from now on). In other words, it is the result of evaluating the *BF* after replacing its variables for *true* or *false*.
SAT is widely used in Computer Science because it was the first problem proved to be NP-Complete[1][2] which allowed a lot of NP[3] problems be reduced to it.

What is a Pseudo-Boolean Formula?

In propositional logic, a *BF* is defined as following[2]:
Let P be a set of predicate symbols like $p, q, r, ...$

- All predicate symbol of P is a formula.
- If F and G are formulae, then $(F \wedge G)$ and $(F \vee G)$ are formulae too.
- If F is a formula, then $(\neg F)$ is a formula.
- Nothing else is a formula.

This representation has some limitations because it can only express properties which are *true* or *false*.

Pseudo-Boolean Formulas are functions of the form $f : B^n \rightarrow \mathbb{R}$. For example, the following formula is a *Pseudo-Boolean Formula* (PBF from now on): $3x + 5y$. Therefore, *BF* are a special case of *PBF* where the domain is $d = \{0, 1\}$.

Pseudo-Boolean formulae minimization

PBF minimization is a well known NP-Hard[4] problem.
It does the following:

[1] An interpretation which satisfies the formula.
[2] NP and NP-Hard.
[3] Nondeterministic polynomial time.
[4] NP-Hard: at least as hard as the hardest problems in NP (more)

Given some *PB Constraints* of the form $\sum_{i=1}^{n} x_i w_i \leq k$, where $w_i, k \in \mathbb{I}$ and $x_i \in \{0,1\}$, and a cost function , the goal is minimize it $min(a_1 x1 + ... + a_n x_n)$ in a way that all the restrictions are satisfied.

There is a big research in this field, more specifically in encoding *PBF* into *CNF*. In this paper, Hölldobler, Manthey, Steinke[4], some relevant *PBF* into *SAT* encodings are explained and a new one is proposed. One of the authors of this paper, Steinke, is also the author of *PBLib*.

1.3 Background

During the past semester (Q1 2017/2018), under the supervision of Dr. Jordi Cortadella, I had been developing a C++ library.
This tool allows the users to represent *BF* in a C++ program in an intuitive way, do operations between them and convert them into *Binary Decision Diagrams* (BDD from now on). However, the main functionality of this library is the conversion from a *BF* to *CNF*.
As previously explained, *CNF* is a particular type of a *BF*, a conjunction of disjunctions. *CNF* is an important format because it is the standard input for *SAT Solvers*A.1.
As shown in this paper, *Mitchell, Selman, and Levesque[6]*, there is a correlation between the number of variables, the number of clauses and the hardness of solving the *CNF*.

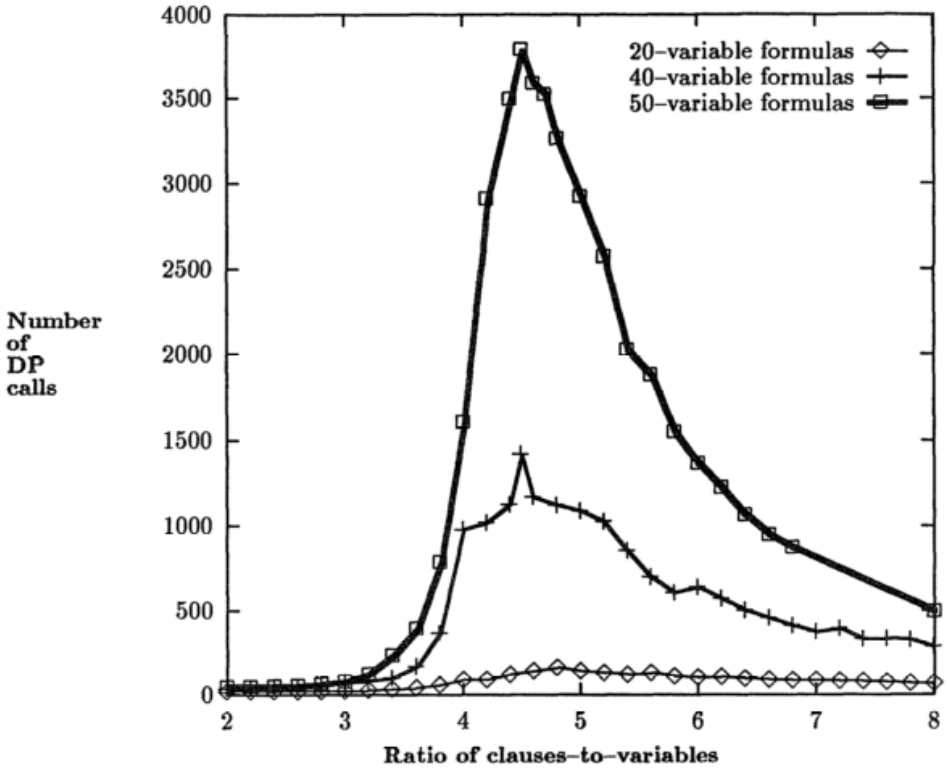

FIGURE 1.1: Median number of recursive DP calls for Random 3-SAT formulas, as a function of the ratio of clauses-to-variables.
Extracted from Mitchell, Selman, and Levesque[6]

Therefore, an improvement of the input *CNF* of the *SAT Solver* can reduce a lot the hardness of the problem.

This is the main goal of the library, try to reduce the size of the final *CNF* resulting from applying different converting methods on the original BF.

1.4 State-of-the-art

In this section, previous projects are discussed.
The first one is PBLib. PBLib is a C++ toolkit for encoding *PB Constraints* into *CNF*. As explained in *Steinke*[8], PBLib implements a lot of encodings:

At most one	At most K	PB
sequential[11]*	BDD[7, 4]**	BDD
bimander[6]	cardinality networks[1]	adder networks
commander[8]	adder networks[4]	watchdog[10]
k-product[3]	*todo: perfect hashing*[12]	sorting networks[4]
binary[2]		binary merge[9]
pairwaise		sequential weight counter[5]
nested		

* equivalent to BDD, latter and regular encoding
** equivalent to sequential counter

FIGURE 1.2: PBLib implemented encodings
Extracted from Steinke[8]

PBLib does not only implement these encodings, the most interesting thing is that it can decide which encoder provides the most effective translation.
It is very hard to compete on this aspect against PBLib, but it is not very user-friendly. For this reason, on of the goals of this project is to add a layer between the user and PBLib to simplify how the user declares the *PB Constraints*.

The second project we will talk about is the explained in the *Background section*. This project adds a new encoding for *BF* which can be extended to *PB Constraints*. This encoding will be studied and if it achieves good metrics it will be implemented to *PB Constraints*

1.5 Motivations

Informatics Logic is taught in this[5] faculty. In that course, I realized how important is *logic* through its lecturer, Dr. Robert Nieuwenhuis, and its activities.

In the first coursework, we had to code a *SAT Solver* which used *Unit Propagation*. With this activity, I comprehended how hard and substantial is the study of *logic* and all its context. For example, how *logic* is used in Artificial Intelligence and Planners.

When the time of deciding the *TFG* arrived, I contacted my actual supervisor, Dr. Jordi Cortadella, and he proposed me some topics and ideas. Finally, we agreed on doing this project.

[5] Facultat Informàtica de Barcelona

The motivation for this project is trying to deepen into the topic and contribute on it.

1.6 Stakeholders

In this section, the Stakeholders of the project are defined. Stakeholders are entities which are affected, directly or indirectly, by the solution developed in this project.

1.6.1 Target audience

This tools targets all the entities (researchers, companies, ...) which work with *PB minimization* and use *SAT Solvers*.

1.6.2 Users

The users will be C++ programmers due this tool is developed in this language.

1.6.3 Beneficiaries

All those entities which work with *PB minimization*. For example AI, SAT Solvers, Planners, ...

Chapter 2

Project Scope

2.1 Project Formulation

As mentioned before1, 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.

2.1.1 General objectives

Pseudo-Boolean minimization

For the problems of the formed by some *PB Constraints* $min(c_1x_1 + c_2x_2 + \ldots + c_nx_n \leq k)$ and a cost function , the goal is to find and assignment for $\{x_1, x_2, \ldots, x_n\}$ in a way that the cost function is minimum.

$$min(a_1x1 + ... + a_nx_n)$$

Previously1, it has been explained that this types of problems are *NP-Hard*. 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.

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

Multi-threading (Optional)

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

2.2 Scope

2.2.1 What and how?

To achieve all the general objectives2 of the project, the following stages have been established:

- Analyze, refactor[2] 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. (Optional)

2.2.2 Possible obstacles

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

Base project

The existing project *Background section*1 has not been developed following an adequate methodology. This could be responsible for a poor code quality. Building on top of a program with this characteristics could have terrible consequences because it would produce a lot of bugs and malfunctions hard to solve in the future.
For this reason, it is better to improve the quality of the existing code because it will avoid problems in the future.

Schedule

Because this is a final degree project the scope is limited, in this case until June of 2018. Considering the learning stage, analysis, requirements study, and other deviations which could appear, the time available to develop the project could be drastically reduced. Moreover, this project will be developed in an Erasmus stay which makes harder the planning. For these reasons, a good and realistic planning are key steps to take advantage of time and reduce contingencies.

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

PBLib

One of the main requirements of this project, *Pseudo-Boolean minimization*, is planned to be done with *PBLib* library. This is an obstacle because *PBLib* could not be compatible with the project causing compiling errors and therefore some time would have to be spent solving them. Also, *PBLib* could not have the expected functionalities, in which case a substitute should be found or, even worst, having to implement *PBLib* functionalities which would take a lot of time.

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.

2.3 Methodology and Rigor

Research is a vast process with no clear path between *a* and *b*. For this, it is important to follow some directions. A methodology will provide some guidelines to avoid possible problems, be more efficient and do the project more manageable.

2.3.1 Methodology

The methodology adopted for this project will be Agile[3]. It is important to clarify that this methodology will not be followed strictly but adapted to this particular case where there is only one developer and all the objectives are well defined. The main characteristics followed from Agile in this project will be:

- Short cycles
- TDD (Test-Driven Development)
- Weekly scrums with the supervisor

2.3.2 Tools

In this chapter the, development tools will be introduced.

Git

Git will be used in this project as a Version Control System because it allows maintaining a tracking of all the changes made (commits), and what is more important, return to them at any time. In addition to this, it enforces a short cycle development (because commits are small units of work) and the developer has to document them which matches perfectly with Agile methodology. GitHub will be the repository service used.

Trello

Trello is a simple and flexible web board which helps to organize tasks and its state. It will be used in this project to manage tasks and priorities.

[3]Methodology based on the on the adaptability in front of any change to improve exit possibilities.

2.3.3 Communication

Due to my conditions, I'm currently studying abroad in an Erasmus program, all the communication will be made through electronic means. The majority of it will be made using e-mail but if it is necessary a video conference could be done.
The minimum communication with the supervisor will be a weekly e-mail report where all the tasks done during this period will be explained. Problems or questions will be also exposed if any.

2.3.4 Rigour and Validation

Rigor and Validation for this project are relevant.
The surrounding of it, such as *Artificial Intelligence, Planners, Cryptographic Protocols verification, ...*, are widely used nowadays and have been becoming more popular lately. This means that this project could have a big repercussion and be used by some professionals. For this, it is important to guarantee the validation and correctness of the project.
During the development, TDD will be used to avoid unnecessary code (possible origin of bugs) and assure the correctness of the implementation. It is also possible to formalize and prove all the operations done by the software.
Finally, my supervisor could give me some orientation and validate, if necessary, the operations done.

Chapter 3

Project Planning

3.1 Schedule

3.1.1 Estimated project duration

For this project there have been estimated 450 hours of work, starting on **19th of February** and ending on **23rd of June**.

3.1.2 Considerations

The original plan could be modified to be adapted to deviations. Agile methodology implies that some new requirements can appear which could modify the planning. It is hard to do a realistic planning with Agile methodology because the iteration's requirements are not fully known until the Planning stage.

Because this project will be developed sequentially by only one person, the realization of a PERT diagram has been discarded. Nevertheless, some part of the documentation will be done in parallel.

3.2 Resources

For the development of this project, three types of resources will be needed.

3.2.1 Human Resources

- One person working 20 hours per week until the finalization of the project.

3.2.2 Material Resources

- Lenovo IdeaPad U330T
  This laptop will be used to write the documentation and develop the project.

3.2.3 Software Resources

- Trello: Web application to manage project tasks.
- teXstudio: LateX editor to write all the documentation.
- e-mail: Communication tool used to contact the supervisor.
- Atom: Text editor to write the code.

- Git: VCS to backup and keep tracking of the project.
- C++: Language used for the development.
- PBLib: C++ library for Pseudo-Boolean encodings.
- CLion: Code editor focused on C++.
- Google Test: Unit testing framework for C++ developed by Google.

3.3 Project Planning

GEP

This task corresponds to the work done during the GEP course. This task has not any dependency but the work done will be used for the final documentation.

The estimated time for this stage is 70 hours.

Initial Stage

This stage will be used for defining the requisites to accomplish, the architecture of the software and refactor the previous code. Also, the required tools will be installed.

The estimated time for this stage is 90 hours.

Iterations

Because Agile methodology will be followed, the project has been divided into iterations. There will be a total of 3 iterations: Pseudo-Boolean minimization, Timeout strategies, and Multithreading, being this last one optional.

For each iteration, 80h of work are estimated.

Planning
This stage will be used for defining the scope of the iteration and goals.

This stage will be 10 hours long.

Development and TDD
In this stage, the iteration will be developed and tested.

This stage will be 60 hours long.

Finalization
In this stage, all possible bugs will be solved and feedback from the supervisor will be taken.

This stage will be 10 hours long.

Final Stage

Here, all the development will be finished and it will be used for finishing all the documentation and prepare the final presentation.

This stage will take 50 hours.

3.3.1 Gantt Diagram

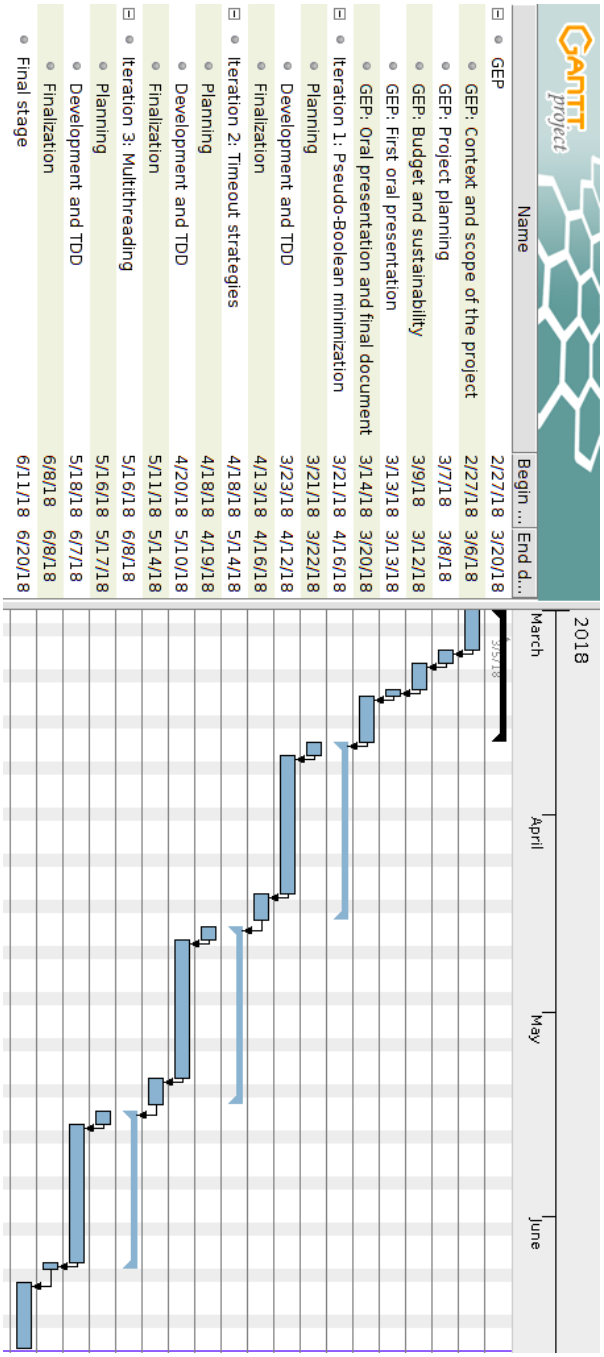

FIGURE 3.1: Gantt diagram of the project

3.4 Alternatives and Action Plans

Because of using an Agile methodology, the project functionalities can be easily adapted during the development.

3.4.1 Potential deviations

Incorrect estimations

It could be that the purposed estimations are not correct and be underestimated or overestimated. In the first case, if some iteration takes less time than expected, the next iteration will be started immediately. On the other case, if an iteration takes more time than expected, more hours will be added, for example, weekend hours. If even with this countermeasures the iteration is delaying the project some optional improvements will be discarded in order to guarantee the main functionalities and the project would be finished before the deadline. The effect on the resources would be more electricity consumed by the laptop on the added hours.

Summer internship

The deadline for this project is the 23rd of June. I have applied for a CERN internship from 4th of June until 31st of August.

In case of being accepted into the program, the final stage would have to be done during the internship. Since the final stage is for doing the final document and the final presentation, it can be lightened by paralleling part of its work during other stages.

The duration of the project is not affected because the only thing that would be done is a different distribution of the work. For the same reason, it would not have any effect on the resources.

Compatibility issues

As explained before, in section X, PBLib could cause some compatibility issues with the existing software which works with another library, Cudd. If this happens, in the best situation some hours will be spent fixing this issue or finding a substitute. In the worst situation, the functionalities of PBLib would have to be implemented which would take a lot of time. The project duration could be affected but it would be always finished before the deadline using weekend hours. The resources which will be used in addition is more electricity consumed by the laptop.

Base project code quality

The existing project could have some quality issues and be a source of bugs which would cause a lot of delays. For this reasons, and as s explained in section Y, the first stage of this project will be focused on assuring quality code of the existing project. The project duration would not be affected because part of the first stage is focused on that guaranteeing project finalization.

As explained in each potential deviation, the priority is to finish the project before the deadline. As said, extra-hours from the weekend can be used to work on the project to accomplish all the objectives. If even these hours are not enough, then the optional objective *Multithreading* will not be realized.

Chapter 4

Economic Management

In this section, all the costs of the project are exposed.

4.1 Direct costs

Direct costs are those that have a direct relationship with the manufacture of the product. In this case, the only direct costs are the human resources.

4.1.1 Human resources

The cost of the human resources has been estimated with the following expression: $Cost = \frac{Salary}{Hour} \times ExpectedHours$. The salaries have been extracted from PagePersonnel study[7]. In this study, the salaries are expressed per year. In average, there are 1.500 working hours per year. To obtain the price per hour, the salary per year has been divided by the working hours per year.

Taking into consideration the Gantt chart from the previous deliverable, the dedication of each role has been defined as follows:

Stage	Project Manager	Software Architect	Developer
GEP	70	0	0
Initial Stage	30	30	30
Iteration 1,2,3	6	147	87
Final Stage	30	0	20

TABLE 4.1: Hours destined to each stage per role

Role	Estimated hours (h)	Price/hour (€)	Total cost (€)
Project Manager	136	27	3.672
Software Architect	177	25	4.425
Developer	137	14	1.918
Total			10.015

TABLE 4.2: Human resources budget

4.2 Indirect costs

Indirect costs are those that do not have a direct relationship with the manufacture of the product. In this case, the indirect costs are Hardware, Software, and some others.

4.2.1 Hardware

According to *Agencia Tributaria*[1], the maximum number of years to amortize a computer equipment is 8. Therefore the amortization of Hardware resources has been calculated following this expression: $Amortization = \frac{Price}{8 \times 12} \times 5$

Product	Price (€)	Units	Useful life (y)	Amortization (€)
Lenovo IdeaPad U330T	899	1	8	46,83
Total				46,83

TABLE 4.3: Hardware resources budget

4.2.2 Software

For software resources, free tools have been selected and student discounts have been used to minimize the total cost.

Product	Price (€)	Units	Useful life (y)	Amortization (€)
GitHub	6,10/month	5	N/A	30,5
GitHub student pack	-6,10/month	5	N/A	-30,5
Clion	6,90/month	5	N/A	34,5
JetBrains Product Pack for Students	-6,90/month	5	N/A	-34,5
Atom	0,00	1	N/A	0,00
TeXstudio	0,00	1	N/A	0,00
Total				0,00

TABLE 4.4: Software resources budget

4.2.3 Other resources

Internet connexion price has been extracted from Pepephone[2] plan, which is 34,6€ per month.
kWh price has been extracted from Selectra. The average price per kWh is 0,12€. In office supplies paper packs, books, pens, ... are included.

4.3 Contingency

The contingency percentage for direct costs has been estimated following my experience on past projects. For indirect costs, the budget is easier to estimate therefore a small percentage has been selected.

[1] Agencia Tributaria - amortizations
[2] Pepephone fibra

Product	Price(€)	Units	Total (€)
Internet connexion	0,047/h	450 hours	21,15
Power consumption	51Wh	450 hours	2,75
Print	0,05/page	400 pages	20
Office supplies	50	1	50
Total			93,9

TABLE 4.5: Other resources budget

Concept	Price (€)	Percentage (%)	Total (€)
Direct costs	10.015	30	3.004,5
Indirect costs	140,73	15	21,11
Total			3.025,61

TABLE 4.6: Contingency budget

4.4 Unforeseen

The first unforeseen is that the computer breaks. In this case, a new one will be bought. The other unforeseen events are that the stages of the project being extended. For each stage, a 50% delay has been estimated.

Unforeseen	Cost (€)	Probability (%)	Total (€)
Broken computer	1.300	5	65
Delay GEP stage	945	15	141,75
Delay initial stage	990	15	148,5
Delay iteration 1	842,5	15	126,38
Delay iteration 2	842,5	15	126,38
Delay iteration 3	842,5	15	126,38
Delay final stage	545	15	81,75
Total			816,14

TABLE 4.7: Unforeseen budget

4.5 Total budget

In conclusion, the total budget of the project is:

	Cost (€)
Direct costs	10.015
Indirect costs	140,73
Contingency	3.025,61
Unforeseen	816,14
Total	13.997,48

TABLE 4.8: Total budget

4.6 Control management

The control management mechanisms will be used to study and compare deviations.

The Human Resources is an initial estimation, therefore, the estimated cost and the real cost obtained once the project is finished will be compared. In any case, an hour follow-up will be done for each iteration and the functionalities implemented to see if the planning is accurate, or correct possible deviations and decide which functionalities could be added or deleted in order to accomplish the planning. Another method to solve the possible deviations could be reorganizing the Gantt chart.

At the end of the project, the original estimated budget will be compared with the real one. Finally, a study of which deviations and unforeseen appeared will be done and check if they can be covered by the contingency budget. This analysis will be very useful to realize future budgets and to apply the extracted conclusions.

The indicators used for that are: Variance in cost by rate, efficiency variance, variance in totals, ...

Chapter 5

Sustainability and Social Commitment

5.1 Sustainability Matrix

In this section, the sustainability matrix is resumed according to the numbers described here[3].

	PPP	Useful life	Risks
Environmental	7	20	-4
Economical	7	15	-10
Social	8	15	0
Sustainability range	58		

TABLE 5.1: Sustainability matrix

5.2 Economic dimension

5.2.1 PPP

The estimated budget of the project can be found in table 4.8. The estimated budget is 13.997,48€. This number has been estimated taking into account the working hours of each role, the hardware and software used, indirect costs, contingency, and unforeseen events.

5.2.2 Shelf life

Nowadays, the no optimization of Pseudo-Boolean encodings implies that the problems are bigger and harder which causes a long execution and more consumption of resources. With the optimizations that this project will study, the final execution time could be reduced therefore the power needed to solve the problem which translates into a more reduced cost.

5.2.3 Risks

As exposed previously, some risks are problems with the planning, problems with the tools used,...
The main risk is that the optimizations proposed are not useful in a practical environment.

5.3 Environmental dimension

5.3.1 PPP

The estimated electric usage for this project can be found in this table 4.5. The estimation has been done with this expression: $E = \frac{W}{h} \times T$. In this project $E = \frac{51W}{1h} \times 450h = 22,950kW$

It is hard to minimize more the impact of this project. Some strategies are turning off the computer when not using it, minimizing the amount of paper used, ...
Some resources are reused, for example, instead of writing all the functionalities, some C++ libraries will be used.

5.3.2 Shelf life

It is hard to measure the footprint of this project along with all its useful life. It will depend on the success of the project and how many people will use it.

Currently SAT problems are executed in SAT-Solvers using some optimizations. As explained before this problem is NP-Complete which among other things implies that there is no known algorithm which can solve it in polynomial time. In other words, solving SAT is very time and resource expensive. Also, SAT is widely used in many fields. For example, computational complexity, databases, programming languages, artificial intelligence and system verification. This translates into a big electricity consumption and a huge footprint. For example, the MareNostrum[5] supercomputer spends 1,3MW/year.

This project purposes more optimizations to reduce the execution time. Even if these optimizations are small, because SAT is widely used, it could have a huge impact. It will have a positive impact because it will reduce the total CO^2 emissions released by the computers used to solve them.

5.3.3 Risks

The footprint of this project could be worst than expected if the development of it is extended.

5.4 Social dimension

5.4.1 PPP

This first stage of the project, GEP, will improve my management and planning skills, my English abilities, how to document and budget projects.
The other stages will expand my knowledge about informatics and the opportunity to put in practice a lot of skills developed during this degree.
Finally, my ability to present in front of people and defend the work done during these months.

5.4.2 Shelf life

This project will improve a lot of fields because SAT-Solvers are widely used. For example, Planners, Artificial Intelligence, ... which can have an unpredictable impact

in the life of people.

Currently this problem is solved using other techniques. The solution that this project purposes is an addition to them (it is not exclusive). There is a real need for this type of projects because as said previously, SAT is an NP-Complete problem therefore any improvement on this field will reduce the hardness of the problem with all the consequences this implies.

5.4.3 Risks

The only negative impact that this project can have is not being used. In this case, it will not be used and the society will remain unchanged.

Appendix A

More information

A.1 Why SAT Solvers use CNF as input format?

There are two mains reasons for this: Equisatisfiability and Computational Complexity. Let us start with the first one:

Two *BF* are **equisatisfiable** if and only if both have the same *models*. This may seem the same as equality but it is not because in an equality relationship both *BF* have to have the same variables.
This is important because between a *BF* and its result from a *CNF* transformation the equisatisfiability is preserved which means that if the *SAT Solver* finds a *model* for the *CNF*, then this *interpretation* will be also a *model* for the original *BF*.

The second reason is computational complexity. Let us have a look at the following table:

	DNF	CNF
TAUT	NP	P
SAT	P	NP

TABLE A.1: Complexity of deciding if a *BF* is SAT or TAUT depending of its format.

So as a *BF* can be converted into a *CNF* in linear time while preserving equisatisfiability, *SAT Solvers* will use them to target satisfiability.

Bibliography

[1] Stephen A. Cook. "The complexity of theorem-proving procedures". In: *Proceedings of the third annual ACM symposium on Theory of computing - STOC '71*. New York, New York, USA: ACM Press, 1971, pp. 151–158. DOI: 10.1145/800157.805047. URL: http://portal.acm.org/citation.cfm?doid=800157.805047.

[2] Rafael Farré et al. *Notas de Clase para IL - 2.Definición de la Lógica Proposicional*. Barcelona, 2009. URL: https://app.box.com/file/225148187559.

[3] Jordi Garcia et al. "Artículo invitado La sostenibilidad en los proyectos de ingeniería". In: (). URL: https://upcommons.upc.edu/bitstream/handle/2117/23240/127-1047-1-PB.pdf.

[4] Steffen Hölldobler, Norbert Manthey, and Peter Steinke. "A Compact Encoding of Pseudo-Boolean Constraints into SAT". In: (). URL: https://mail-attachment.googleusercontent.com/attachment/u/0/?ui=2{\&}ik=7328377021{\&}view=att{\&}th=15e2e7b559196d67{\&}attid=0.2{\&}disp=inline{\&}safe=1{\&}zw{\&}saddbat=ANGjdJ9S{_}JppQCGQWibmAw70nm3SJsSOBhJg1PGGwc}A5wVWlyGHOKOsaPgNPEw3Y8EjoJoRyRKQ03MZA1Uma98pSK7xDtv5FPsTktHevmFB7ZP79m3vMbP7MvPgILP}2u{_}{_}eWwhfC14QE82TOnITUjuH7rGsdMKwsKwRrkwi-CoyPoJT8RAmSh2jKobXWTIejzVljGM8WWvOm}pgrV1LSjW4clhxqLfJfA-jW7Hi-AHADvop{_}Z{_}IiGErkOr-rhH9aMR7Z6-pcC4aECx046{_}NOeCx5Fhmf-z71ZuWwUZ3PrMtEdDrDoJrmLDa2sWDQIuNom4yykZjVqBNSNoimvTOdUk31}CnkIjv5T15jmLv-hZGLO1f77keyiLsiHKGLZ{_}HAJGw9oLDXThmzABOivSmGjszgdj{_}X1ODplVtdsqWtqViPDVmbct0tXfd33PYgBil3I08hQ.

[5] *MareNostrum 4 supercomputer to be 12 times more powerful than MareNostrum 3 | BSC-CNS*. URL: https://www.bsc.es/news/bsc-news/marenostrum-4-supercomputer-be-12-times-more-powerful-marenostrum-3 (visited on 04/06/2018).

[6] David Mitchell, Bart Selman, and Hector Levesque. "Hard and Easy Distributions of SAT Problems". In: (). URL: https://aaai.org/Papers/AAAI/1992/AAAI92-071.pdf.

[7] PagePersonnel. "Selección y trabajo temporal especializado". In: (). URL: https://www.pagepersonnel.es/sites/pagepersonnel.es/files/er{_}tecnologia16.pdf.

[8] Peter Steinke. "PBLib – A C++ Toolkit for Encoding Pseudo-Boolean Constraints into CNF". In: (2015). URL: http://tools.computational-logic.org/content/pblib/pblib.pdf.