UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Incremental Algorithm for large Netwotks

Project management (GEP)
Deliverable 4: Final report

Pol Forner Gomez

Thesis supervisor: Gerard Escudero Bakx (Department of Computer Science)
Thesis co-supervisor: Edelmira Pasarella Sanchez (Department of Computer Science)
GEP tutor: Joan Sardà Ferrer
Degree: Bachelor's Degree in Informatics Engineering (Computing)
February, 2024

Contents

Context

This work is a bachelor degree of a computer engineering degree, specialization in Computing. The degree is done in the Facultad d'Informatica de Barcelona (FIB) of Universitat Politàcnica de Catalunya (UPC) and is directed by Gerard Escudero Bakx and supervised by Edelmira Pasarella Sanchez.

This project is based on the master thesis made by Juan Pablo Royo Sales in 2021 [1]. that was also supervised by Edelmira. Juan Pablo master thesis is about an implementation of an algorithm and my objective is to improve algorithm he implemented. And that is why, in this first section, I would like to introduce some concepts that are important to understand the basics of the Juan Pablo work and also to understand the concepts that I will be using in this project. Therefore, here only will be explained the concepts that are important to my work, summarizing and adding some notes to Juan Pablo's work. If you are interested in more detailed explanation of the concepts, you can read Juan Pablo's work.

1.1 Introduction of concepts

Nowadays, the amount of data that has been generated is huge and more important, it is growing every day. Sensors, social networks, and other sources generate data that needs to be processed and analyzed to extract useful information. The main problem is that, when we try to process this data, we can not use the traditional methods that we use with small amount of data, because the time and resources needed are too high. To solve this problem, we need to develop new algorithms and techniques that can deal with.

Streaming

In addition to having all that data, the amount of data normally it can not be stored and it comes in what is called a data stream. A data stream is a sequence of data that made available over time, meaning that we can not store all the data and we need to compute it in time. For example, a sensor of temperature that sends the temperature every second, a traffic camera that registers all car plates that pass in front of it or just a social network that generates a huge amount of data every second.
This kind of data mentioned before needs to be processed and sometimes the data never ends, so we can not wait to finish to give a result and we must give results along the way.

Incremental algorithms

Here is where incremental algorithms come in.[2] Incremental algorithms give us the ability to obtain results from subsets of data and then update the results before finishing the whole data. This is very useful, because some problems do not need to be solved with all the data or maybe we are not interested in the final result. Recovering a previous example, if we are interested in which models of cars drive in certain road, we can use the camera that registers the car plates to get the answer. It is stupid to wait until the end of data to give the answer (also because it never ends), so we can give a result when we check it. In conclusion, incremental algorithms could be a good approach to solve some problems.

Paralelism

One of the most important techniques for dealing with time problem is parallel computing or parallelism. Parallelism allows us to divide the work and process it in different machines concurrently, reducing the time needed to process the data. When we try to fight against this huge data problems, we must find a

solution that can be parallelized given that modern machines have multiple cores and we can use them to process the data.
If we put together streaming and parallelism, it can be distinguished two computational models:

- **Data Parallelism**
  This model splits the data and processes it in parallel. All the computations that perform some action over a subset of data, do not have any dependency with other parallel computations. This model has the advantage that it can implement stateless algorithms, allowing to split and process the data into different machines without contextual information. Nonetheless, this model has the disadvantage that when we need to be aware of the context, it is penalized.

- **Pipeline Parallelism:**
  This model splits the computation in different stages and each stage takes the result of the previous stage to make the computation. The parallelization is done by parallelizing the stages. The main advantage is that stages are non-blocking, meaning that we do not need to process all data to execute the next stage. This allows us to make incremental algorithms. In spite of that, the main disadvantage is that one stage could be the bottleneck of the pipeline and delaying all the process.

Dynamic Pipeline Paradigm

Now that we talked about these 3 concepts: incremental algorithms, streaming and parallelism, I can introduce the next concept that I am going to be working in this project.
The dynamic pipeline Paradigm is a Pipeline Parallelism model "*based on a one-dimensional and uni-directional chain of stages connected by means of channels synchronized by data availability*". [1, Page 9, 2.2] This chain is called Dynamic Pipeline and it can grow and shrink with the continuous arrival of data. So we can use a data stream to feed the pipeline and it will process the data growing and shrinking the as needed. With this paradigm we can implement incremental algorithms and process the data in parallel easily.

1.2 Problem to be solved

In his work, Juan Pablo implemented an incremental algorithm using dynamic pipeline paradigm to resolve a graph problem: finding bitriangles in bipartite graphs. He decided to implement the algorithm using the functional programming language Haskell, and because of the no existence of one, he created a framework to implement the dynamic pipeline paradigm. Here is where my project comes in, I will use his framework to implement an easier problem to understand the dynamic pipeline paradigm and then I will try to improve the original algorithm that he made.

As said before, the algorithm to improve finds bitriangles in bipartite graphs and here i will not explain the problem since I consider that it is a little difficult and the nature of the problem is not decisive enough for my resolution. If you are interested in the problem, you can read Juan Pablo's work [1, Page 5, 1.1].

The easier problem that I chose for learning about the Haskell framework is the word counting problem. As easy as it sounds, the problem does not need further explanation: we just need to count the number of occurrences of each word in a dataset.

1.3 Stakeholders

This project is co-supervised by Edelmira Pasarella Sanchez, who is a researcher and supervised Juan Pablo work. She and her team developed the algorithm so they are the main stakeholders of this project. Also the director of this project, Gerard Escudero Bakx, is the one who proposed me to do this project because he was interested in the Haskell framework made by Juan Pablo. So Gerard is also a stakeholder of this project.

Apart from these direct stakeholders, this project will help to add knowledge to the community about the dynamic pipeline paradigm code examples and more important, will add more code and knowledge to the world about the Haskell framework of Juan Pablo.

Justification

Well first we need to ask ourselves if is it worth or necessary to improve the algorithm. As I do not really know well about the algorithm, I talked with Edelmira and discuss some points where we can improve the algorithm. Their team has done some improvements since Juan Pablo's work, and also tell me about some weak points of the algorithm. So yes, I think that is worth to improve the algorithm.

With my actual knowledge of the algorithm, I can not tell if it is better to improve the algorithm or to make a new one. But another time Edelmira told me that the algorithm was good and that it was worth to improve it.

Another point to take into account if it is worth to use the Haskell framework made by Juan Pablo. This may not be clear now, because this framework is unique and there are no other examples of it. This work can be very useful for testing the framework and extracting a conclusion about this.

Scope

3.1 Objectives

This project has the following main objectives:

- **Word counting problem:** The first goal of this project is to make an implementation, using the Haskell framework for dinamic pipeline paradigm, of the word counting problem.
- **Improve Juan Pablo's algorithm:** The second goal is to improve the implementation of the algorithm mentoined before.

Apart from these main objectives, I have some sub-objectives that I would like to achieve at the end of the project:

- Learn about the dynamic pipeline paradigm.
- Improve my Haskell skills, from my actual basic level to a competent level.
- Learn about the inner workings of the Haskell programming language.
- Acquire knowledge of how to do a bachelor degree project for future projects like master thesis.

3.2 Requirements

For the correct development and validity of my final result, it is necessary that my final implementation solves all cases correctly and more efficiently on average than the implementation from which we started. It is also important to follow a correct and modular structure to facilitate possible modification. My optimization has to be in line with known improvements and must be correctly validated. My solution also has to be understandable and adaptable, in order to improve its reuse and sharing. As for the code, I have to make sure it is well documented. Finally, it is necessary that all Haskell functionalities used are updated and supported

3.3 Potential obstacles and risks

Like all projects, you always have to take into account the potential obstacles and risks that can appear. Here are the principal ones that I have identified:

3.3.1 Time Limit

I will say that this is one of the most important risks, because as there is a deadline to finish the project, an inconvenience can make to not finish the project. Despite this, I this that I have done a good objective planning and I will be able to redirect the project if any problem appears.

3.3.2 Dynamic pipeline framework

I did not use and examined the framework and a problem that worries me is that the framework has some bugs or is not well implemented. This can potentially make me lose a lot of time but as I check, Pablo did a good work documenting it so I trust that I will not have a lot of problems.

Methodology and rigor

4.1 Methodology

For this project, I will following a methodology based on the Scrum methodology, as I have always work with it and I have had good results. Originaly, Scrum [3] was designed for team work, but I will going to take the idea and adapt it to my project, that only I do it individually.

Scrum is a methodology that is based on the iterative and incremental development of the project, where the project is divided into small tasks that are done in a short period of time called sprint. This sprints have 3 main phases: planning, development and review. Taking this concepts into acount, I will divide the project into tasks and I will be following 1 week sprints. It will help me keep good control of the pace of the project and not fall asleep or fall behind. Apart from the experience using this methodology, another of the main reasons for using it is the potential to redirect the project in case of possible problems, as I consider crucial.

4.2 Project monitoring and validation

As good programming practices, I will be using git [4] to control the versions of the code. This will allow me to have a backup of the code and a practical way to a acces to previous versions. Also, I will be using Trello as a task manager, to keep track of the tasks and the progress of the project. Trello is very useful and combines very well with the methodology that I have chosen, since I will be able to have a record of the tasks to be done and the tasks completed.

To validate all about the Haskell work, I will be asking and following Gerard advice. Since Gerard has a great level of Haskell language, he will be able to help me validate that I follow a good structure and use of Haskell. Finally, facing the end of the project, Edelmira will help me with the correction and validation of the improvement and implementation of the algorithm.

Description of tasks

Before starting, I would like to thank my dear friend Alex Herrero to share with me his Latex project that he did for GEP months ago [5]. I will use his Gantt chart as template for mine, so I will not need to spend time in the creation and investigation of pgfgantt package.

This project officially started the second week of February 2024 although some mails and meets were done before with supervisor and co-supervisor. I will not include this previous work in the planning, as I consider not necessary. This work will be done during the following months, until the third week of June 2024, approximately one week before the oral defense. This is a total of 17 weeks and TFG is an 18 ECTS project, so it will need 450 hours of work, 26 hours per week approximately. Here I will define the principal sections of my project, with the possible resources needed and the minimum hours needed:

5.1 Project definition and planning (G)

This section includes everything done in GEP, since its completion is mandatory and has set deadlines. So, all the following task are documentation and only requires my personal laptop since I have the entire environment required to develop in LaTeX. I also may need to contact my supervisor to check some details and ask for some doubts and also check for my GEP tutor feedback of every deliverable. This task has a linear dependency (**see Figure 5.1**) because of GEP structure (in reality the only dependence is that G4 needs to be done the last one).

- **G1** Context and Scope
  This corresponds to the first deliverable of GEP, where is defined the context and scope of the project.
  Role: Project Manajer
  Expected dedication: 20 hours

- **G2** Planning
  This corresponds to the second deliverable of GEP, where is defined the planning of the project.
  Role: Project Manajer
  Expected dedication: 12 hours

- **G3** Budget and Sustainability
  This corresponds to the third deliverable of GEP, where is defined the budged and sustainability of the project.
  Role: Project Manager
  Expected dedication: 18 hours

- **G4** Final Document
  This corresponds to the last deliverable of GEP, where contains all the previous deliverables together with the necessary improvements using the feedback.
  Role: Project Manajer
  Expected dedication: 25 hours

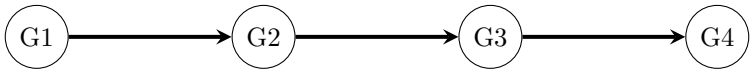

Figure 5.1: Task Dependencies for Project definition and planning, self elaborated

5.2 Research and Learning (RL)

In this section is included all the research and learning that I need for the correct development of the project. Here I will only need access to internet to search for information and access to the different pages and documents that I will be using. The principal internet resources that I will use are: Juan Pablo's TFM **TFM**, Haskell notes made by Jordi Petit from the subject LP **ApuntesHaskell** and the Dynamic Pipeline Framework repository **GitHubDynamicPipeline**. I may also need to contact my supervisor for Haskell doubts and also Juan Pablo if some critical doubt appears. This section also have some dependence (**see Figure 5.2**) because of the nature of the tasks.

- **RL1** Haskell Refresh
  The refresh of my Haskell knowledge and improvement of it. Check for possible libraries and codes
  Role: Project Manajer
  Expected dedication: 20 hours

- **RL2** TFM assimilation
  The reading and understanding of Juan Pablo's work.
  Role: Researcher
  Expected dedication: 25 hours

- **RL3** Dynamic Pipeline Framework
  Intensive review of the framework repository.
  Role: Haskell Developer
  Expected dedication: 30 hours

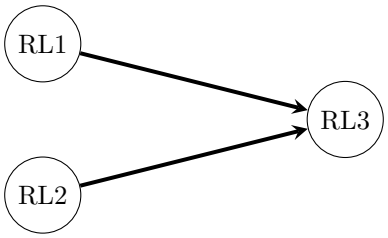

Figure 5.2: Task Dependencies for Research and Learning, self elaborated

5.3 First Algorithm Development (FA)

This section includes the development of the first algorithm, the one that will be put to test my Haskell knowledge acquired. Here I will need my personal laptop (as I have the entire environment required) and access to the GitHub repository **GitHubTFG**. For all the Haskell doubts I may need to contact my supervisor and for Dynamic pipeline doubts I should contact my co-supervisor as she is the expert. This task has a linear dependency (**see Figure 5.3**), but a critical testing error may need to reimplement some parts of the code, generating a circular dependency.

- **FA1** Algorithm Scaffold
  Here will be set the base of the algorithm and the form of the dynamic pipeline. Check for possible libraries and codes
  Role: Haskell Developer
  Expected dedication: 20 hours

- **FA2** Algorithm Implementation
  This part is the pure implementation of the algorithm.
  Role: Haskell Developer
  Expected dedication: 50 hours

- **FA3** Algorithm Testing
  Testing of the algorithm implementation, littles changes are also contemplated here (no big changes)
  Role: Haskell Developer
  Expected dedication: 10 hours

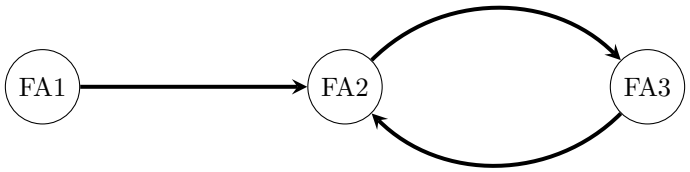

Figure 5.3: Task Dependencies for First Algorithm Development, self elaborated

5.4 Second Algorithm Development (SA)

This is the final section and the goal of this work, here I will try to improve the original algorithm. Again, here I will only need my personal laptop and access to internet resources. This section will need a lot of contact with my supervisor and co-supervisor, specially my co-supervisor, who is the expert in the algorithm. Here the dependence are similar to the previous section (**see Figure 5.4**), as is also a development of an algorithm. This task is especial, because it is difficult to estimate the exact scope of my project, so maybe I make some improvement and then have more time to repeat this process and make another improvement.

- **SA1** Finding Weak Points
  Here I will be looking for possible improvements in the code.
  Role: Haskell Developer and Researcher
  Expected dedication: 15 hours

- **SA2** Improvement Scaffolds
  Here will be set the base for the improvements of the algorithms find in the previous task Check for possible libraries and codes
  Role: Haskell Developer
  Expected dedication: 15 hours

- **SA3** Algorithm Implementation
  This part is the pure implementation of the algorithm.
  Role: Haskell Developer
  Expected dedication: 80 hours

- **SA4** Algorithm Testing
  Testing of the algorithm implementation, little changes are also contemplated here (no big changes). Here more time is needed in comparation of previous section as the size of the code and algorithm
  Role: Haskell Developer
  Expected dedication: 20 hours

5.5 Documentation (D)

Finally, this last section is a bit special because it is not planned to be done at the end of the all the previous sections. It is planned to be done in parallel as the project is developed and the different sections

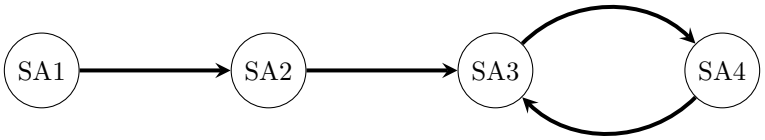

Figure 5.4: Task Dependencies for Second Algorithm Development, self elaborated

are finished. Here, all I need is my personal laptop to write all the documentation using LaTeX. Here I may need to contact for some help and feedback, but is not mandatory. There are no direct dependence, but we can consider that we need to complete one section or task before starting to write, so we can consider a graph like shown below (**see Figure 5.5**).

- **D1** Documentation of RL section
  This corresponds to the documentation of the research and learning section.
  Role: Project Manager
  Expected dedication: 10 hours
- **D2** Documentation of FA section
  This corresponds to the documentation of the first algorithm section.
  Role: Project Manager
  Expected dedication: 20 hours
- **D3** Documentation of SA section
  This corresponds to the documentation of the second algorithm section.
  Role: Project Manager
  Expected dedication: 20 hours
- **D4** Final Documentation
  This corresponds to the union of all the previous parts, with the additional parts (conclusions, bibliography, etc.) and the revision of the entire document.
  Role: Project Manager
  Expected dedication: 50 hours

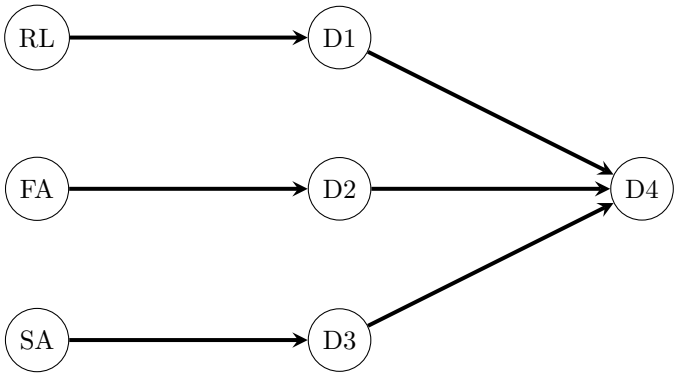

Figure 5.5: Task Dependencies for Documentation, self elaborated

Estimations and Gantt chart

6.1 Summary table

Now that we defined all the tasks, minimum hours and dependence, we can make this following summary table:

Description	TAG	Hours	Previous Tasks	Requirements	Human Resource
Project definition and planning	**G**	**75**	**-**	**-**	**-**
Context and Scope	G1	20	Laptop	None	None
Planning	G2	12	G1	Laptop	GEP Tutor
Budget and Sustainability	G3	18	G2	Laptop	GEP Tutor
Final Document	G4	25	G3	Laptop	GEP Tutor
Research and Learning	**RL**	**75**	**G**	**-**	**-**
Haskell Refresh	RL1	20	None	Laptop, Books	Supervisor
TFM assimilation	RL2	25	None	Laptop	Supervisors
Dynamic Pipeline Framework	RL3	30	RL1, RL2	Laptop	Juan Pablo
First Algorithm Development	**FA**	**80**	**RL**	**-**	**-**
Algortim Scaffold	FA1	20	None	None	Supervisors
Algorith Implementation	FA2	50	FA1	Laptop	Supervisor
Algorithm Testing	FA3	10	FA2	Laptop	None
Second Algorithm Development	**SA**	**130**	**FA**	**-**	**-**
Finding Weak Points	SA1	15	None	Laptop	Co-supervisor
Improvement Scaffolds	SA2	15	SA1	None	None
Algorith Implementation	SA3	80	SA2	Laptop	Supervisor
Algorithm Testing	SA4	20	SA3	Laptop	None
Documentation	**D**	**100**	**RL,FA,SA,G**	**-**	**-**
Documentation of RL section	D1	10	RL	Laptop	None
Documentation of FA section	D2	20	FA	Laptop	None
Documentation of SA section	D3	20	SA	Laptop	None
Final Documentation	D4	50	D1, D2, D3	Laptop	Supervisors
Total (G + RL + FA + SA + D): 460 hours					

Table 6.1: Summary of Project Planning, self elaborated

6.2 Gantt chart

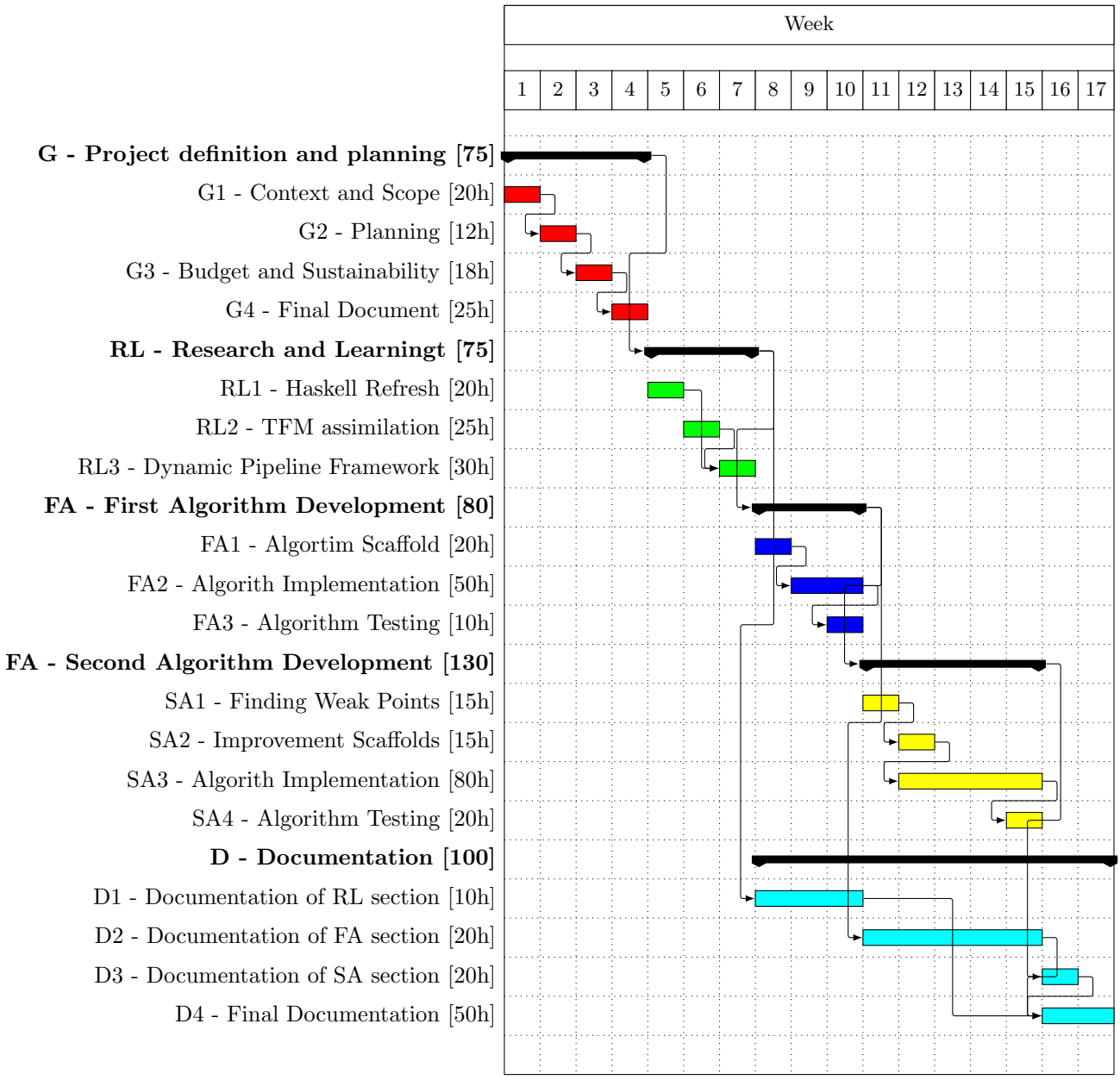

Figure 6.1: Gantt Diagram of this project, self elaborated

Risk management: obstacles and alternatives

As mentioned in previous sections of this work, there are some potential risks that may appear during the development of the project. Here will be explained how I will manage them and the alternatives that could be done in case of critical obstacles.

7.1 Obstacles

Time deadline

As said previously, this is one of the most important risks of the project. There is not a lot of time, but I have some tools to deal with this problem. Firstly, the main objective is to obtain more time if needed and my principal activity (and the one that spends more time) is my job as intern. I have flexibility in my schedule and I can reorganize my week to spend more time in the project. Also, I have some holidays that I can use to spend more time in the project.

Dynamic Pipeline Framework

I explained that I never used before this framework and I may have some problems with it or it could have some bugs or limitations. The principal tool to deal with is to contact with his creator, Juan Pablo. Also, I can spend more time studying the framework because I saved some time in case of this obstacle. This problem could potentially add from 10 to 30 hours to the project.

7.2 Alternatives

If some of the previous obstacles appear and I can not deal with them, I could be forced to make some changes in the project. So the principal alternative is to reduce the scope of the project, reducing the time spent in the last section: the improvement of the algorithm. This part is the longest of the project and it is planned to be able to be reduced if needed. It is very comfortable, since I do not need any additional resource and I can adapt it to the time that I have left.

Budget

In this section I will identify the costs of the project. I will break down the costs into different categories, then I will estimate the costs of each one and finally I will propose a management control system to monitor the costs.

8.1 Identification of costs

Here are all the cost group by categories that I will consider in this project.

Hardware

In my project the only hardware that is needed is a computer. We need a computer for all the process: research, programming, writing the documentation, etc. If we needed to test our algorithm with a large dataset, we maybe would need a computer with a high performance or a more specialized hardware, but in this work I will not consider this case. I will consider a generic computer / laptop.

Software

In this project is only planned to use a few software tools. Starting with the programming language, I will use Haskell. Haskell has a compiler called Glasgow Haskell Compiler (GHC) and would be sufficient for the project. I'm also using Visual Studio Code as a text editor, for both writing the code and the documentation. The next software that I will use is LaTeX, for writing the documentation. I also will use GitHub for the version control of the project, that uses Git.

Human resources

Truthy I'm the only human resource that is working in this project, but I will consider me as a person assuming different roles. As mentinoded in the planning section, I will be taking 3 diferents roles:

- Researcher: For tasks RL2, SA1 and some documentation
- Haskell Developer: For tasks RL3, all FA tasks and all SA tasks
- Project Manager: For all G and D tasks

Other costs

I will not consider other costs as the cost of the internet connection, the electricity and the cost of the paper and ink for printing the documentation. As I consider very specific costs and do not have a huge impact in the total cost. Other cost that could be considered is the cost of the transportation, cost of the supervisor and co-supervisor work and other stuff that could be needed for the project. As I consider that these costs are difficult to estimate, I will not consider them in this project.

8.2 Cost estimates

Now that all costs are identifies, here I will estimate the cost of each one, considering the duration of the project and the cost of each resource.

Hardware

I will consider a standard price for a good laptop, that is around 1000 euros. Obviously a laptop have a longer life than the project duration, so I will consider the cost of the laptop as a cost that is distributed in time, amortization.

$$\text{Amortization} = \text{Laptop Cost} \cdot \frac{\text{Project Duration}}{\text{Laptop Lifespan}}$$

Normaly, laptops have a lifespan of 3 to 5 years with a avegare of 40 hours per week, so I will consider 4 years, 40 hours per week as the lifespan of the laptop. **empty citation** Also I will consider that the whole project will need the laptop, so there will be a total of 460 hours.

$$\text{Amortization} = 1000€ \cdot \frac{460\text{hours}}{4\text{years} \cdot 52\frac{\text{weeks}}{\text{year}} \cdot 40\frac{\text{hours}}{\text{week}}} = 55,29€$$

Software

All the software that I will use is free **empty citation** So the total amount of money spend in software will be 0€.

Human resources

I will be tacking the average salary of each role from glassdoor **GlassDoorResearcher GlassdoorProjectManagerGlas** a well known website for job search and salary information. We will check Spain salaries and we will consider 2000 hours per year. Here we can see the cost of the human resources, assuming a social security multiplier of 1.3.

Role	Avg. Salary (€/h)	Hours	Total (€)	Total with Social Security (€)
Researcher	15	85 hours	1275	1657.5
Haskell Developer	18	200 hours	3600	4680
Project Manager	20	175 hours	3500	4550
Total	18.2	460	8375	10887.5

Table 8.1: Human Resources Cost, self elaborated

Risk plan

Some risks may appear during the project, so I will add some extra money to the budget to cover it.

1. Laptop damage: The laptop could be damaged and need to be repaired. I will consider 100€ for this risk.

2. Extra hours: The project could take more time than expected and each hour of the workers is so expensive. Considering 10 % more time, gives us 837 €, so I will add 1000€ to the budget.

3. For other risks and unexpected costs, I will add 10 % of the total budget, to cover it, 1100 €.

Puting all together, the total amount of money for the unexpected costs is 2037 €.

Total budget

Lets sumarize all the costs and calculate the total budget.

Source	Cost (€)
Hardware	55,29
Software	0
Human Resources	10887.5
Unexpected	2037
Total	**12979.79**

Table 8.2: Total budget, self elaborated

8.3 Management control

Once we have our budget, I can create a indicator so I can monitor the deviation from the initial plan. This indicator (C) will be the difference between the expected and actual cost:

$$C = C_estimado - C_real$$

I can use this indicator both in a general way and with each subsection of the budget, in order to be able to analyze what may be causing a variation in the budget and be able to fix it, either by modifying the budget itself or by restructuring the project.

Sustainability report

9.1 Economic Dimension

After analized my project and made the budget, we can observe that the main cost of the project is the human resources. Asuming that my plannign is correct, I can confirm that no unnecessary costs are being generated, beacause I'm using free software and the minimun hardware required. Also this project have the goal of improving an implementation of an algorithm, meaning a possible improvement in the efficiency of the algorithm, which could lead to a reduction in time, which means a reduction of the cost.

9.2 Environmental Dimension

As mentioned before, the only environmental impact of the project is the energy consumed by the laptop. The energy and impact are minimum, so we can consider this project as environmentally friendly. Also as previous point, the project could lead to a reduction in time, which means a reduction of the energy consumed. A good point about projects like mine is that their useful life can be considered 'unlimited'. This is because it is a research and development project, providing utility at all times.

9.3 Social Dimension

In personal terms, I think it will have a great impact on my life and career as a student, since this project will be the first 'big project', which will help me learn from mistakes for future projects. This project may not have a big social impact, but it can do your part in the world of computing and especially in the world of Haskell programming.

Bibliography

[1] Juan Pablo Royo Sales, "Incremental algorithm for large netwotks," 2021. [Online]. Available: https://upcommons.upc.edu/bitstream/handle/2117/361615/163090.pdf?sequence=1.

[2] A. M. Sharp, *Incremental algorithms: solving problems in a changing world*. Citeseer, 2007.

[3] "What is scrum? — scrum.org." (), [Online]. Available: https://www.scrum.org/resources/what-scrum-module (visited on 02/27/2024).

[4] "Git." (), [Online]. Available: https://git-scm.com/ (visited on 02/27/2024).

[5] A. Herrero, "FACULTAT INFORMATICA DE BARCELONA (FIB) UNIVERSITAT,"

[6] "Haskell." (), [Online]. Available: https://www.cs.upc.edu/~jpetit/Haskell/#1 (visited on 03/04/2024).

[7] J. P. R. Sales, *Jproyo/dynamic-pipeline*, original-date: 2021-03-27T11:12:27Z, Jul. 21, 2021. [Online]. Available: https://github.com/jproyo/dynamic-pipeline (visited on 03/04/2024).

[8] Pol, *Polforner/TFG*, original-date: 2024-03-04T09:51:17Z, Mar. 4, 2024. [Online]. Available: https://github.com/polforner/TFG (visited on 03/04/2024).

[9] "What is the average lifespan of a computer?" (), [Online]. Available: https://www.hp.com/in-en/shop/tech-takes/post/average-computer-lifespan (visited on 03/11/2024).

[10] "Salary: Researcher in spain 2024," Glassdoor. (Mar. 10, 2024), [Online]. Available: https://www.glassdoor.com/Salaries/spain-researcher-salary-SRCH_IL.0,5_IN219_KO6,16.htm (visited on 03/11/2024).

[11] "Salary: Project manager in madrid, spain 2024," Glassdoor. (Mar. 11, 2024), [Online]. Available: https://www.glassdoor.com/Salaries/madrid-project-manager-salary-SRCH_IL.0,6_IM1030_KO7,22.htm (visited on 03/11/2024).

[12] "Salary: Software developer in spain 2024," Glassdoor. (Mar. 1, 2024), [Online]. Available: https://www.glassdoor.com/Salaries/spain-software-developer-salary-SRCH_IL.0,5_IN219_KO6,24.htm (visited on 03/11/2024).