UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

Incremental Algorithm for large Netwotks

Project management (GEP)
Deliverable 2: Planning of the project

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

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 [1]. 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:

1.1 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 1.1**) 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.
  Minimum hours: 20 hours

- **G2** Planning
  This corresponds to the second deliverable of GEP, where is defined the planning of the project.
  Minimum hours: 12 hours

- **G3** Budget and Sustainability
  This corresponds to the third deliverable of GEP, where is defined the budged and sustainability of the project.
  Minimum hours: 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.
  Minimum hours: 25 hours

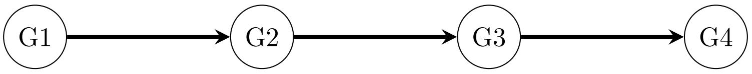

Figure 1.1: Task Dependencies for Project definition and planning, self elaborated

1.2 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 [2], Haskell notes made by Jordi Petit from the subject LP [3] and the Dynamic Pipeline Framework repository [4]. 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 1.2**) 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
  Minimum hours: 20 hours

- **RL2** TFM assimilation
  The reading and understanding of Juan Pablo's work.
  Minimum hours: 15 hours

- **RL3** Dynamic Pipeline Framework
  Intensive review of the framework repository.
  Minimum hours: 20 hours

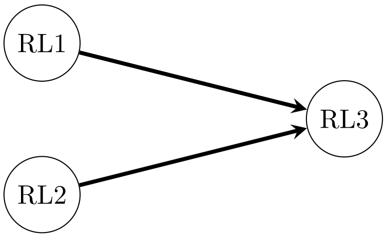

Figure 1.2: Task Dependencies for Research and Learning, self elaborated

1.3 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 [5]. 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 1.3**), 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
  Minimum hours: 15 hours

- **FA2** Algorithm Implementation
  This part is the pure implementation of the algorithm.
  Minimum hours: 35 hours

- **FA3** Algorithm Testing
  Testing of the algorithm implementation, littles changes are also contemplated here (no big changes)
  Minimum hours: 5 hours

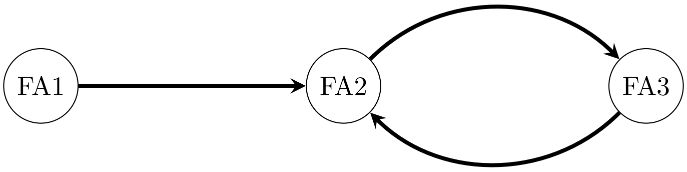

Figure 1.3: Task Dependencies for First Algorithm Development, self elaborated

1.4 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 1.4**), 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.
  Minimum hours: 10 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
  Minimum hours: 5 hours

- **SA3** Algorithm Implementation
  This part is the pure implementation of the algorithm.
  Minimum hours: 70 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
  Minimum hours: 10 hours

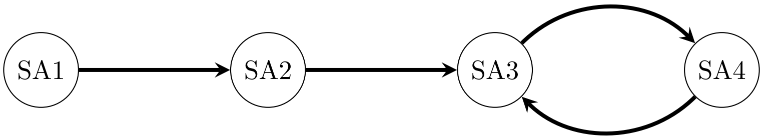

Figure 1.4: Task Dependencies for Second Algorithm Development, self elaborated

1.5 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 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 1.5**).

- **D1** Documentation of RL section
  This corresponds to the documentation of the research and learning section.
  Minimum hours: 10 hours
- **D2** Documentation of FA section
  This corresponds to the documentation of the first algorithm section.
  Minimum hours: 20 hours
- **D3** Documentation of SA section
  This corresponds to the documentation of the second algorithm section.
  Minimum hours: 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.
  Minimum hours: 30 hours

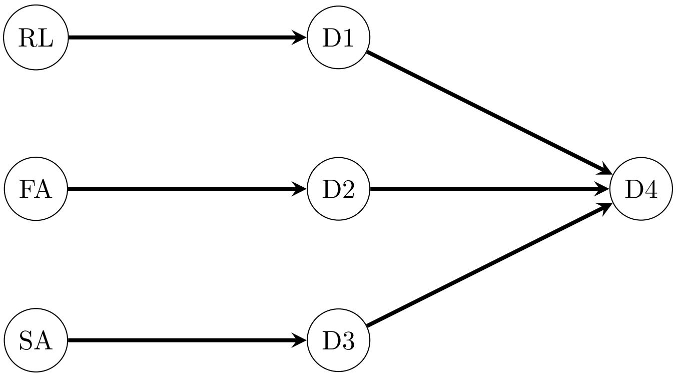

Figure 1.5: Task Dependencies for Documentation, self elaborated

Estimations and Gantt chart

2.1 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 2.1: Summary of Project Planning, self elaborated

2.2 Gantt chart

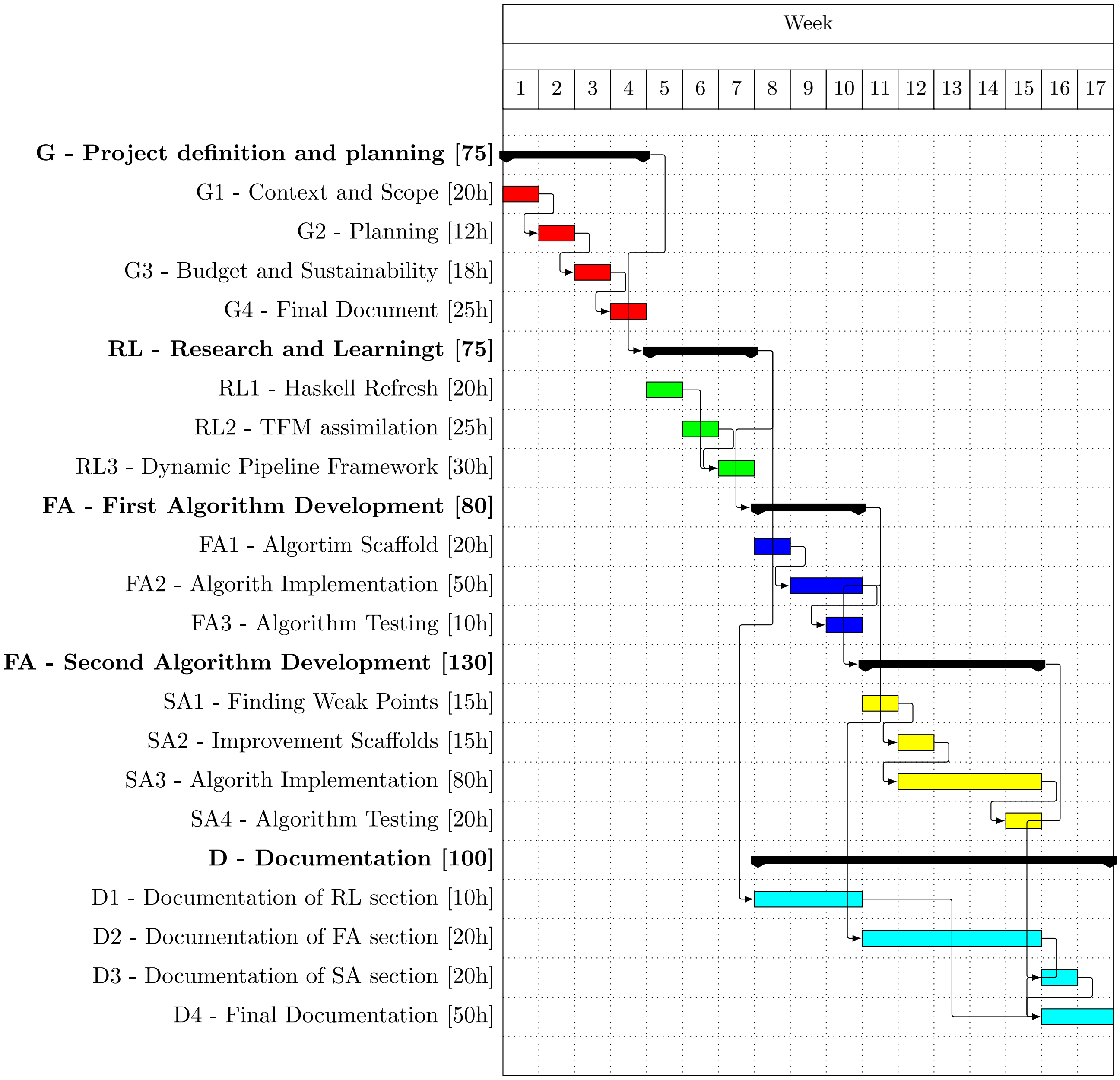

Figure 2.1: Gantt Diagram of this project, self elaborated

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.

3.1 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.

3.2 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.

Bibliography

[1] A. Herrero, “FACULTAT d’INFORMÀTICA DE BARCELONA (FIB) UNIVERSITAT,”

[2] E. Pasarella and M.-E. Vidal, “FACULTAT d’INFORMÀTICA DE BARCELONA (FIB),”

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

[4] 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).

[5] 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).