DEVELOPMENT OF A MULTI-OBJECTIVE RESOURCE-CONSTRAINED SCHEDULING METHOD FOR THE UTILITY CONSTRUCTION SECTOR

AUTHOR: S.W.L.G. (STEF) KOSTERS
S2394685 - S.W.L.G.KOSTERS@STUDENT.UTWENTE.NL

LEAD SUPERVISOR: PROF. DR. IR. M.R.K. (MARTIJN) MES
SECOND SUPERVISOR: DR. IR. J.M.J. (MARCO) SCHUTTEN
EXTERNAL SUPERVISOR: T. (TIJMEN) LINTSEN
EXTERNAL ADVISOR: DR. IR. R.H. (ROB) BEMTHUIS

26/08/2024

UNIVERSITY OF TWENTE.

Preface

Dear reader,

You are about to read my Master's thesis "Development of a Multi-Objective Resource-Constrained Scheduling Method for the Utility Construction Sector". This research was conducted at CAPE Groep in Enschede in collaboration with Aannemingsmaatschappij Hegeman B.V. in Nijverdal. This research is the final assignment for my Master's in Industrial Engineering and Management with a specialisation in Production and Logistics Management at the University of Twente.

At CAPE Groep, I had a great time during which I learned and developed a lot. First, I would like to thank all my colleagues at CAPE Groep for their interest and the pleasant working atmosphere. I especially want to thank Tijmen Linsen for being my supervisor at CAPE Groep and assisting with the progression of my thesis. I also want to thank the people at Hegeman for their confidence and the valuable sessions we had. Special thanks to Jarno Marsman for the valuable sessions and all the information I needed from Hegeman to conduct this research.

I would also like to thank Martijn Mes for being my first supervisor and guiding me in the right direction during meetings and feedback sessions. Additionally, I want to thank Marco Schutten for being my second supervisor and for the valuable insights and feedback provided. Furthermore, I want to thank Rob Bemthuis for his role as an external advisor, providing valuable input during brainstorming sessions and contributing to the creation of this assignment.

Lastly, I want to thank my friends, girlfriend, and family for supporting me during the execution of this research.

I hope you enjoy reading my Master's thesis!

Stef Kosters

Enter, August 2024

Management Summary

Aannemingsmaatschappij Hegeman B.V. faces challenges due to the absence of a standardised methodology for creating and revising operational schedules in utility construction projects. Currently, executors use their own methods with varying levels of quality. This makes operational schedules highly dependent on individual experience. The lack of a standardised methodology causes problems in project progression which causes delays in delivery dates. This research aims to develop methods to create and revise schedules that should help the executors.

Methodology

To efficiently create and revise schedules, this research develops two different methods. The first method creates operational schedules. The second method revises operational schedules after disruptions.

The first method develops the Multi-Objective Resource-Constrained Critical Path Method (MORC CPM). This MORC CPM develops an operational schedule while considering precedence relations and resource constraints. The MORC CPM consists of five phases. Phase 1 creates an operational schedule without taking into account capacity constraints and only considering precedence relations. Phase 2 uses a constructive heuristic to make the schedule feasible when introducing capacity restrictions. Phase 3 aims to improve the schedule using simulated annealing. The objective function minimises the project duration, and resource variability while maximising the resource utilisation. Resource variability is defined as the fluctuations in resource allocation. Phase 4 derives personnel and crane schedules from the operational schedule. Crane schedules allocate crane usage to specific activities. Phase 5 assigns activities to idle employees to do some (preparing) work.

The second method in this research is the Schedule Revision Method (SRM). The SRM assists executors in revising operational schedules after a disruption in case simple solutions such as working overtime or renting more equipment cannot solve the disruption. The SRM aims to make the schedule feasible again with minimal changes, avoiding the need for extensive rework by the executor and minimising project duration.

Key findings

The experiments conducted with the MORC CPM and the SRM demonstrate the following:

- Both methods are effective in enhancing schedule efficiency and reliability. The MORC CPM optimises resource utilisation and reduces project duration. The SRM effectively handles disrupted activities with minimal impact on the current schedules.
- MORC CPM:
    - The running time increases exponentially with the number of activities indicating scalability issues. Doubling the number of activities can lead to a tenfold increase in computation time.
    - Increasing employee capacity reduces project duration and improves resource utilisation. However, beyond a certain point, it is not beneficial to add more employees since this causes other resources to become the bottleneck. In that case, adding more employees leads to lower resource utilisation.
    - There is a high correlation between the different parts of the objective function. This means that optimising one part positively influences the other parts.

- The SRM gives more efficient solutions than delaying the whole schedule because of the disruption.

Recommendations

This research leads to several recommendations and directions for future research:

- Investigate whether it is possible to develop a method based on the MORC CPM to create an overall plan for the tactical level.
- ECOLOGIC can investigate the integration of the scheduling methods into the digital twin. This integration could improve the quality of the actions the digital twin proposes.
- Conduct research on implementing these methods company-wide and provide training to the users.
- Iteratively improve the methods based on feedback from the executors. Use a feedback mechanism to collect feedback.

Conclusion

Executors at Hegeman will benefit from the methods and tools for creating and revising operational schedules. These methods improve efficiency and reliability, positively impact project progression, and reduce delays in delivery dates. The quantitative improvements the methods provide vary across different problem instances. The development of the methods thus tackles the action problem of the frequent occurrence of delayed project delivery dates.

List of Figures

List of Tables

List of Abbreviations

- **Hegeman**: Aanemingsmaatschappij Hegeman B.V.
- **SfP**: Software for Planning
- **CPM**: Critical Path Method
- **RCPSP**: Resource-Constrained Project Scheduling Problem (RCPSP)
- **MORC CPM**: Multi-Objective Resource-Constrained Critical Path Method
- **SA**: Simulated Annealing
- **EST**: Earliest Starting Time
- **EFT**: Earliest Finishing time
- **LST**: Latest Start Time
- **LFT**: latest Finish Time
- **PF**: Predecessor Fraction
- **SF**: Successor Fraction
- **RCS**: Resource-Constrained Scheduling
- **SRM**: Schedule Revision Method

Table of Contents

1 Introduction

This chapter introduces the research on the development of scheduling methods for the utility construction sector. Section 1.1 introduces the ECOLOGIC project and the companies involved in this research. Section 1.2 describes the motivation for the research from different perspectives. Section 1.3 analyses the problem context by identifying the action problem and core problem. Section 1.4 discusses the research objective. Lastly, Section 1.5 dives into the research design.

1.1 ECOLOGIC and Company Introduction

Section 1.1.1 introduces ECOLOGIC which is the project this research contributes to. Section 1.1.2 presents CAPE Group, which is the company for which this research is conducted. Section 1.1.3 introduces Aannemingsmaatschappij Hegeman B.V., which is the company providing the case and has the highest interest in this research.

1.1.1 ECOLOGIC

ECOLOGIC is a research project focusing on solving logistical challenges in the construction sector. The aim is to do this by using smart data technologies with the help of a carbon digital twin. A carbon digital twin is a digital replica of a physical object, process or system that includes tracking and simulating carbon emissions and their impact. The digital replica continuously updates with data from its physical counterpart allowing for simulations to predict performance and behavior. Applying simulation techniques enhances predictability and supports decision-making (Van der Horn & Mahadevan, 2021).

The carbon digital twin to be developed in the ECOLOGIC project should enable the identification of disruptions in planning and execution through data integration and new AI-based models. ECOLOGIC aims to improve sustainability in the Dutch construction sector by developing data-driven insights and advanced analysis techniques. The objective of the ECOLOGIC project is as follows: "To improve the competitiveness, resilience, and sustainability of the Dutch construction logistics industry by developing and demonstrating reliable data-driven insights and advanced analytics techniques for an anticipatory and adaptive logistics planning to minimise carbon footprint while optimising operations.". ECOLOGIC is a project involving the University of Twente and many companies which have an interest in the construction sector. Two of the involved companies are CAPE Groep and Aannemingsmaatschappij Hegeman B.V. This thesis is being executed in the interest of these two companies.

The ECOLOGIC project started at the end of 2023 and is planned to be finished at the end of 2026. The project includes multiple work packages each involving different companies. The objectives of ECOLOGIC are:

1. Develop and demonstrate innovative carbon digital twin concepts.
2. Demonstrating the actual functioning of the disruption management system, with the goal of achieving greater visibility and control over multi-objective emission reduction goals.
3. Creating value propositions that aim to disseminate and value knowledge

1.1.2 CAPE Group

This research is being conducted at CAPE Groep in the interest of Aannemingsmaatschappij Hegeman B.V. CAPE Groep is a distinguished software development and system integrator consultancy company with its headquarters in Enschede, a city in the east of The Netherlands. CAPE Groep specialises in delivering agile model-driven solutions to the Dutch logistics industry and helps organisations transform digitally. CAPE Groep mainly focuses on the sectors of Transport & Logistics, Supply Chain, Agrifood, and Smart Construction. CAPE Groep often sees opportunities in research projects like ECOLOGIC since CAPE Groep sees this as a chance to gain a competitive advantage over its competitors.

1.1.3 Aannemingsmaatschappij Hegeman B.V.

Aannemingsmaatschappij Hegeman B.V. (Hegeman) is a well-established construction company operating in the Civil and Utility Construction Sector. The company is located in Nijverdal, a village in the east of The Netherlands. Hegeman is a family business in which currently the fourth generation is leading the company. Hegeman was founded in 1927 and since then it has grown towards a large company executing large projects.

The projects Hegeman executes vary a lot and include, for example, the construction of care homes, nursing homes, schools, offices, shopping centres, underground parking garages, churches, industrial halls, factories and much more. By valuing people and prioritising quality, Hegeman has maintained its reputation as a reliable and innovative construction partner.

1.2 Research Motivation

The largest benefits from this research are for Hegeman. The scheduling methods and tools this research creates draw on information from Hegeman's executors and project leaders. Moreover, the methods should be developed such that Hegeman can effectively use them. Despite that with slight modifications these methods can probably be used industrywide, they are specifically for the needs of Hegeman.

From the perspective of ECOLOGIC, this research aims to contribute to getting more efficient and resilient construction sites. So, this research contributes to the objective of ECOLOGIC discussed in Section 1.1.1. Moreover, this research can be a building block in the development of a fully working digital twin. Scheduling methods should cause schedules to become more efficient, which could be a useful addition to the digital twin.

CAPE Groep considers this research exploratory, leveraging their experience in the construction sector to broaden their expertise and capabilities. This study provides CAPE Groep with increased engagement in the construction industry, enhancing their knowledge and connections.

1.3 Problem Context

This section analyses the problem context. First, Section 1.3.1, discusses the action problem. Next, Section 1.3.2 dives into the problem cluster. Lastly, Section 1.3.3 identifies and discusses the core problem derived from the problem cluster.

1.3.1 Action Problem

The utility construction sector is a highly complex sector to operate in (Elsayegh & El-Adaway, 2021; Majumder et al., 2022; Viklund Tallgren et al., 2020). Utility construction projects involve the construction of buildings for non-residential use. Each project is unique, making it more challenging to learn from past mistakes compared to housing projects where projects have more similarities.

Action problem: Frequent occurrence of delayed project delivery dates.

The action problem emerges from discussions with two executors (in Dutch: uitvoerders) and a project leader. Meeting the delivery date is crucial since postponing the delivery date can be costly for Hegeman. The norm here is that the delivery date is always met. However, the reality is that this is not always the case and the planned delivery date often has to be delayed multiple times during a project.

Before the start of a project, the client sets a delivery date in the tender. Then, construction companies try to get the project by making an offer. Next, the client chooses a construction company that may execute the project. When Hegeman thinks the delivery date is unrealistic, they mention this in the offer. In general, the delivery date proposed by the client is the delivery date in the contract.

1.3.2 Problem Cluster

Figure 1.1 shows the problem cluster of this problem context. The action problem described in Section 1.3.1 is in the red box on the right of Figure 1.1. This action problem has several underlying causes. Some of them are influenceable in this context, others are not. This section elaborates on this problem cluster and dives into each box separately.

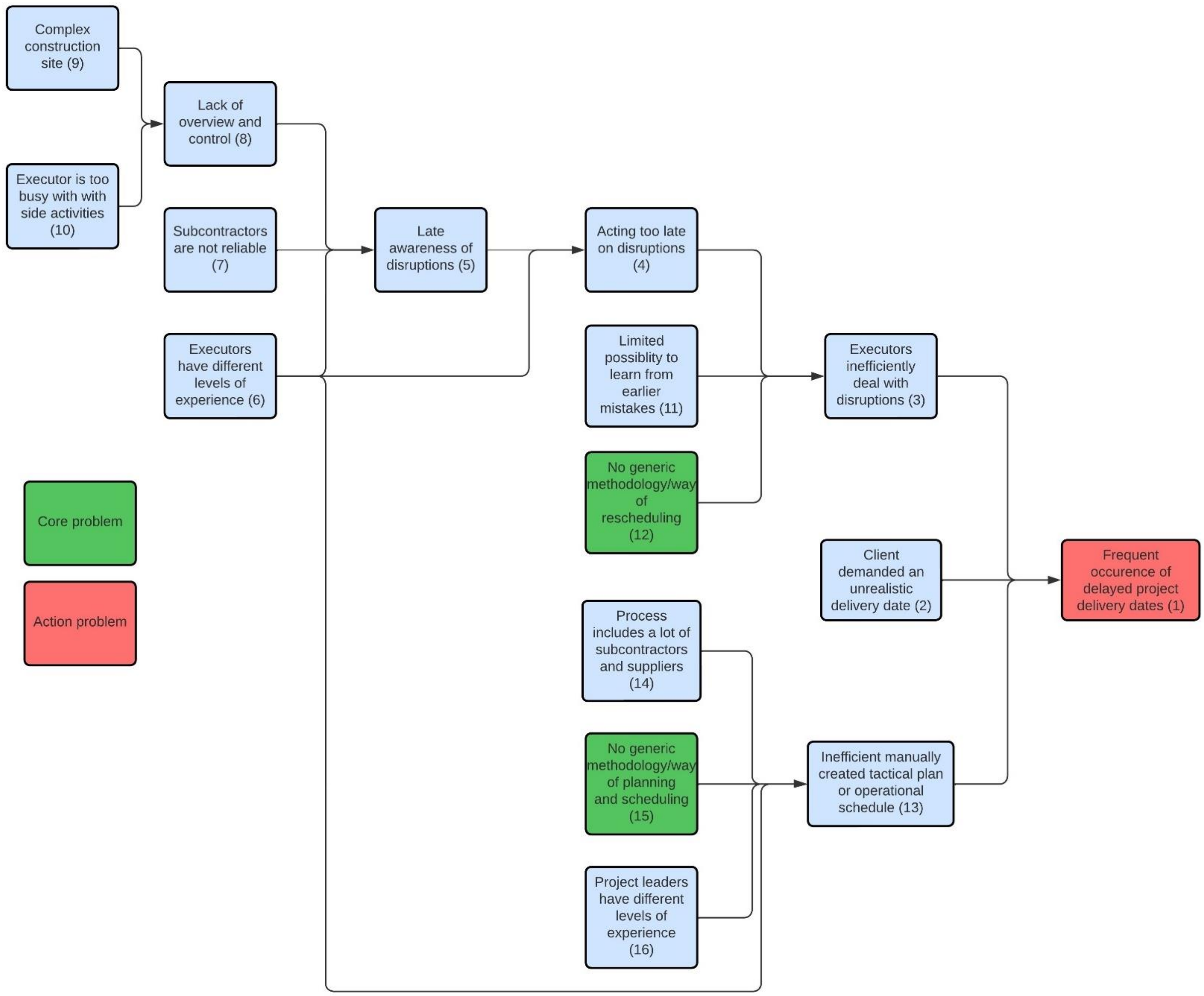


Figure 1.1 - Problem Cluster

It is important to consider that these observations are specific to Hegeman. However, as the core activities are equal across different utility construction companies, the problem cluster can be representative of challenges faced by other utility construction companies.

As mentioned in Section 1.3.1, the action problem is the frequent occurrence of delayed project delivery dates (1)[1]. This action problem has three possible direct causes. The first possible direct cause is that the client demands an unrealistic delivery date (2). If this is the case, Hegeman mentions this in their offer and it is up to the client how to assess this.

[1] For readability, behind each box in Figure 1.1 is a number in between brackets to make this section more readable.

The second possible direct cause of the action problem is that executors inefficiently deal with disruptions (3). This can have three possible direct causes. The first possible direct cause for executors inefficiently dealing with disruptions is that they act too late on disruptions (4). There are a lot of possible disruptions including bad weather, delivery delays, breakdowns, conflicts, and labour shortages, but also external factors like policy changes. A possible cause of late intervention is the late awareness of disruptions (5). This has again three possible reasons. The first possible reason is that executors have different levels of experience (6). Some executors have years of experience and know better how to signal and deal with certain disruptions. Other executors have less experience in this field and do not know how to properly deal with these disruptions. In these cases, it is more likely that the delivery date needs to be delayed. The second direct cause of late awareness is that subcontractors are not always reliable (7). Subcontractors do not always care about whether they work according to the schedule made by the executor. For the subcontractors, project delay is not their problem, but for Hegeman, this can cause large problems. The third possible direct cause of signalling disruptions too late is that there is a lack of overview and control in the construction site (8).

The lack of overview could have many reasons, but this discussion is limited to the two most likely causes. The first possible cause is that utility construction sites are highly complex (9). This is something that must be accepted since it cannot be influenced. The complex construction sites are a characteristic of the utility construction sector. The construction sites are among other things complex because of the high number of parties involved, the diverse interests, policies, complex contracts, and their uniqueness (Sears et al., 2008). The second possible cause for the lack of overview and control is that the executor can be too busy with side activities (10). This can lead to a lot of inconvenience for the executors. The executors would like to see that they become less busy with side activities like time registration for the employees. This is such an important problem that another student is going to do a bachelor's assignment on this at CAPE Groep.

The second possible direct cause of why executors inefficiently deal with disruptions is that there is a limited possibility of learning from earlier mistakes (11). This is because most projects executed by Hegeman are unique and include different challenges although some projects do share similarities. The third possible direct cause of inefficiently dealing with disruptions is that there is no generic methodology or way of rescheduling (12). When rescheduling is necessary, executors manually make changes in the schedule. On top of that, some executors tend to easily delay all activities. This can have disastrous causes for meeting the delivery date. So, the existence of a generic methodology or way of rescheduling can help in this.

The third direct cause of the action problem is that there can be an inefficient manually created overall plan or operational schedule (13). Not all plans and schedules are inefficient but since they are created manually by the executor, these plans and schedules are not often (close to) optimal. Having an inefficient manually created plan or schedule can have four possible reasons within the scope of this research. The first possible reason is that the process includes a lot of subcontractors and suppliers (14). This makes creating an operational schedule complex since the other parties also have capacity limits and their own plans and schedules. This gives the executor of Hegeman additional constraints in creating their schedule. The second possible cause of an inefficient plan or schedule is that there is no generic methodology and way of planning and scheduling that the project leaders and executors use (15). Each executor has its own way of working, some of them are more efficient than others. This is also inconvenient in cases of disruptions. Another possible cause for an overall plan being inefficient is that project leaders have different levels of experience (16). Finally, a possible cause of having an inefficient operational schedule can be that executors have different levels of experience (6).

1.3.3 Core Problem

First, this section identifies the core problem. Next, this section explains the core problem.

Core Problem Identification

The core problem should be a problem that has no direct cause (Heerkens & Van Winden, 2017). This left us with the following possible core problems:

- Clients demand unrealistic delivery dates (2)
- Executors have different levels of experience (6)
- Subcontractors are not reliable (7)
- Complex construction site (9)
- Executor is too busy with side activities (10)
- Limited possibility to learn from earlier made mistakes (11)
- No generic methodology/way of rescheduling (12)
- Process includes a lot of subcontractors and suppliers (14)
- No generic methodology/way of planning and scheduling (15)
- Project leaders have different levels of experience (16)

As mentioned in Section 1.3.2, some problems are influenceable, some are not. First of all, problem (2) is limitedly influenceable since the client determines what delivery date he expects. However, Hegeman can mention it in the offer but it is always up to the client what he thinks of it. This could cause Hegeman not to get the job or Hegeman has to deal with a tight delivery date. Problem (9) is a characteristic of the utility sector that cannot be changed. For the same reason, problem (11) is something cannot be changed and has to be dealt with. Problem (14) is limitedly influenceable because subcontractors and suppliers have limited ability to adjust their plans, schedules, or capacity. Subcontractors and suppliers are important in construction projects. Hegeman has some influence on this, but this is not suited as a core problem to tackle the action problem. Regarding problems (6) and (16), there is limited influence on the hiring process but this is not influenceable from a problem-solving perspective.

The possible core problems remaining are (7), (10), (12), and (15). As mentioned in Section 1.3.2, problem (10) is such an important problem that a student is going to do a bachelor assignment on this at CAPE Groep. Problem (7) is limitedly influenceable. Hegeman can aim to find reliable subcontractors regarding planning and scheduling issues. However, there are more important requirements to consider when choosing a subcontractor. Therefore, problem (7) cannot be solved in this context.

This left us with two possible core problems. The core problems are related in that a generic methodology or way of working is missing. Addressing both problems simultaneously allows tackling two problems with less effort than when tackling individually. Hence, the two core problems of this research are problems (12) and (15). Formulating these two problems together leads to one core problem. Regarding problem 15, the focus specifically lies on scheduling and thus not on planning.

Core problem: Executors lack generic methods for creating and revising operational schedules in utility construction projects, negatively influencing the efficiency and reliability of project delivery.

Core Problem Explanation

The core problem in this research is that executors use no generic methods for creating and revising operational schedules in utility construction projects. Having no generic scheduling methods impacts the efficiency and reliability of project delivery.

Creating operational schedules is a critical task in the construction of buildings. Integrating generic methods for creating and revising schedules provide executors guidance in their work. This will mainly help executors with little experience.

The creation of an operational schedule is a recurring task, whereas an overall plan is created once. When the client chooses Hegeman as the construction company to do the project, the project leader starts by identifying several milestones for the construction project. These milestones are intermediate deadlines and consist of tasks that are in between different building phases. These milestones include, for example, the completion of the cement screed, the finalisation of the steel structure, the achievement of wind- and watertight status, and the connection of power. Based on these milestones, the project leader creates an overall plan including all tasks for the whole project. This plan does not yet include details but it is a rough plan.

When the project has started, the executor continuously keeps up an operational schedule, which is more detailed than the overall plan created by the project leader. This operational schedule includes each specific activity that should be performed and shows which project team or subcontractor has to perform this activity. The executor creates these schedules manually. When there is a disruption in the process, the executor adapts the schedule manually. Therefore, creating and revising the operational schedule highly depends on the experience of the executor. There is no generic methodology for creating this schedule. So, whether the delivery date can be met, among other things, depends on the quality of work from the executor.

The core problem is sufficiently solved when methods increase the efficiency and reliability of operational schedules eventually causing increased efficiency and reliability in project delivery dates.

1.4 Research Objective

This section discusses the research objective. The core problem of this research, as discussed in Section 1.3.3, is that executors lack generic methods for creating and revising operational schedules which has a negative influence on the efficiency and reliability of project delivery. This research wants to tackle this core problem to solve the action problem discussed in Section 1.3.1. When the core problem has been tackled, the frequency of project delivery date delays should decrease. To tackle this problem, this research aims to develop methods that can support scheduling decisions.

Research objective: Develop operational scheduling methods to create and revise schedules, aiming to enhance efficiency and reliability in meeting delivery dates for construction projects.

This research objective has some secondary advantages as well. First, finishing earlier and working more efficiently leads to reduced CO_2 emissions. By that, society profits from this research as well. Moreover, finishing earlier increases client satisfaction since the client can use the building earlier. Next, when this research is a success, for Hegeman, this leads to lower costs of not meeting delivery dates. For the executors, this research has advantages as well. It gives them better control over the plan and schedule and they know there is a system to support them. Moreover, this can simplify their work and can increase job satisfaction. Furthermore, this research can lead to competitive advantages by enabling Hegeman to complete projects earlier, reducing costs, and improving their position in winning tenders.

1.5 Research Design

This section outlines the design of the research. Section 1.5.1 discusses the approach of the research. Section 1.5.2 outlines relevant research questions to reach the goal of the thesis. These research questions can be seen as a guide through the thesis. Moreover, Section 1.5.2 shows the thesis outline and each research question is linked to a chapter here. Section 1.5.3 discusses the scope of the research. Finally, Section 1.5.4 discusses the deliverables of the research.

1.5.1 Approach

After focussing on the problem context in Chapter 1, the focus shifts to the current situation in Chapter 2. Next, in Chapter 3, the literature review will lay the foundation of the research. This literature review aims to identify and discuss existing scheduling methods and techniques and challenges in construction scheduling which can be useful as input for the subsequent chapters. Chapter 4 develops the method for creating operational schedules. Chapter 5 focuses on developing a method for revising the operational schedules. Chapter 6 discusses how both methods can be implemented in tools. Chapter 7 experiments with the method for creating operational schedules. Lastly, Chapter 8 performs experiments with the method for revising operational schedules.

1.5.2 Research Questions & Thesis Outline

This section presents the key research questions and outlines the structure of the thesis. These research questions guide the objectives of each chapter and together contribute to answering the main research question.

Main Research Question

The research focuses on both creating and revising operational schedules. The main research question mainly focuses on the design phase of this research.

Main RQ: How should the methods for creating and revising operational schedules be designed such that they are practically applicable to the users?

Chapter 2 – Current Situation

RQ CH2.1: What is the current way of planning and scheduling, and what challenges do project leaders and executors face in these processes?

RQ CH2.2: What are the components of the overall plan and the operational schedule and how does the planned progress compare with actual project realisations?

Chapter 2 analyses the current situation and discusses the challenges of the current processes at both tactical and operational levels, as these are highly interrelated. The chapter begins with an overview of Hegeman's organisational structure. Next, the chapter analyses the current way of creating an overall plan. After that, it discusses the current way of creating an operational schedule and operational challenges. Finally, the chapter identifies all relevant stakeholders in the planning and scheduling process.

Chapter 3 – Literature Review

RQ CH3: What planning and scheduling methods exist in the utility construction sector?

Chapter 3 is a literature review that serves as a basis for building the scheduling methods. It investigates various challenges in construction project planning and scheduling and dives into existing methods.

Chapter 4 – Schedule Creation Method Development

RQ CH4: What is an effective method for creating operational schedules that is practically applicable to users?

Chapter 4 develops the method for creating operational schedules. This chapter starts with a general problem description. After that, this chapter describes the inputs necessary for the method. Next, it discusses the assumptions and simplifications. Lastly, the method will be developed.

Chapter 5 – Schedule Revision Method Development

RQ CH5: What is an effective method for revising operational schedules that are practically applicable to users?

Chapter 5 develops a method for revising operational schedules in case of a disruption. This method should help mitigate the impact of disruptions on the project progression. The method should result in a feasible schedule that maintains realistic timelines and resource availability.

Chapter 6 – Implementation

RQ CH6: How can the Schedule Creation Method and the Schedule Revision Method be effectively implemented in a practical tool?

Chapter 6 focuses on integrating the Schedule Creation Method and the Schedule Revision Method into two separate tools. The integration of these methods into tools allows executors to use the methods.

Chapter 7 – Experimental Analysis – Schedule Creation Method

RQ CH7: How does the Schedule Creation Method perform and what is the quality of the output schedules?

Chapter 7 aims to validate the method for creating operational schedules. The method will be applied to test data. This chapter functions as a validity and reliability check.

Chapter 8 – Experimental Analysis – Schedule Revision Method

RQ CH8: How effective is the Schedule Revision Method in minimising the impact of disruptions on project schedules compared to current methodologies?

Chapter 8 assesses the performance of the Schedule Revision Method through an experimental analysis. This chapter uses real-case scenarios to compare the Schedule Revision Method to make a valid conclusion.

1.5.3 Scope

In utility construction projects, it is common for the project leader to create an overall plan and for the executors to create operational schedules. This research is probably applicable industry-wide, however, this research specifically focuses on Hegeman. Hegeman is involved in the development of the methods and tools. Based on the experience of an executor and project leader, the methods and tools will be developed. So, the scope of this research is limited to addressing the specific challenges and requirements within Hegeman.

1.5.4 Deliverables

The main deliverables of this thesis include the following:

1. **Two scheduling methods**: The main deliverables of this research are two scheduling methods. The methods together aim to streamline the scheduling process for utility construction projects in Hegeman. One method creates schedules and the other method revises schedules.
2. **Two scheduling tools**: To apply the methods in practice, both methods are supported by user-friendly scheduling tools that facilitate the creation and revision of operational schedules.
3. **This thesis document**: This thesis is an important deliverable as well. This thesis should effectively and systematically outline the problems and try to solve them. Moreover, it includes findings, analysis and conclusions. This should be a useful overview of the current process for Hegeman.

2 Current Situation

This chapter outlines the current situation of the planning and scheduling process. First, Section 2.1 explains the organisational structure in Hegeman. Next, Section 2.2 describes the different types of plans and schedules Hegeman uses for their projects. After that, Section 2.3 details how the project leader creates an overall plan. Section 2.4 describes how the executor creates an operational schedule. Section 2.5 discusses the operational challenges and disruptions an executor can face. Lastly, Section 2.6 identifies the stakeholders involved. Most information in this chapter is gathered from conversations with an executor and project leader from Hegeman.

2.1 Organisational Structure

Hegeman is a construction company operating in the utility construction sector. Hegeman executes multiple projects at the same time. Because these projects are highly complex, each project involves multiple employees of Hegeman. This section highlights the main people involved in the planning and scheduling process. Section 2.6 broadens this by looking at all the stakeholders interested in this research. For every project there are, in general, three different roles to be occupied:

- **Project leader**: The project leader creates the overall plan upfront before the project starts. Moreover, the project leader is the contact person for the client during the preparation, realisation, and aftercare phases of the project. Together with the client, the advisors, and the subcontractors, the project leader completes all these project phases and gives daily support to the work preparator.
- **Work preparator**: The work preparator is responsible for the technical preparation of the project and corresponding activities. The work preparator mainly focuses on the drawings of the building and the budget of the whole project. The work preparator does not have much to do with planning and scheduling but plays an important role in preparing the project.
- **Executor**: The executor is responsible for creating the operational schedules. He does this approximately every three to four weeks and schedules at least six weeks upfront (often much more). Based on the operational schedule, the executor creates a personnel schedule and a crane schedule. The personnel schedule and crane schedule will be explained later in this section.

When a project is large or complex, it could be the case that some roles need to be split up and they use for example an assistant executor.

Hegeman distinguishes two phases in construction projects, which include rough construction and finishing (in Dutch: ruwbouw & afbouw). Rough Construction is the initial phase focusing on the construction of the main structure of the building. Finishing is the second phase focusing on completing the interior and exterior of the building. This includes things like flooring, painting, tasks that have to do with electricity, and much more. In the finishing phase, the building transforms from an unfunctional empty building to a functional, designed and finished building. Ideally, the finishing phase starts after the rough construction phase is finished. However, due to the necessity to finish the project on time, some tasks of the finishing phase can already start before the rough construction phase is finished.

2.2 Types of plans and schedules

In terms of planning and scheduling, Hegeman makes use of different types of plans and schedules:

- **Overall Plan**: The overall plan serves as a basis for the whole project. It outlines key milestones, includes tasks[2], and provides a broad overview of the whole project timeline.
- **Finishing Plan** (in Dutch: Afbouwplanning): The finishing plan is derived from the overall plan and focuses on the finishing phase of the project. The finishing plan is equal to the overall plan but without the rough construction phase and with a bit more detail. However, it is not as detailed as an operational schedule.
- **Operational Schedule:** The operational schedule breaks down the tasks from the overall and finishing plans into more specific activities. It is more detailed and specifies what activity should be done at what moment by whom. So, the overall and finishing plans are input for the operational schedule.
- **Personnel Schedule:** The executor derives the personnel schedule from the operational schedule. In this, employees are assigned to specific activities each day.
- **Crane Schedule:** For large projects, the executor derives the crane schedule from the operational schedule. In this, the crane is assigned to specific activities. In most cases, there is one crane available for a project.

Two of the three mentioned roles have a major influence on planning and scheduling. The project leader creates the tactical and finishing plans and the executor creates and updates operational schedules based on these plans. When developing plans and schedules, Hegeman uses software called Software for Planning (SfP). This software is simple and easy to learn so everyone can work with it. However, this software does not allow for optimisation and Hegeman only uses it for manually creating and visualising plans and schedules.

Unfortunately, there is no data available regarding the fulfilment of projects. Only plans and schedules created in the past are available. Hegeman is willing to record the data regarding the fulfilment of projects but these are plans for in the future.

2.3 Current way of creating the Overall Plan

As mentioned in Section 2.2, the project leader creates the overall plan before the project starts. In principle, the overall plan is made once and remains unchanged throughout the project[3]. An overall plan includes all work that has to be done during the entire project. Projects vary a lot in length so the time horizon of the overall plan varies as well. The finishing phase of the overall plan is often not detailed yet because most subcontractors at this stage are unknown and much is uncertain about the finishing phase.

The project leader derives the finishing plan (in Dutch: afbouwplanning) from the overall plan. The finishing plan is an overall plan only including the finishing phase of the construction. This plan can be developed when all the subcontractors have been contracted. The finishing plan is tactical but looks a lot

[2] To distinguish between the components of an overall plan and an operational schedule, the term "tasks" is used for the overall plan, while "activities" is used for the operational schedule.

[3] The overall plan should only be revised in case it would not be representative anymore. For example, when the whole design of the building changes during the execution of the project.

like an operational schedule. The finishing plan is usually made shortly before the finishing phase begins. In general, it is made by the project leader but it could also be that the executor helps in this.

The overall plan includes milestones, which can be seen as (intermediate) deadlines during the entire project. The milestones include for example the start of the project, the moment rough construction has been finished, the moment the floor has been poured, the moment the building is wind and water-tight, and the delivery date. These milestones can also be seen as important things that have to be finished to start a new set of tasks involving new parties. For example, when the building is wind and water-tight, tasks that need dry space can start.

Furthermore, the overall plan includes all tasks that should be performed but those are not detailed yet. When a task is in the overall plan, it could be that this can be subdivided into multiple activities. These activities are only part of the operational schedule made by the executor. The project leader creates the overall plan in SfP.

In the overall plan, the project leader uses two different time scales. These time scales are shown in Figure 2.1. On top, there are the real weeks including all actual working days. Because of the weather, in the construction sector, not all days are workable days. So, when planning for the long term, it would not be realistic to assume all working days are workable days[4]. Therefore, the project leader uses historical data about how many workable days a month has on average. Based on that, a new time scale has been derived and is shown below the actual time scale.

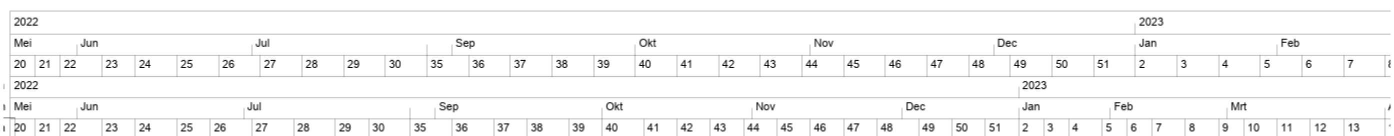


Figure 2.1 – Time Scale – Overall Plan

It never happens that the project goes exactly as the overall plan prescribes. However, the overall plan should be used as a guide during the whole project. The progress of the project can be measured by a progress line in the overall plan (in Dutch: standlijn). By this progress line, it is immediately visible whether the project is still on schedule. If this is not the case, the project leader can act timely. When drawing the progress line, the previous progress lines are often not removed. This is because it then shows the progression compared to the previous measuring moment. Drawing a progress line happens approximately every four weeks. Figure 2.2 shows a progress line in an overall plan drawn in week 7 of 2023 with the previous progress lines still visualised to show the progression[5]. The progress line is drawn from the top of the plan straight to the bottom of the plan if the project progresses as it should. However, if some tasks are delayed, the line deviates to the left.

[4] Some activities can be executed on nonworkable days but this only can be the case for activities that need to be executed inside a building. However, in that case, these days are still assumed to be nonworkable days since productivity is not as high as desired.

[5] Since the entire planning is too large to show, only a small piece is shown here.

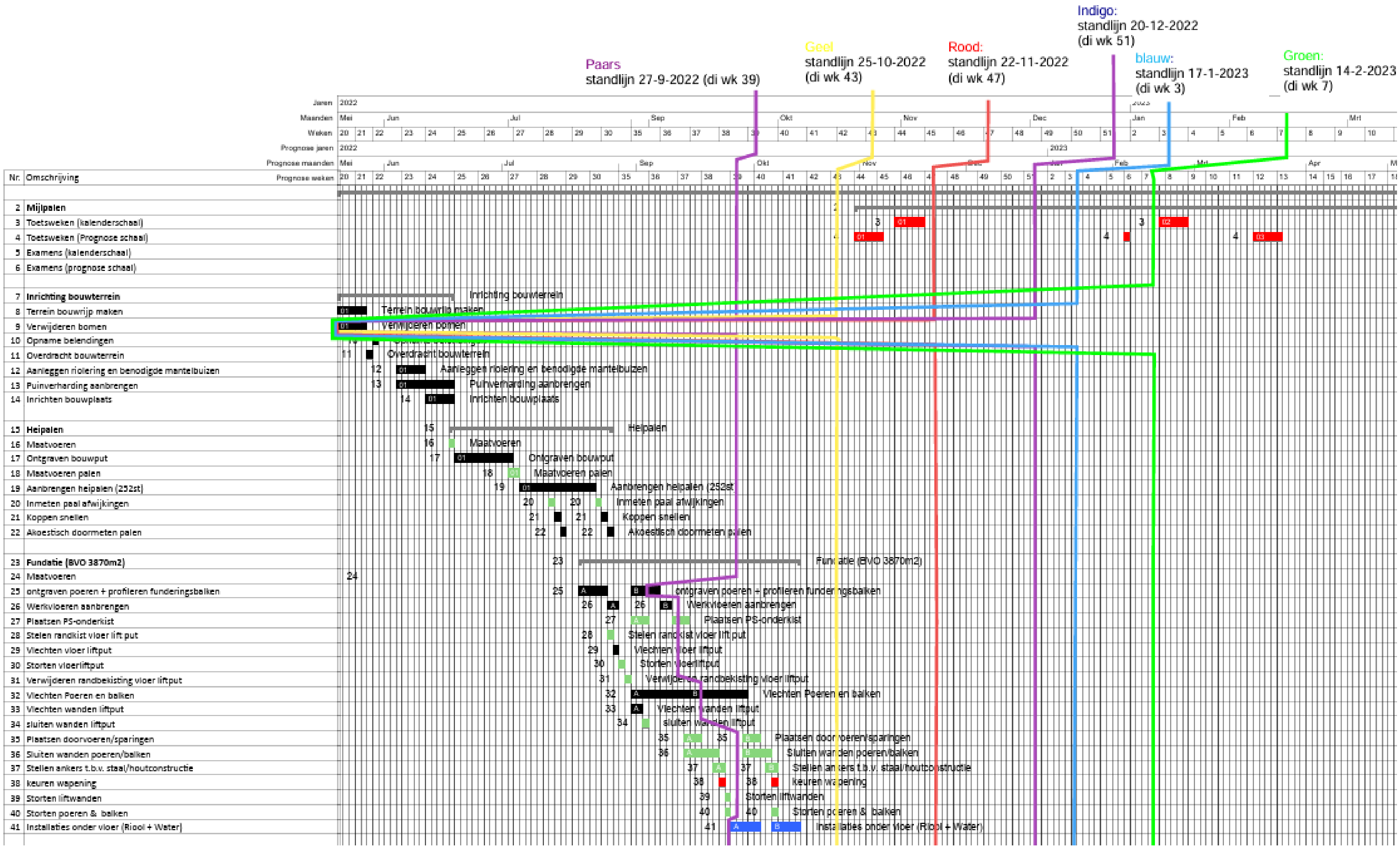


Figure 2.2 - Progress Line

In the overall plan, several tasks can be executed step-by-step at different moments. For example, floor by floor or from the left to the right of the building. This allows activities to be split up in the plan and gives planning opportunities. An example of a split-up activity is shown in Figure 2.3. In Figure 2.3, the scaffolding is built up floor by floor.

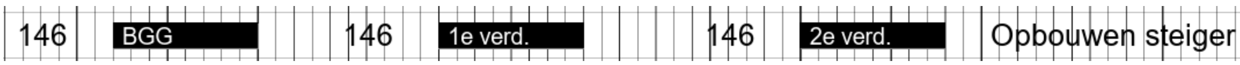


Figure 2.3 - Example of task split-up - Overall Plan

2.4 Current way of creating an Operational Schedule

The overall plan made by the project leader only includes the tasks to be done without all the underlying activities. So, only using an overall plan is not enough to start working. The operational schedule created by the executor is much more detailed and includes all underlying activities. The executor adjusts, updates and improves an operational schedule approximately every three to four weeks such that he exactly knows what needs to be done and who is going to do it. In the operational schedule, the executor schedules at least six weeks ahead but most executors schedule more weeks ahead. Just like the project leader, the executor uses SfP as well to create the operational schedule. The time scale used in the operational schedule includes all working days, so the operational schedule does not take into account unworkable weather. Figure 2.4 shows an example of an operational schedule made during the finishing phase for a recent project. The operational schedule could also be made from milestone to milestone. An efficient operational schedule should increase the chance of milestones being finished on time which should increase the chance the project is finished before its delivery date.

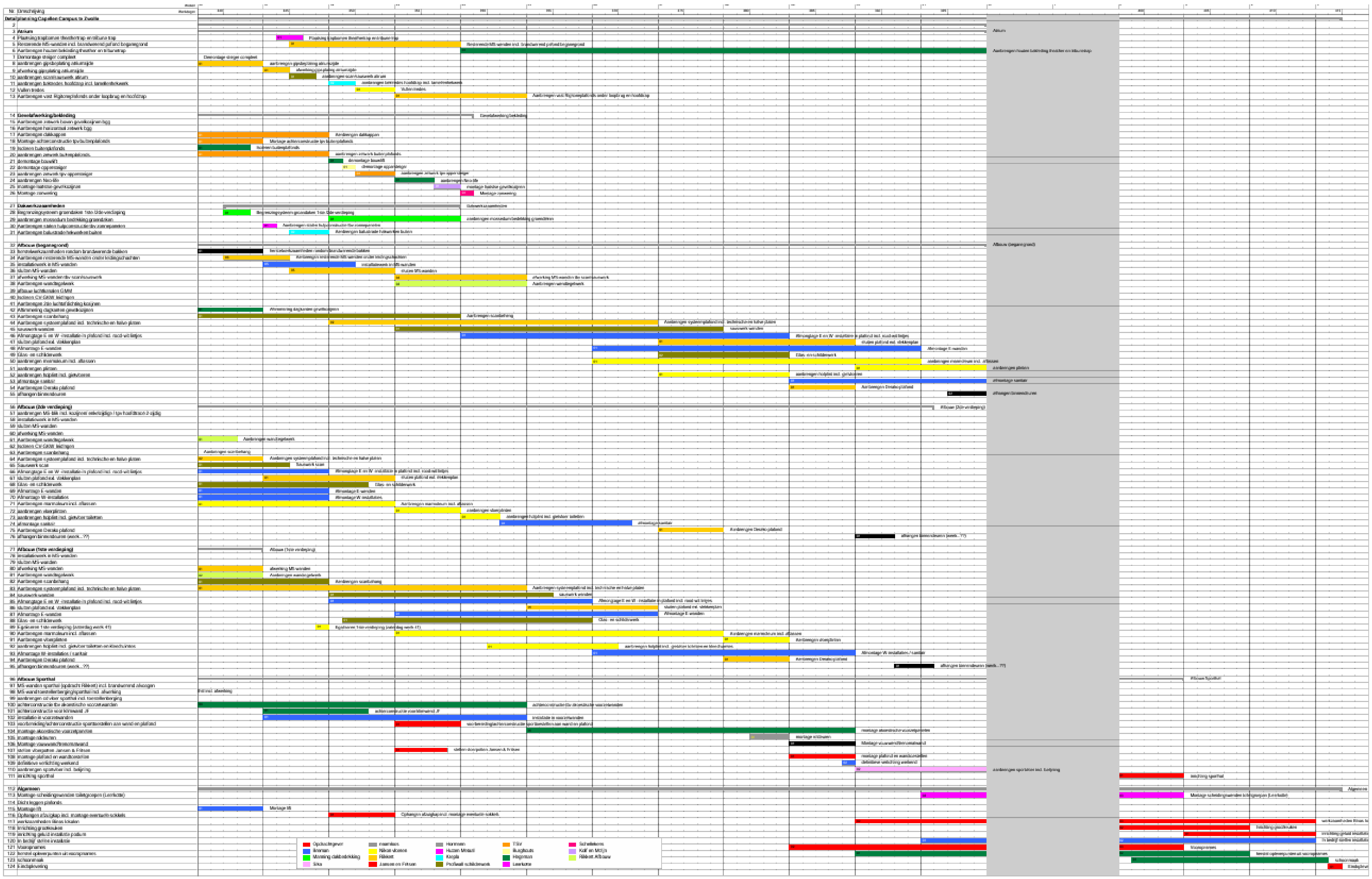


Figure 2.4 - Example of an Operational Schedule during the Finishing phase made in SfP

When creating the operational schedule, the executor organises the activities by location where they should be executed. All activities that are part of the operational schedule are derived from the specifications (in Dutch: bestek). The specifications include all requirements of the building down to the smallest detail.

A large fraction of the activities need to be executed by subcontractors. Only a fraction of the people at the construction site are employees of Hegeman. During the finishing phase, more subcontractors work at the construction site than during the rough construction phase because of the high necessity to have different specialities. This requires good coordination from the executor. In the rough construction phase, mainly employees from Hegeman do the work.

For visibility reasons, each subcontractor gets its colour in the schedule and a legend at the bottom shows which subcontractor belongs to which colour. When the executor has created the schedule, he sends this schedule to all subcontractors who need to fulfil their tasks in the upcoming weeks and asks for their reaction. Most of the time, the subcontractors do not have a problem with the schedule. However, it could be the case that the subcontractor thinks the amount of time he gets is not realistic or the timing is not possible. In these cases, the executor should change his schedule. In practice, subcontractors tend to pay little attention to the schedule created by the executor. This causes the necessity for the executor to call the subcontractors one or two days in advance to ensure that they are coming according to the schedule. Moreover, the subcontractors often do not see the problem of their work not being finished on time. When the subcontractor is not finished on time, the executor often has to find this out by himself. This is a big problem in the daily work of the executor and can cause a lot of scheduling problems.

The executor takes a look at the schedule daily. Every day in the morning, he checks the schedule to know what people should be working at the construction site on that particular day. Especially in the rough construction phase, it is important to follow the schedule since a delay from one activity can lead to large delays. Every morning, the executor also checks what is going to be delivered on that day. Since there is no large storage area on most construction sites, supply happens mainly on a Just-In-Time (JIT) basis. Delivery dates are based on the purchasing planning that is created before the project starts. The executors do the final call for materials manually via email but the date is based on the purchasing planning. The executor interviewed for this research would like to see the final call date in the operational schedule since forgetting to do the final call is a frequently occurring cause for delaying activities.

Before scheduling the activities, the executor has to determine the duration of these activities. For all activities, there is an established norm regarding the duration based on past projects. After Hegeman completes a project, they do a recalculation to gather data about the duration of each activity. This historical data can then be used to estimate the duration of activities for other projects. These durations are assumed to be deterministic without buffer or variation. In exceptional cases, the executor can choose to schedule one buffer day. For example, this can be done for floor pouring since this is one of the most important activities during the whole project.

When the operational schedule has been created, the executor derives the personnel schedule from it. In this, he assigns specific activities to each employee of Hegeman who works at the construction site on a particular day. Once per week, the executor has a meeting with the manager who creates the personnel planning and he then lets him know how many people he needs next week. The aim here is to work with an approximately equally large group every week. The executor also derives a crane schedule from the operational schedule. In this crane schedule, all activities that require crane movements are scheduled such that the crane capacity is not exceeded. The crane schedule is made with the same frequency as the personnel schedule. Figure 2.5 shows the relationships between the plans and schedules made, the frequency of these plans and schedules, and the person who makes them.

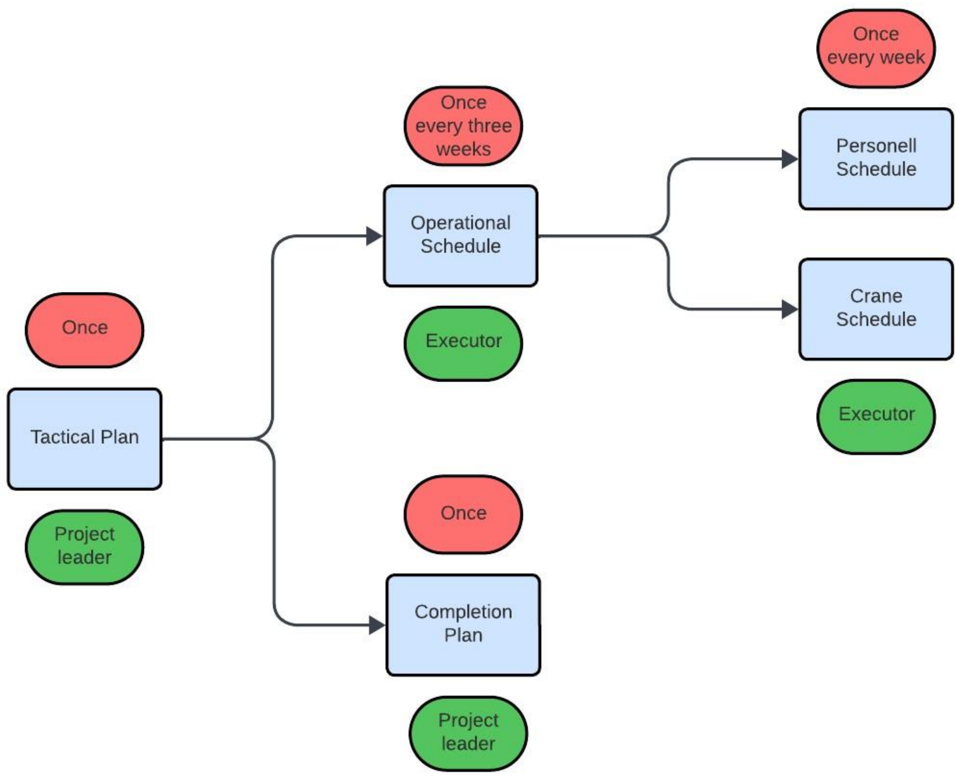


Figure 2.5 - Derivation and Frequencies of Plans and Schedules

2.5 Operational Challenges and Disruptions

Despite the high effort invested by project leaders and executors in creating reliable plans and schedules, they often encounter disruptions that prevent plans or schedules from being executed as intended. This section discusses these disruptions and assesses their impact. Moreover, this section discusses some other operational challenges executors can encounter. The following disruptions and operational challenges are the most likely ones to occur during the execution of plans and schedules:

- **Resource Availability**: When creating an operational schedule, the executor takes resource availability into account. However, it could be the case that for some reason certain resources are unavailable at moments they are needed. For example, a crane could be broken or an employee could be sick. This can cause the operational schedule should be revised.
- **Weather conditions**: During the development of the schedule, it is hard to predict what the weather conditions are on each day in the schedule horizon. Bad weather conditions can cause the necessity to revise the operational schedule and can cause delays. For example, wind can cause a crane cannot be used, or snow can cause no outdoor activities can be executed.
- **Delivery issues**: Another important challenge for executors to deal with is the delivery of materials. When the final call for delivery is not given on time, this can cause a delay in activities

eventually causing a delayed delivery date. Moreover, some materials can have lead times of up to a year. Delivery issues can cause activities not to be executed at the scheduled date causing possible refinement of the operational schedule.

- **Communication and Coordination problems**: Effective communication and coordination with suppliers and stakeholders is important. Unreliable suppliers and poor information sharing can result in delayed materials and missed deadlines.
- **Human mistakes**: The utility construction sector highly depends on human actions. Human mistakes can cause the plans and schedules not to be executed as expected causing the necessity for rescheduling. Human mistakes can be e.g., mistakes during an activity causing rework, contract and quotation do not match, or misperception about who needs to do what.
- **Regulatory issues:** Utility construction projects have to deal with regulatory issues. For example, tree removal may be restricted to certain seasons. Another example is that projects have to meet CO_2 standards which can cause large delays when not met.
- **Unexpected site conditions:** There are numerous examples of unexpected site conditions causing disrupted schedules. For example, a traffic accident might block access for delivery trucks. Another example is finding archaeological findings causing the project to be delayed.
- **Other disruptions and unforeseen events:** The disruptions and challenges discussed are just the most common ones. During a construction project, a lot of disruptions and unforeseen events can happen.

Executors have varying experience and knowledge and deal with disruptions differently. Some executors delay the entire schedule, others find creative solutions. Therefore, the impact of a disruption on the schedule depends on the experience and knowledge of the executor.

2.6 Stakeholders

Multiple stakeholders have an interest in the planning and scheduling process and, therefore, have an interest in this research. The main stakeholder is Hegeman. Hegeman is interested in this research because it aims to optimise their business process. If this research becomes a success, it likely increases the chance that delivery dates will be met and will save them money.

The second group of stakeholders are the executors and project leaders working at Hegeman. If this research has the expected outcome, implementation of the tool would make their work easier and the plans and schedules would become more efficient.

The third group of stakeholders are the clients of Hegeman. In principle, clients like to see that their building is finished on time. By creating more efficient plans and schedules, the buildings are more likely to be finished at the initiated delivery date.

The fourth stakeholder is the ECOLOGIC project and the people and companies involved in it. ECOLOGIC profits from this research by the methods and tools that optimise plans and schedules. When building the digital twin, research can be done on how to integrate this tool into the digital twin.

The fifth stakeholder is CAPE Groep. CAPE Groep profits from this research by having a student who is doing research in the utility construction sector. This increases their knowledge and experience in this sector and they can use this for other projects. Moreover, it increases their relationships with both Hegeman and the partners of ECOLOGIC and it gives them opportunities to further develop this tool in low-code software.

2.7 Conclusion

This chapter outlined the current situation of the planning and scheduling process in construction projects. In the planning and scheduling process, the project leader creates the overall plan beforehand and the executor creates every three to four weeks an operational schedule. The project leaders and executors make their plans and schedules in a software program called Software for Planning (SfP). The high amount of subcontractors involved in the project makes scheduling complex and calls the need for adequate coordination between all parties. The group of stakeholders consists of Hegeman, the executors and project leaders, the clients of Hegeman, the ECOLOGIC project with the companies and people involved, and CAPE Groep. To conclude, this chapter has provided insights into the current way of planning and scheduling at Hegeman to form the basis for this research.

3 Literature Review

This chapter lays the foundation for the development of the scheduling methods for this research. Section 3.1 introduces the concepts of planning and scheduling applied to the construction sector. Section 3.2 discusses a hierarchical planning framework and links the framework to this research. Section 3.3 addresses challenges in construction project planning and scheduling. Section 3.4 discusses the Resource-Constrained Project Scheduling Problem (RCPSP). Section 3.5 explores various planning and scheduling methods and techniques including the Critical Path Method (CPM) and the Critical Chain Method (CCM). Moreover, this section compares serial and parallel scheduling approaches. Section 3.5 ends by comparing related scheduling problems discussed in relevant academic literature. Lastly, Section 3.6 introduces the simulated annealing heuristic as it is an effective heuristic for solving complex optimisation problems, fitting this research.

3.1 Introduction

First, this section introduces the concepts of planning and scheduling. Next, this section identifies two different planning levels. This section ends with an introduction about different methods, tools, and techniques in planning and scheduling to build upon in Section 3.5.

According to Saad et al. (2015) & Majumder et al. (2022), the most substantial part of the construction process is the planning process by which the construction plan is created. Planning allows an easy understanding of how the project team should work, increases the ability of risk identification, and will help in achieving the project objectives (Aghimien et al., 2018). Because of the high complexity of the projects, the construction plan can be used as a guide during the execution of the project. The construction sector is highly fragmented because of the involvement of subcontractors leading to inefficient operating work processes (Viklund Tallgren et al., 2020). For these reasons, the process of creating a construction plan requires time from the project leader (Majumder et al., 2022).

Planning and scheduling are complex since there are different stakeholders involved who have their own interests. It is crucial to communicate the plans and schedules with subcontractors and employees to work efficiently (Majumder et al., 2022; Viklund Tallgren et al., 2020). Collaborative planning and scheduling aim to involve and integrate the different stakeholders to get more reliable plans and schedules (Al Nasseri & Aulin, 2016; Elsayegh & El-Adaway, 2021).

In the context of this research, two different levels of planning and scheduling can be distinguished. First, a project leader creates a tactical construction plan for the whole project. Based on that, operational schedules can be created and maintained continuously. As discussed in Section 2.1, operational schedules are more detailed than the tactical construction plans and include the identification of the necessary services and resources (Majumder et al., 2022).

There exists a lot of advanced planning and scheduling methods and tools. However, because of the limited knowledge about the existence, applicability and efficiency of these methods, executors and project leaders tend to choose traditional methods such as the Critical Path Method (Sheikhkhoshkar et al., 2023; Winch & Kelsey, 2005). These traditional methods do not encompass all constraints. The quality of plans and schedules depends a lot on the experience and insights of the executor, project leader, or other person who creates the plans or schedules (Saad et al., 2015). According to Zwikael (2009), construction project managers do not always spend enough effort in the planning and scheduling process. Planning decisions made in an early stage of the project can have a large impact on the delivery date and state of the final product (Aghimien et al., 2018). Project planning has a high impact on the outcome of the project and is therefore crucial to spend effort in (Viklund Tallgren et al., 2020; Zwikael, 2009). Creating a good project plan has several advantages. First of all, it reduces uncertainty and improves the efficiency of the operation. Moreover, it gives a better understanding of the project objectives and allows for monitoring and controlling the work that needs to be done (Zwikael, 2009).

According to Saad et al. (2015), planning techniques can be classified into two types, which include Location-based planning techniques and Activity-based planning techniques. Location-based planning techniques focus on repetitive activities at different locations in the building for which schedules are graphically presented as a series of production lines in which each line represents a repetitive activity. Activity-based planning techniques are more like a network of activities with relationships. There are dependency constraints and the Critical Path Method (CPM) is a common approach for creating plans and schedules (Saad et al., 2015; Seppänen & Kenley, 2005). According to Tang et al. (2013), CPM is the most used project scheduling method. CPM will be discussed in Section 3.5.1.

There are numerous planning and scheduling methods and tools aiming to deal with challenges in planning and scheduling and increase efficiency. Sheikhkhoshkar et al. (2023) identify examples of some methods and they highlight the existence of CPM. Moreover, Sheikhkhoshkar et al. (2023) discuss the Critical Chain Method (CCM), which is related to CPM. CCM will be discussed in Section 3.5.2. Other methods Sheikhkhoshkar et al. (2023) discuss are the Linear Scheduling Model (LSM), Line of Balance (LOB), Location-based management system (LBMS), Fourth Dimension of Building Information Modeling (4DBIM), Last Planner System (LPS), Takt Time Planning (TTP), Triconstraint Method (TCM), and Simulation-based Methods (SM). Yarramsetti & Kousalya (2015) mention methods like Ant Colony Optimisation, Particle Swarm Optimisation (PSO), and Genetic Algorithm (GA). Majumder et al. (2022) discuss Q-scheduling, Program Evaluation and Review Technique (PERT), Line of Balance (LOB), Resource-Oriented Scheduling, and Gantt charts. Next to CPM and CCM, Viklund Tallgren et al. (2020) mention critical-space-analysis (CSA) and Last Planner System (LPS). Most of these names imply the objective of the specific method.

3.2 Hierarchical Planning Framework

Various researchers have developed hierarchical planning frameworks providing a structured approach to project planning. De Boer (1998) developed a framework including four planning levels:

1. Strategic resource planning (Strategic)
2. Rough-cut capacity planning (Tactical)
3. Resource-constrained project scheduling (Tactical/Operational)
4. Detailed scheduling (Operational)

The lower the level, the less uncertainty is involved. Linking these levels to this research, the fourth level corresponds to the operational schedule. De Boer (1998) introduces an additional level between the tactical and the operational levels, making this framework particularly relevant to our study.

Hans et al. (2007) use the same levels but call both the third and fourth levels operational. In this research, the third level is considered as being tactical. The overall plan corresponds to the third level in this framework and has a small overlap with the second level. Creating the overall plan mainly has to do with scheduling the task but capacity is interrelated with this. While the focus of this research is on the fourth level, the method that will be developed for creating operational schedules can potentially also be applied to the third level.

3.3 Challenges in Construction Project Planning and Scheduling

Because of the high complexity, there are a lot of challenges involved in construction project planning and scheduling. This brings us to the first challenge, which is about the high uncertainties in construction projects (Elsayegh & El-Adaway, 2021; Winch & Kelsey, 2005). These uncertainties consist of external events, which include labour disturbances, unreliable weather, unexpected rework, material delivery delay and other unforeseen events (N. Kim et al., 2021; Sheikhkhoshkar et al., 2023; Tang et al., 2013). When disruptions occur, there is a mismatch between the desired and the actual scheduling causing replanning or rescheduling to be necessary but it is a challenge to do this properly (N. Kim et al., 2021).

The next challenge is to take subcontractors' schedule information into account. Currently, not all companies do this properly. Taking the schedules of subcontractors into account would probably lead to better project schedule coordination (Choi, 2012). Different companies tend to work individually without coordination between the different plans and schedules (Al-Emad et al., 2017; Nawaz et al., 2021; Sheikhkhoshkar et al., 2023). Moreover, communication in general is an important challenge to consider. All parties involved should communicate with the right persons to work effectively (Al Nasseri & Aulin, 2016).

Another challenge is the fragmented character of the sector, which requires improved coordination between different parties. Communication mainly happens via paper and is often inefficient or inadequate, which can cause delays and costs because of the ineffectiveness of this type of communication (Choi, 2012; Elsayegh & El-Adaway, 2021; Sheikhkhoshkar et al., 2023).

The next challenge is to deal with the dependency on personal experience (Choi, 2012). Since planning and scheduling in the construction sector heavily rely on logical thinking and experience, each planner or scheduler works differently causing some planners to work less effectively and some plans and schedules to be less reliable (Nawaz et al., 2021; Winch & Kelsey, 2005). Because of simplicity and because it works, companies and people are not always willing to change their way of creating plans and schedules causing them to miss the opportunity to work more effectively and efficiently.

According to Viklund Tallgren et al. (2020), an important challenge for pre-construction planning is the lack of available time for the person who makes this plan causing the quality of the plan to be moderate. This could also be a problem in making the operational schedule. It is important to spend enough time and effort in making the plans and schedules to increase efficiency.

The discussion concludes with the challenge of a shortage of people (Al-Emad et al., 2017). When there is a shortage of employees or other people in the construction site, some work cannot be finished before it should be. This includes illness as well. When employees or other people are unexpectedly ill for a couple of days, this could delay the project.

3.4 Resource-Constrained Project Scheduling Problem (RCPSP)

The Resource-Constrained Project Scheduling Problem (RCPSP) is a well-known and extensively studied problem in the literature. According to Abbasi et al. (2006), it is even the most important problem in project scheduling. The basic RCPSP is a relatively simple problem aiming to minimise project duration. The problem aims to schedule activities over time while considering precedence relations and resource capacity limitations (Naber & Kolisch, 2014; Tritschler et al., 2017). Some extensions of the RCPSP in the literature aim to optimise other objectives but most versions aim to minimise project duration. The RCPSP is an NP-hard problem and is aimed to be solved through optimisation methods like heuristics (Hartmann & Briskorn, 2022).

Different approaches to solving the RCPSP have been researched in the literature. Abbasi et al. (2006) develop a multi-objective approach in which the project duration is aimed to be minimised while the robustness is aimed to be maximised to make the schedule more reliable.

Hartmann & Briskorn (2022) discuss all variants and extensions being considered in the literature such that it can function as an overview of the RCPSP. The list with extension directions is too large to discuss entirely but they can be categorised into five different categories. These categories include generalisations of the activity concept, alternative precedence constraints and network characteristics, extensions of the research concept, different objectives, and multiple projects.

Most variants of RCPSP assume deterministic types of RCPSP but Abdolshah (2014) highlights the distinction between deterministic and non-deterministic types of RCPSP. Moreover, Abdolshah (2014) discuss the distinction between exact and heuristic methods. Because RCPSPs are NP-hard problems, exact solutions can only be found for small problem instances or simplified versions of the RCPSP. For realistic problems, heuristics are necessary for good solutions. Next, Abdolshah (2014) discusses different heuristics and meta-heuristics to solve the RCPSP.

3.5 Planning and Scheduling Methods and Techniques

This section dives into specific methods to develop an overall plan or operational schedule. Section 3.5.1 discusses the well-known Critical Path Method (CPM). After that, Section 3.5.2 dives into the related Critical Chain Method (CCM). Next, Section 3.5.3 explains the difference between serial and parallel approaches. Lastly, Section 3.5.4 compares the scheduling method from this research with methods from seven different articles based on fourteen different criteria.

3.5.1 Critical Path Method (CPM)

The Critical Path Method (CPM) is the most used planning and scheduling method in the construction industry (Saad et al., 2015; Tang et al., 2013). As the name implies, CPM aims to create a path through all critical activities. Critical activities are activities that may not be delayed because a delay in one of these activities leads to a delay in the completion of the project (Atin & Lubis, 2019; Bishnoi, 2018; De Boer, 1998).

The first step in CPM is to list all required activities to finish the project including their durations and predecessors. If there is an earliest starting time and a latest finishing time for some activities, these should be written down as well (Bishnoi, 2018). The next step is to draw a network diagram with the activities being the boxes and the arrows indicating the dependencies. In these boxes, information about the activity can be stored. The following step is to identify the shortest project duration using a forward pass (Cynthia, 2020). The next step is to establish the total slack and the critical activities using a backward pass (Lu et al., 2008). After that, the schedule generated based on this critical path can then be represented as a Gantt chart or bar chart and should be updated regularly. The critical path consists of activities that directly impact the project’s completion date if they are delayed (Bishnoi, 2018). The flexibility of each activity can be measured by the slack, which is the time that the activity may be delayed. The project flexibility can be calculated by summing up the slack of all activities (Kreis et al., 2019).

According to Winch & Kelsey (2005), there are two key issues when using CPM. The first issue is that it does not consider variability in durations. CPM is completely deterministic and only uses a predetermined or average duration for each task making it difficult to obtain a reasonable schedule (Kim et al., 2021). In environments with limited uncertainty, CPM is a very powerful scheduling tool. The second key issue is that CPM does not consider the resourcing of activities (Majumder et al., 2022; Winch & Kelsey, 2005). Resourcing comprises among other things assigning employees and materials to tasks and activities.

When applying CPM to a case where resources are required, it ignores the resources and assumes given times which is not realistic in most cases (Abeyasinghe et al., 2001). According to Sears et al. (2008), resources in the construction sector include manpower, materials, construction equipment, and subcontractors. Lu & Li (2003) aims to add resource constraints to CPM and proposes a new method called Resource-Activity Critical-Path Method (RACPM). By incorporating resource constraints, more realistic schedules can be created but at the cost of having a more complex model. Abeyasinghe et al. (2001) also incorporate resource constraints in the model and have the objective of developing a heuristic from which the output comes as close as possible to the CPM output regarding the shortest duration. Kim & de la Garza (2003) propose another method that incorporates resource constraints, which has been called the Resource-Constrained Critical Path Method (RCPM). This method is a step-by-step approach that combines both CPM and Resource-Constrained Scheduling (RCS). Step 1 is to perform standard CPM. Step 2 is to perform serial RCS. Step 3 uses a backward pass to calculate the latest start and finish times considering both the resource constraints and precedence relations. Step 4 creates a final schedule of RCPM and step 5 identifies alternative schedules to give some flexibility in scheduling options (Kim & de la Garza, 2005).

Precedence relations could also be fractional, indicating that a portion of an activity needs to be finished before the next activity can start. For example, 50% of the floor needs to be poured to start the next activity. Most models and research do not consider fractional precedence relations. Bokor et al. (2011) discuss a method called the Precedence Diagramming Method (PDM), which takes into account complex dependencies between different activities. Multiple articles use relationships between the start and/or finish times of related activities called Generalised Precedence Relations (GPR). The literature identifies four different GPRs including Start-to-Start (SS), Finish-to-Finish (FF), Start-To-Finish (SF), and Finish-to-Start (FS) (Alfieri et al., 2011; Baydoun et al., 2016; Ökmen, 2013). For example, when the FS relation is four time units, there should be at least four time units between the preceding activity and the next activity.

3.5.2 Critical Chain Method (CCM)

When someone decides to use some buffer days in the schedule, CPM can still be used by extending the duration of the activities to be sure the activities are finished on time. However, this probably leads to high idle time in between activities, which is unnecessary since other activities can probably already start. An example of how this would look is shown in Figure 3.1. In Figure 3.1, the red boxes are the average durations of activities and the yellow boxes are the buffer days. This way of using buffers can lead to the emergence of Parkinson's law. Parkinson's law states that people use the time they get to complete a job (Kim et al., 2021). So, by creating a buffer, people tend to work more slowly because the deadline is farther in the future. People often need deadlines to finish early.

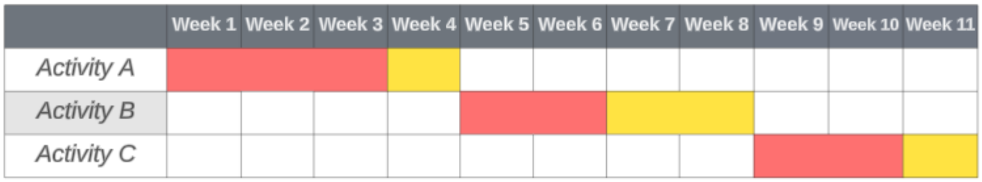


Figure 3.1 - Critical Path Method - Example with buffer

The Critical Chain Method (CCM) is a technique that uses buffer time but still allows activities to start when predecessors are finished. CCM prevents project overruns and properly handles uncertainties in schedules using buffers. Unlike traditional methods, CCM focuses on resource availability as the primary constraint ensuring that resources are optimally utilised throughout the project. CCM builds buffers where necessary since delay at that point can lead to problems in project completion. For that reason, CCM focuses on the activities that form the longest sequence of dependent tasks when considering resource constraints, which differs from the critical path in that it takes resource constraints into account (Ghaffari & Emsley, 2015). CCM places the buffer at the end of the chain so there is one buffer to catch all variability, lowering the total buffer and protecting overall project completion (Leach, 1999). This is also a way to deal with Parkinson's law since there is an initial tight deadline (Kim et al., 2021). An example of how a CCM schedule would look is shown in Figure 3.2.

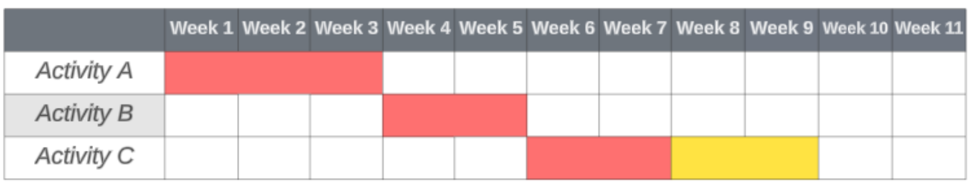


Figure 3.2 - Critical Chain Scheduling - Example

According to Kim et al. (2021), there is limited empirical evidence for CCM implying its ineffectiveness in the construction sector and previous research failed to show that it works. This highlights the limited suitability of CCM in the construction sector. Moreover, CCM is mainly a theoretical method with limited applications in reality.

3.5.3 Serial vs Parallel Approach

The literature distinguishes two different approaches for scheduling activities: the serial and the parallel approach (Abeyasinghe et al., 2001; Kim, 2020; Kim & de la Garza, 2005; Lu & Li, 2003).

In the serial approach, a priority rule determines the order of activities based on various criteria such as duration, urgency or slack time (Abeyasinghe et al., 2001). The activity with the highest priority is scheduled first, and resources become available as each activity is completed. The next activity with the highest priority is then scheduled, and this process continues until all activities have been scheduled (Kim, 2020).

The parallel approach determines which activities can start at the beginning of each time unit. A decision set is created for each time unit including activities that may start. If capacity restrictions cause not all activities to start simultaneously, the activities with the highest priority index may start (Kim, 2020). Activities that are not scheduled are considered again during the next time step. Then, the activities with the highest priority are scheduled. After that, the priority indices are updated for the next time unit.

3.5.4 Comparison of Scheduling Methods

Based on the explanation in Section 3.5.1, the CPM is the method to be the basis for the method to be developed in this research. The CPM is well-suited for this research for several reasons. First, it is a structured approach widely used in the construction industry meaning there is a lot of literature and case studies available. Additionally, CPM focuses on critical activities allowing for prioritising activities and calculating slack. Furthermore, as Hegeman does, CPM assumes deterministic durations of activities. Lastly, there are numerous extensions available, some of which will be analysed in this section.

This section analyses articles that use some variant of a Resource-Constrained Critical Path Method. Table 1 shows what topics these articles address compared to what this research addresses. This table provides insights into existing methods that can contribute to this research. These methods lay the foundation for the Scheduling Method Chapter 4 develops for this research.

	Woodworth & Shanahan (1988)	Bowers (1995)	Abeyasinghe et al. (2001)	Kim & de la Garza (2003)	Lu & Li (2003)	Pantouvakis & Manoliadis (2006)	Nkasu (1994)	This research
Identifies critical path	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Identifies critical sequence/resource-constrained CP	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
RCS Method	Parallel	Parallel	None	Serial	Serial	Serial	Parallel	Parallel
Heuristic	Min Slack	Min LS	Companion Activities	Latest Start Time + Shorest duration D + Total Float	Work Content	Min LST	Randomly select/Project completion time	Priority index
Multiple resources	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Capacity > 1 allowed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Flexible resource capacity	Not specified	Not specified	No	Yes	No	No	Not specified	Yes
Identify resource links	Yes, during backward pass	Yes, during forward and backward pass	Yes, during forward pass	Yes, during forward pass and backward pass	Yes, after forward pass	Yes, during forward pass	No	Yes, during backward pass
Fractional/ Generalised Precedence relationships	No	No	Yes	No	No	No	No	Yes
Multi-objective	No	No	No	No	Yes	Yes	Yes	Yes
Objective optimisation	Minimise makespan	Minimise makespan	Minimise makespan	Minimise makespan	Minimise makespan and utilisation	Minimise makespan and maximise smoothness (resource continuity)	Minimise makespan	Minimise makespan, maximise work continuity & resource utilisation
Handles uncertainty	No	No	No	No	No	No	No	Yes
Optimal Solution guaranteed	No	No	No	No	No	No	No	No
Resource variability	No	No	Yes	Yes	Yes	Yes	Yes	Yes

Table 3.1 - Comparison of Scheduling Methods from the Literature

The articles in Table 3.1 are compared based on whether they identify critical paths, and critical sequences, whether they use a parallel or serial approach, what heuristic they use, whether they can handle multiple resources, whether a capacity greater than 1 is allowed, whether the resource capacity is flexible, whether they use some type of fractional precedence relations, whether they are multi-objective, what objective(s) they minimise, whether they guarantee an optimal solution, and whether they take into account resource variability. Resource variability is defined as the fluctuations in resource allocation between different days. These articles are compared to this research to show what components overlap and what components do not. The term "critical sequence" has not been discussed before. The critical sequence is the same as the critical path in that it identifies what activities are critical. However, it considers not only precedence relations but also resource relations (Woodworth & Shanahan, 1988). For example, Activity A might be delayed by four days according to its precedence relations without impacting its successor. However, if the resources allocated to Activity A are required for another activity immediately after its completion, Activity A becomes critical. In this case, Activity A is not in the critical path but it is in the critical sequence.

Most methods are applicable in practice but Abeyasinghe et al. (2001) classify their method as a theoretical method. Together, all methods lay the foundation for this research. On the other hand, most methods have one objective which is minimising project duration. However, this research is multi-objective and next to minimising the project duration and resource variability, it aims to maximise utilisation. The scheduling problems in Lu & Li (2003) and Pantouvakis & Manoliadis (2006) are multi-objective and discuss adding resource utilisation and resource variability to the method.

3.6 Simulated Annealing

Simulated annealing (SA) is a well-known improvement heuristic in optimisation problems. Improvement heuristics aim to improve initial solutions by exploring or exploiting solutions spaces. In the literature, various improvement heuristics exist, all having their strength and weaknesses. Effective improvement heuristics aim to exploit promising neighbourhoods and thoroughly explore these promising neighbourhoods to find the best solutions (König & Beißert, 2009).

SA is an effective metaheuristic that balances both intensification (exploitation) and diversification (exploration). SA starts with an initial solution and iteratively applies small changes to this solution aiming to find better solutions. To assess whether a newly found solution is better than an earlier found solution, an objective function is necessary which can be either maximised or minimised. If the new solution is better, it becomes the new current and best solution. If a newfound solution is not better, it may still be accepted as the current solution based on a probability that decreases over time (Henderson et al., 2006; Van Laarhoven et al., 1992).

The probability by which a new solution is accepted is called the acceptance probability. In the beginning, the acceptance probability is high causing almost every newly found solution to be accepted as the current solution. This is to explore the whole solution space and find promising neighbourhoods. After a fixed number of iterations called the Markov-chain length, the temperature decreases by a factor-alpha, shifting the focus towards exploitation where mostly better solutions are accepted (Rader, 2010). In the end, SA focuses more on intensification since then it aims to find the best solution for a promising neighbourhood.

One of the main advantages of simulated annealing compared to other metaheuristics is the balance between intensification and diversification. Comparing this to a method like Local Search, simulated annealing is more likely to find the global optimum since Local Search only focuses on exploitation only accepting better solutions and ends up in a local optimum instead of the global optimum. Simulated annealing is especially useful for problems with large solution spaces (Henderson et al., 2006).

3.7 Conclusion

Chapter 3 provided insights into the literature on planning and scheduling, with a specific focus on construction project planning and scheduling. This chapter highlighted the crucial role of accurate planning and scheduling to guarantee smooth project execution and timely project completion.

This chapter started by introducing the concepts of planning and scheduling. Next, different planning levels have been identified.

The third section of this chapter focused on the challenges in construction project planning and scheduling. It highlighted the complexity of the field and discussed the most important challenges.

The fourth section discussed the well-known, extensively researched Resource-Constrained Project Scheduling Problem (RCPSP). This problem knows a lot of variants and extensions in the literature all focusing on different deviations from the basic model formulation.

The fifth section of this chapter discussed several methods relevant to this research. First, it discussed the most used method in construction scheduling, the Critical Path Method (CPM). Next, it discussed a method derived from CPM that aims to deal with variability, the Critical Chain Method (CCM). After that, the difference between serial and parallel approaches has been explained. This section ended by comparing the method from this research with methods from seven different articles. This comparison has been done based on fourteen different criteria.

The last section introduced the Simulated Annealing heuristic which fits well for problems with large solution spaces instances as in this research. In summary, Chapter 3 serves as a foundation for the next chapter, which develops the methods for this research.

4 Schedule Creation Method Development

This chapter develops a method to create operational schedules. Section 4.1 discusses the problem description to formally introduce the problem, its constraints, and its objective function. Section 4.2 discusses all the input information required to create the operational schedule. Section 4.3 dives into the assumptions and simplifications for the scheduling method. Section 4.4 develops the Multi-Objective Resource-Constrained Critical Path Method (MORC CPM) for creating operational schedules.

4.1 Problem Description

Hegeman faces challenges in creating efficient schedules due to the absence of a generic scheduling method. This chapter develops a method for the executors to use when creating their operational schedules to resolve the issue of frequent delays in project delivery dates. Currently, executors create inefficient operational schedules because they rely on common sense and individual experience, leading to inconsistencies and inefficiencies. The current way of creating operational schedules is inefficient because the schedules are made manually without standardisation making it difficult to maintain consistency and create reliable schedules.

When creating an operational schedule, several constraints should be considered:

1. The first constraint to consider is the limited resource availability. Activities not executed by subcontractors require employees, cranes, or other resources limiting all activities to be executed simultaneously. In general, subcontractor activities do not require any resources from Hegeman, although cranes can be shared. Additionally, one activity can require multiple resource types.
2. The second constraint is that some activities may only be executed during some parts of the year or may not start before a certain date because of subcontractor availability. For example, cutting down trees is only allowed between 15 September and 15 March. Therefore, the activity requires a release date and a due date in between which the activity should be executed.
3. The third constraint is the existence of (fractional) precedence relations. Construction projects involve activities that are highly dependent on each other, requiring an established order of execution. This means that activity can only start when its predecessors have progressed far enough. In some cases, the predecessors should be finished before an activity may start. However, there are situations where an activity can begin when a certain fraction of the preceding activity is finished. For example, a succeeding activity may start when 50% of its preceding activity is completed. In the literature, fractional precedence relations are known as Finish-Start relations in which the number of days between the finish date of the preceding activity and the start date of its successor (Abdolshah, 2014; Alfieri et al., 2011; Hartmann & Briskorn, 2022; Kreis et al., 2019; Lu et al., 2008; Naber & Kolisch, 2014; Ökmen, 2013). However, percentages are preferred in this research because they indicate what fraction of an activity should be completed before its successors can start. This provides a more practical approach to managing precedence relations.

The problem in this research is multi-objective. However, to optimise all parts together, it can be useful to merge the different objectives into one objective function. By assigning weights to the different objectives, the different objectives can be combined into one objective function making it single-objective. The objective function aims to minimise project duration while maximising employee utilisation and minimising variability in resource usage between different days. The objective function in this phase is:

$$\min z = w_1 * \left(\frac{PD}{PD_{P2}}\right) + w_2 * \left(\frac{\overline{U_{P2}}}{\overline{U}}\right) + w_3 * \left(\frac{\left(\frac{1}{n}\sum_{r=1}^{R}\sqrt{\sum_{t=1}^{PD}\left(\frac{U_{tr}-\overline{U}_r}{\overline{U}_r}\right)^2}\right)}{\max\left(\sqrt{\sum_{t=1}^{PD}\left(\frac{U_{tr}-\overline{U}_r}{\overline{U}_r}\right)^2}\right)}\right)$$

Where,

- R is the total number of resources
- w_1, w_2, w_3 are the weights for each term representing its relative importance
- PD is the project duration
- PD_{P2} is the project duration in the output schedule from Phase 2
- $\overline{U_{P2}}$ is the average utilisation of resources in the output schedule from Phase 2
- $\overline{U}$ is the average utilisation of resources in the current schedule
- U_{tr} is the utilisation of resources r at day t
- $\overline{U_r}$ is the average utilisation of resource r

In this objective function, the first term aims to minimise the project duration. It does this by multiplying the weight w_1 by the project duration of the current schedule divided by the project duration of Phase 2. The weight of w_1 determines how important the project duration is in the objective value. Dividing by the project duration of the schedule from Phase 2 normalises the first term.

The second term aims to maximise average utilisation. As for the other terms, the second term is multiplied by a weight w_2 to determine its impact on the objective value. Since the objective value is minimised, the utilisation of the current schedule is in the denominator. This guarantees that higher utilisation leads to a lower objective value. So, the higher the utilisation the lower and better the objective value.

The third term aims to balance the workload, so it aims to minimise the gap between the utilisation on a particular day and the average utilisation. First of all, this term should be multiplied by w_3. Secondly, the term calculates the root mean square of the normalised differences between the resource utilisation and the average resource utilisation across all resources for the whole project duration and normalises it by dividing it by the maximum of these values. This guarantees the value to be below 1. The root mean square is used for this because it ensures a balanced distribution between the resources.

The operational schedule should be made for at least the upcoming six weeks. This is a repeating process for the executor, who makes a new operational schedule approximately every two weeks. To keep the start and finish dates of the old schedule if those activities cannot be rescheduled, the executor can use release dates and due dates. If all activities can be rescheduled, the executor can apply the method without fixing the start and finish times of activities.

4.2 Input information for Operational Scheduling Method

To apply the scheduling method this chapter develops, several inputs from the user are required. This section discusses all the inputs necessary to create the operational schedule. The main input for the method is the activities to be scheduled. These activities all include different properties being the input information for the method:

1. Activity duration
2. Predecessors
3. Predecessor percentages
4. Release date
5. Due date
6. Subcontractor (Only non-Hegeman activities)
7. Resource requirements
8. Uncertainty factor

Each activity has its input parameters describing the activity. The first input parameter is the activity duration. The activity duration is assumed to be deterministic since the duration can be estimated very accurately. These estimations are based on subsequent calculations of previous projects and industry standards. For example, Hegeman knows that placing one pile would on average require twenty minutes, then it can estimate how long it takes to place 50 piles by just multiplying twenty minutes by 50.

The second input parameter is the predecessors. The predecessors are the activities that should be finished before starting the activity. The second input parameter is linked to the third input parameter which is the predecessor percentage. The predecessor percentage indicates to what extent a predecessor should be finished to start the activity. For example, it could be that 30 of the 50 piles should be placed to start the next activity. In that case, the predecessor percentage is 60%.

The fourth input parameter is the release date. The fifth input parameter is the due date. These input parameters are optional to be assigned to an activity as most activities would not require this. For example, some activities may only be executed during some parts of the year or a subcontractor indicates that it has only time available in between certain dates. There could be more reasons why these dates are necessary. Allowing the user to fill in this information can make the schedule more realistic.

The sixth input parameter is the subcontractor that should execute the activity. Whether an activity should be executed by Hegeman or by a subcontractor influences the resource requirements. If an activity will be performed by a subcontractor, it requires no employees from Hegeman. Stating the name of the subcontractor should be done for visualisation purposes. By this, the executor can easily see when and for what the subcontractor should come.

The seventh input parameter is the resource requirements. The most important resource to consider while scheduling is the employees (Sears et al., 2008). The executor should indicate how many employees are required to perform an activity. Next to employees, the executor can fill in other resources like small cranes, big cranes, pumps, etc. Moreover, the executor should fill in the capacity of all resources. As mentioned in the previous paragraph, subcontractors require no employees from Hegeman. In principle, the subcontractors do not require any resources from Hegeman, however, in some cases, resources like cranes can be shared with a subcontractor.

The eighth input parameter is the uncertainty factor. As mentioned earlier in this section, the activity durations are assumed to be deterministic. However, in reality, some durations are more predictable than others. For example, suppliers can be unreliable, or there could be geopolitical, environmental, or regulatory factors affecting the completion time of activities. There could be other reasons as well for differing reliability.

To account for these uncertainties, each activity gets an uncertainty factor ranging from 1 to 5. This factor allows the scheduling method to allocate more slack time to activities with higher uncertainty. The uncertainty factor is defined as follows:

- Uncertainty factor = 1: This indicates the finishing date is certain
- Uncertainty factor = 5: This indicates the finishing date is uncertain

By incorporating uncertainty factors, the method can manage delays and improve the reliability of the project.

4.3 Assumptions & Simplifications

This section discusses the assumptions and simplifications for the Multi-Objective Resource-Constrained Critical Path Method (MORC CPM). The assumptions and simplifications are:

- Once an activity has started, this activity is not allowed to be interrupted and finished later. This is a realistic assumption. At least, it is always scheduled in this way but in reality, activities can be interrupted, for example, because of illness or regulatory restrictions.
- The durations are assumed to be deterministic. Hegeman always schedules assuming deterministic durations because, in general, estimated durations seem to be precise.
- Subcontractor activities do not require any additional work for Hegeman. This is a realistic assumption since apart from monitoring and reviewing the work, it does in general require no work from Hegeman.
- Subcontractors are always available to be scheduled. This assumption is realistic since only in unique cases, the subcontractor does not accept the requested dates.
- When the same activity should be performed on different floors or at different locations at different moments, performing an activity on a specific floor or location is considered a separate activity. Otherwise, this conflicts with the assumption that splits are not allowed. Moreover, this makes the method, input screen, and calculations easier without having huge disadvantages.
- All input information provided by the executor is reliable and accurate.
- All workers have the same skill level and can perform any task. If it is necessary to distinguish worker types, another resource type can be created. For example, instead of being an employee, an employee can become a crane operator resource type.
- Set up time in between activities is not considered.

4.4 Multi-Objective Resource-Constrained Critical Path Method

This section develops the Multi-Objective Resource-Constrained Critical Path Method (MORC CPM). This method aims to create a practically applicable schedule while minimising project duration and resource variability and maximising resource utilisation. Section 4.3.1 identifies the five phases of the MORC CPM and explains them in short. Section 4.3.2 dives into Phase 1 which corresponds to the general CPM. Section 4.3.3 discusses Phase 2 which develops a constructive heuristic in which a forward pass includes resource constraints in the schedule from Phase 1. Section 4.3.4 develops a simulated annealing heuristic to improve the current schedule in Phase 3. Section 4.3.5 dives into Phase 4 in which a backward pass identifies resource links and the critical sequence. Finally, Section 4.3.6 develops Phase 5 in which the part of the method which intends to prescribe to the executor what activities to already start to increase employee utilisation to 100%.

4.4.1 Scheduling Phases

The scheduling method developed in this section is based on several existing Resource-Constrained Critical Path Methods as discussed in Section 3.5.4. The method this section develops includes five phases as shown in Figure 4.1.

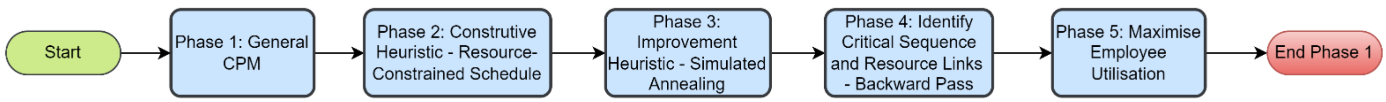


Figure 4.1 - Five Phases - Multi-Objective Resource-Constrained Critical Path Method (MORC CPM)

The first phase corresponds to the general existing CPM aiming to find a schedule not taking into account capacity restrictions. This schedule can give insights to the executor as being a best-case scenario for project completion, assuming there are no capacity constraints.

The second phase starts taking into account resource availability. In this research, the objective is to minimise project duration while aiming to have continuous resource usage and high utilisation. The reason for this is that Hegeman wants to finish the project early while not having employees unoccupied and they want to have the same number of employees working at the site as much as possible. Phase 2 serves as a constructive heuristic using a forward pass to develop a good, efficient schedule.

The third phase aims to improve the schedule developed in Phase 2 by applying the well-known improvement heuristic called simulated annealing. This method uses simulated annealing since it is an efficient method in escaping local optima and finding near-optimal solutions for complex scheduling problems in a relatively short time. By using a probabilistic acceptance strategy, Phase 3 tries to find a close-to-optimal solution.

The fourth phase identifies the critical sequence and resource links by performing a backward pass. This process starts by assigning the Latest Starting Time (LST) and Latest Finish Time (LFT) to activities having no successors and then systematically works backwards until all activities have an LFT and LST. Activities having zero slack are added to the critical sequence. Identifying resource links allows the executor to derive personnel or crane schedules from the operational schedule. For example, if Employee A is scheduled to work on Activity A on day 5, and Activity B on day 6, a resource link exists between Activity A and Activity B. The executor can then create a personnel schedule by assigning employees to activities for each day.

The fifth phase checks where in the schedule employees are unoccupied. Phase 5 deals with these unoccupied employees and checks for activities available to already assign an employee to such that these activities are more likely to finish on time.

4.4.2 Phase 1 – General CPM

This section develops Phase 1 of the MORC CPM. The purpose of Phase 1 is to create a schedule that minimises project duration without considering capacity constraints. While this phase is based on the well-known existing Critical Path Method (CPM), it includes two additions:

1. **Fractional precedence relations**: This is the fraction of an activity that should be finished before another activity can start.
2. **Release Dates**: Activities cannot be scheduled before their assigned release dates.

The steps of Phase 1 are shown in Figure 4.2[6].

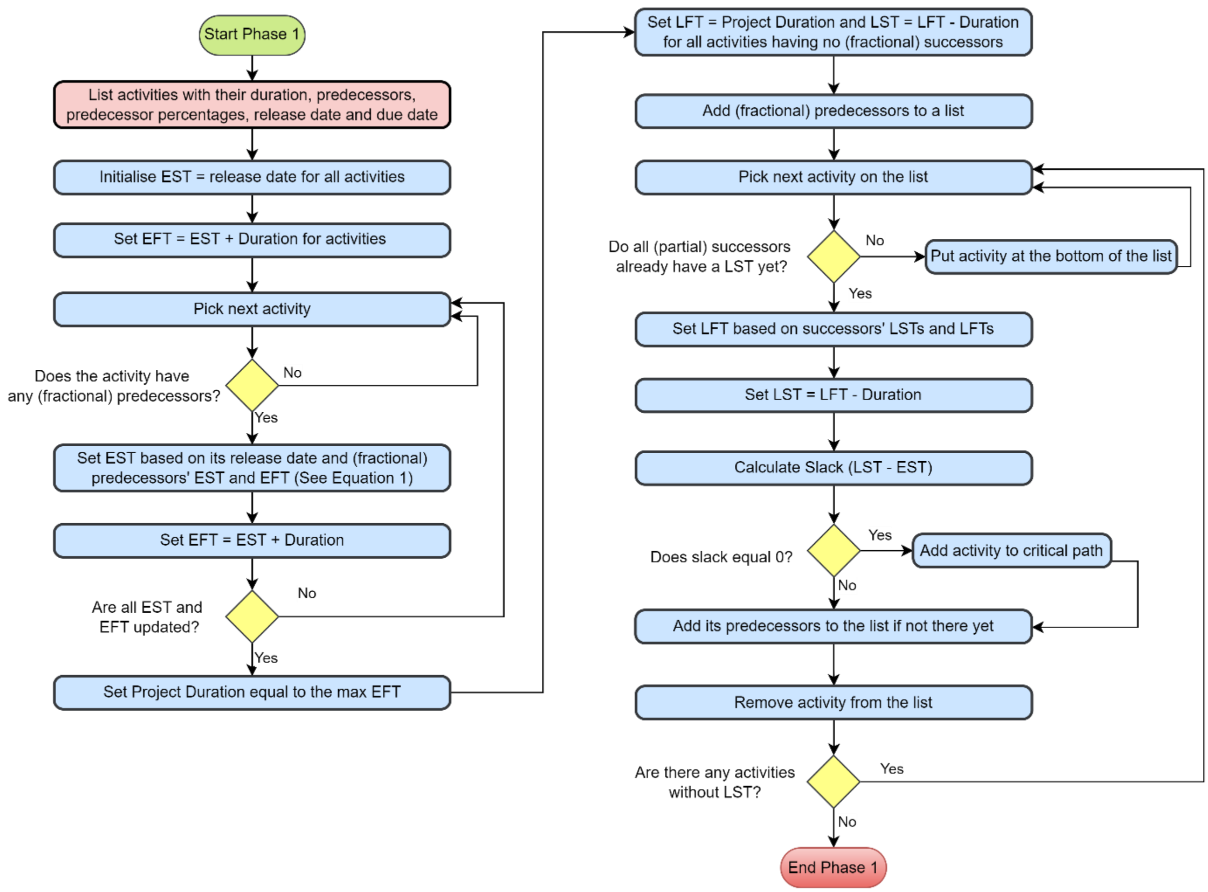


Figure 4.2 - Phase 1 - General CPM

This phase starts with the executor listing all activities, including relevant input information discussed in Section 4.2. The next step is for the tool to perform a forward pass, during which each activity gets an Earliest Start Time (EST) and an Earliest Finish Time (EFT) assigned based on its dependency on its predecessors. The forward pass starts by setting EST equal to the release date for all activities. If an activity has no release date, its EST is set to 0. The EFT is then initialised by adding the duration to the EST.

The procedure continues by selecting the first activity and checking if it has any (fractional) predecessors. If no (fractional) predecessors exist, the EST and EFT remain unchanged since the activity can immediately start after its release date. If the activity does have any (fractional) predecessors, the EST should be set based on the following equation:

[6] Red boxes refer to manual actions to be performed by the user. Blue boxes refer to actions to be performed by the tool.

$$EST_i = \max\left(ReleaseDate_i, \quad \max_{j \in pred(i)}(EST_j + PF_{ij} * EFT_j)\right)$$

Where,

- EST_i is the earliest start time of activity i
- PF_{ij} is the predecessor fraction (predecessor percentage divided by 100) between activity i preceding activity j ($0 \leq PF_{ij} \leq 1$)
- $ReleaseDate_i$ is the release date of activity i
- EFT_j is the EFT of predecessor activity j
- $pred(i)$ is the set of all predecessors of activity i

In this equation, the EST is set to the maximum of its release date and the adjusted finish time of its predecessors. The adjusted finish time is calculated as the EST of each predecessor plus the product of its predecessor fraction (PF) and its EFT. This adjustment ensures that the activity cannot start until the required portion of its predecessor is complete.

The next step is to calculate the EFT for the activity by adding the duration to the EST. When all EFTs are calculated, the project duration can be determined. The project duration should be set equal to the maximum of all EFTs which is the moment the latest activity is finished.

After doing the forward pass, a backward pass is necessary to find the LST and LFT of all activities to calculate the slack and check for how long an activity can be delayed. The first step of the backward pass is to set LFT equal to the project duration and calculate LST by subtracting the duration from the LFT for all activities having no (fractional) successors. After that, the (fractional) predecessors of these activities should be added to a list. Then, the first activity on the list should be picked and it should be checked whether all (fractional) successors already have an LST. If this is not the case, the activity should be put at the bottom of the list to be treated later. Then, the next activity on the list should be picked. This process should be repeated until an activity is found from which all (fractional) successors already have an LST. If an activity has been found for which all (fractional) successors have an LST, the LFT of this activity should be set based on the (fractional) successors' LST and LFT. As in the forward pass, this is simple when there are no predecessor percentages but is more complex when there are predecessor percentages. When there are fractional precedence relations, it follows the same logic as in the forward pass. The following equation can be used to calculate the LFT of the selected activity i:

$$LFT_i = \min\left(DueDate_i, \quad \min_{j \in suc(i)}(LFT_j - (1 - PF_{ij}) * Duration_j)\right)$$

Where:

- LFT_i is the latest finish time of activity i
- $DueDate_i$ is the due date of activity i
- PF_{ij} is the predecessor fraction (predecessor percentage divided by 100) for activity i and successor activity j ($0 \leq PF_{ij} \leq 1$)
- LFT_j is the latest finish time of successor activity j
- $Duration_j$ is the duration of successor activity j
- $suc(i)$ is the set of all successors of activity i

Next, the LST of the activity can be calculated by subtracting the duration from the LFT. Then, the slack can be calculated by subtracting EST from LST. If the slack equals 0, the activity should be added to the critical path. The next step is to add the (fractional) predecessors of the activity to the list if those are not yet added to the list because of another activity. After that, the current activity should be removed from the list. If not all activities have an LST yet, the next activity on the list should be considered and all steps should be repeated. If all activities have an LST, phase 1 is finished and a schedule without capacity restrictions has been created.

4.4.3 Phase 2 – Constructive Heuristic - Resource-Constrained Schedule

The second phase of MORC CPM aims to create a feasible schedule constrained by the capacity of the resources using a constructive heuristic. This phase aims to create a schedule with a project duration that comes as close as possible to the project duration from the schedule of Phase 1 in which no resource capacities are considered yet. Input for this phase is the schedule created in Phase 1. A step-by-step approach should adapt the schedule from Phase 1 to an effective usable resource-constrained schedule in Phase 2. The steps of Phase 2 are depicted in Figure 4.3.

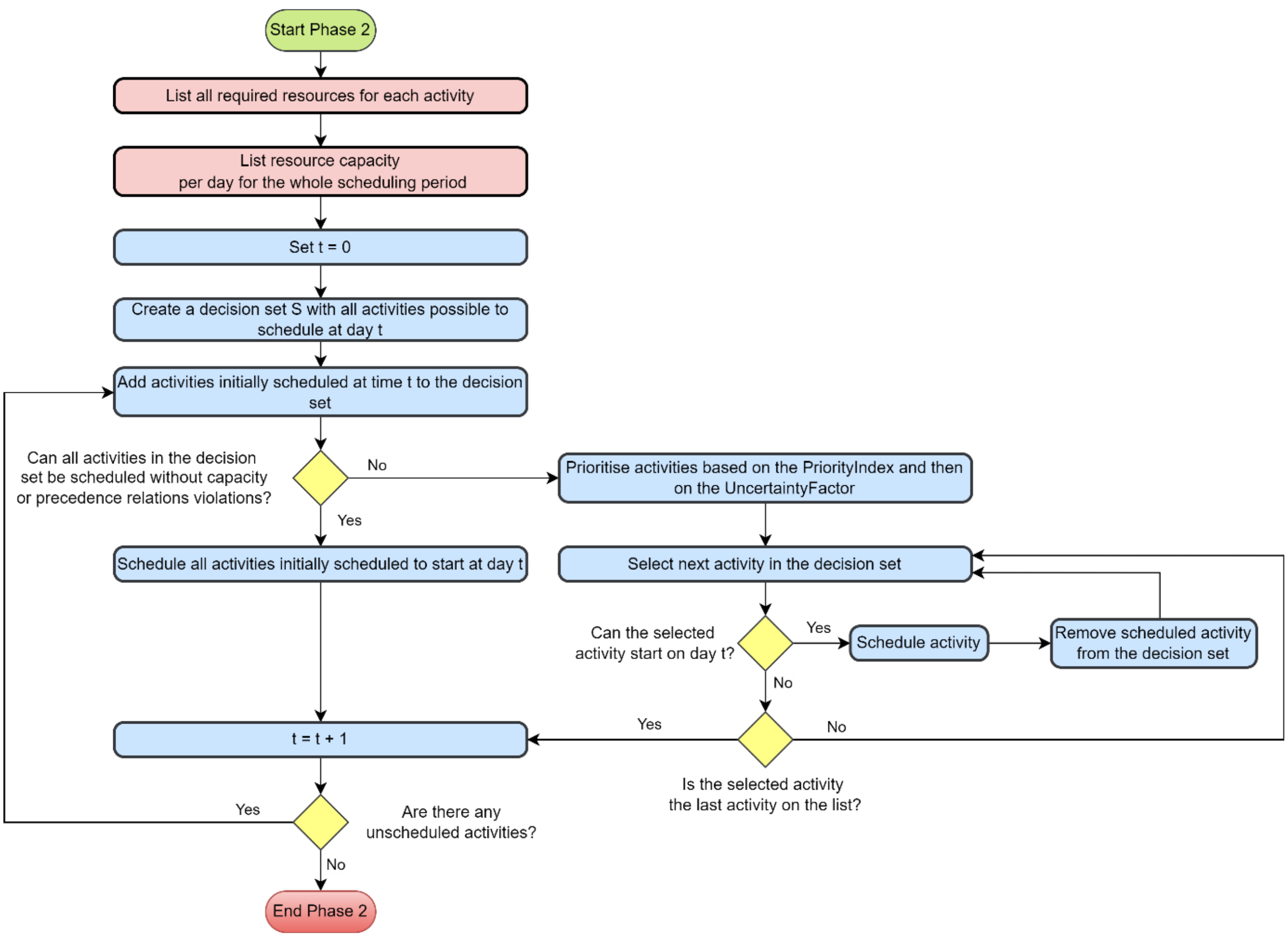


Figure 4.3 - Phase 2 - Resource-Constrained Forward Pass

Phase 2 starts with the executor listing all required resources for each activity and then specifying the resource capacity for the schedule horizon of the operational schedule. In general, the daily capacity is stable over weeks as it is undesired to frequently switch employees between different projects. However, daily capacity can vary because every employee of Hegeman has one free day once every two weeks. From this point, the algorithm should be performed by a tool and do not require manual action from the executor.

First, the time counter t should be set equal to zero allowing for day-by-day considerations. After that, a decision set S for activities to schedule should be created. Then, the activities initially scheduled in Phase 1 to start at time t should be added to the decision set. If all activities initially scheduled at time t can start without capacity and precedence relations violations, these activities should all start at time t. Then, t should be incremented by 1. If not all activities are scheduled yet, the same steps should be followed again.

If it is not possible to schedule all activities in the decision set at time t, activities with higher priority should be scheduled first. This research introduces a priority index calculated as:

$$PriorityIndex = Max\{UncertaintyFactor * \ (DaysonList - SlackPhase1), 0\}$$

Where:

- SlackPhase1 is the slack time assigned to the activity during Phase 1. This represents the number of days an activity may be delayed to have no impact on succeeding activities.
- DaysOnList is the number of days an activity is in the decision set at day t.

The priority index prioritises activities with low slack and high uncertainty since these are more likely to cause delays. The calculation multiplies the uncertainty factor by the difference between the days the activity is in the decision set and its slack time from Phase 1. Activities are prioritised if:

- They are in the decision set for a long time.
- They have low slack meaning they are more critical.

If the slack is greater than or equal to the number of days the activity is in the decision set, the priority index is set to 0. This indicates that the activity can still consume its slack days without immediate scheduling priority.

The activity with the highest priority index should be selected and checked whether it can start on day t. If this is possible, this activity should be scheduled on day t and should be removed from the decision set. Then, the next activity in the decision set should be checked whether it can be scheduled. If an activity cannot be scheduled due to its precedence relations or capacity constraints, the next activity should be considered until all activities have been checked. If all activities have been checked, the day counter should be increased by 1 and activities initially planned on the new day t should be added to the decision set. Then all steps should be followed again until all days have been considered and all activities have been scheduled.

4.4.4 Phase 3 – Improvement Heuristic – Simulated Annealing

Phase 3 aims to improve the initial schedule created in Phase 2 using simulated annealing (SA) as the improvement heuristic. Section 3.6 introduced the simulated annealing algorithm, which this phase applies to improve the schedule further.

To be able to find feasible schedules, the selection of the neighbourhood operator is important. Possible neighbourhood operators in scheduling problems are swap, move, insertion, and inversion (Lin & Yu, 2012). Additionally, various (smart) variants of these operators can be developed (Han et al., 2019).

The move operator is unsuitable for this problem context because relocating an activity typically results in capacity violations because of the high utilisation in the constructive and neighbour schedules.

Furthermore, the swap operator makes maintaining feasibility difficult in this problem context. Since the starting times of two activities change, swaps require double the number of feasibility checks making it less likely to find a feasible solution compared to the insertion operator. Moreover, since activities can be highly connected because of precedence relations, it can be difficult to find two activities that have no (indirect) connection with each other.

The insertion operator seems to be best applicable in this problem context. The insertion operator selects an activity and relocates it in the sequence of activities. Compared to the move operator, the insertion operator shifts the activities scheduled after the selected activities. This reduces the likelihood of capacity violations compared to the move operator.

The last neighbourhood operator to consider is the inversion operator. The inversion operator reverses the order of a subset of activities in the schedule. Since this problem context includes a high number of precedence relations, reversing activities increases the likelihood of precedence violations. Therefore, the inversion operator is not very logical to use in the scenario.

The method uses two different operators aiming to find good solutions. This research switches between the following operators:

1. Insertion
2. Smart insertion

The smart insertion operator can help when the algorithm has difficulties finding better solutions. Making the insertion operator “smart” means that it makes more informed decisions aiming to increase the likelihood of finding feasible schedules. The smart insertion operator prioritises activities scheduled on days with relatively high resource utilisation and schedules the activity with the lowest possible starting time while the insertion operator puts the activity at a random EST.

Figure 4.4 shows the simulated annealing heuristic, adapted to the problem in this research. Phase 3 starts by setting the schedule of Phase 2 as the initial best schedule BS and the current schedule CS and initialising ml = 1 and k = 1. The next step is to generate a neighbour schedule NS using a neighbourhood operator based on the current value of k. The operators are:

- k = 1: Insertion
- k = 2: Smart Insertion

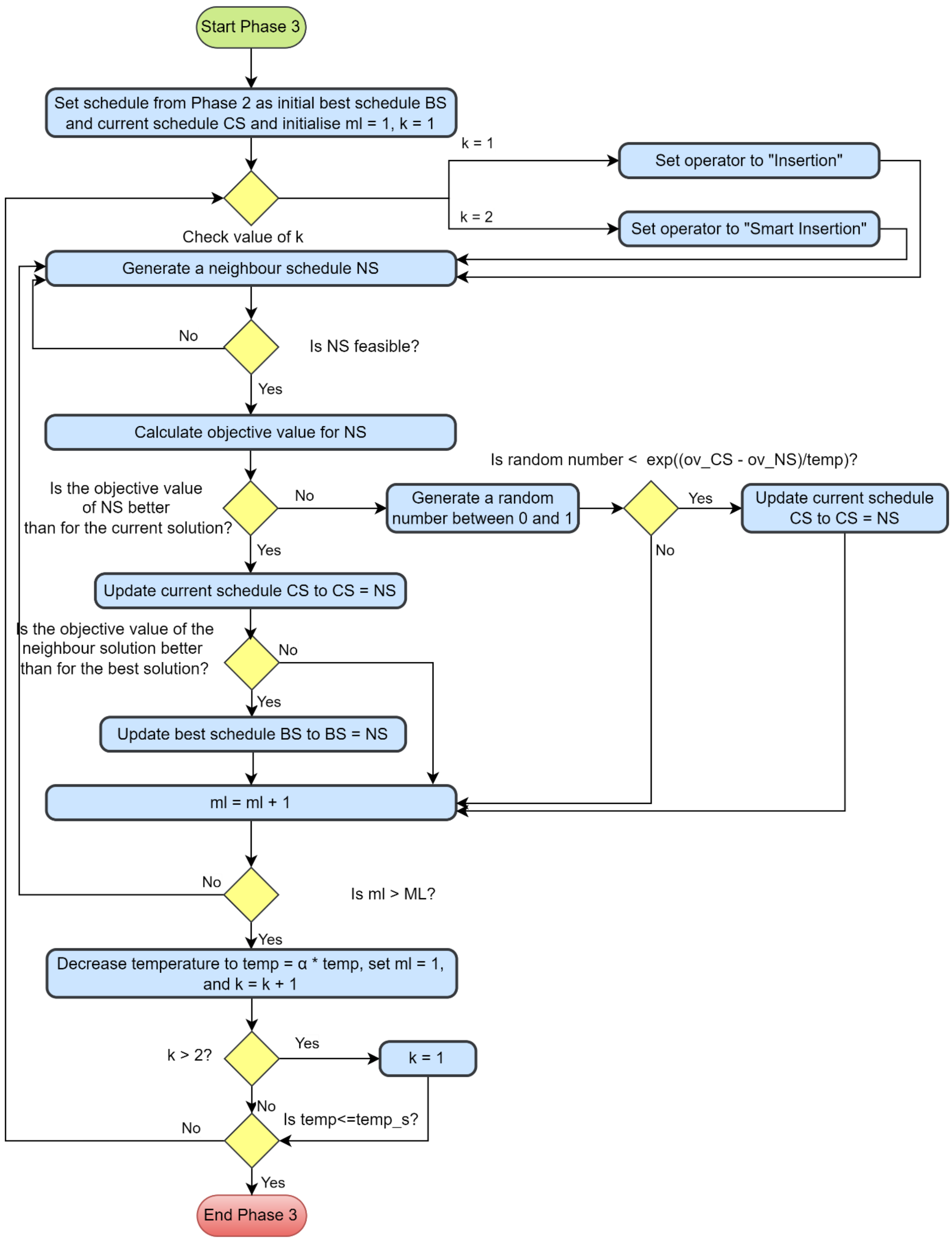


Figure 4.4 - Phase 3 - Improvement Heuristic - Simulated Annealing

After ml iterations, the neighbourhood operator switches. The next step is to generate a neighbour schedule NS by applying the selected operator. A neighbour schedule is feasible when no capacity or precedence restrictions are violated. If the neighbourhood schedule is infeasible, a new neighbourhood schedule should be generated until it finds a feasible one.

If this objective value for schedule NS is better than the objective value of schedule CS, the neighbour schedule becomes the current schedule CS. If the objective value is also better than the objective value of BS, then the neighbour schedule becomes the best schedule BS. If the objective value is not better than the objective value of CS, the neighbour schedule NS is accepted as the new current schedule CS by a certain probability, i.e. when the following formula is true:

$$randomnumber \leq \mathrm{e}^{\left(\frac{ov_{CS}-ov_{NS}}{t}\right)}$$

Where,

- $randomnumber$ is a random number between 0 and 1
- ov_{CS} is the objective value of the current schedule CS
- ov_{NS} is the objective value of the neighbour schedule NS

The Markov chain counter ml is then incremented by 1. If ml equals the Markov chain length ML, the temperature is decreased by multiplying the current temperature by the decrease factor α, ml is reset to 1, and k is incremented by 1. If k exceeds the number of operators, it is reset to 1. If ml does not equal the Markov chain length (ML), a new neighbour schedule should be created and the steps should be repeated. This process repeats until the temperature is lower than or equal to the stopping temperature $Temp_s$. After that, Phase 3 has come to an end and the best schedule BS is the final schedule.

4.4.5 Phase 4 – Identify Resource Links and Critical Sequence – Backward Pass

The objectives of the fourth phase are to identify resource links, to get the critical sequence, and to derive a personnel and crane schedule. The term resource links has been introduced in Section 4.4.1, and the term critical sequence has been explained in Section 3.5.4. To fulfil the objectives of Phase 4, this phase applies a backward pass. This phase does not change the schedule from Phase 3 but aims to give necessary information to the executor. The process of Phase 4 is shown in Figure 4.5.

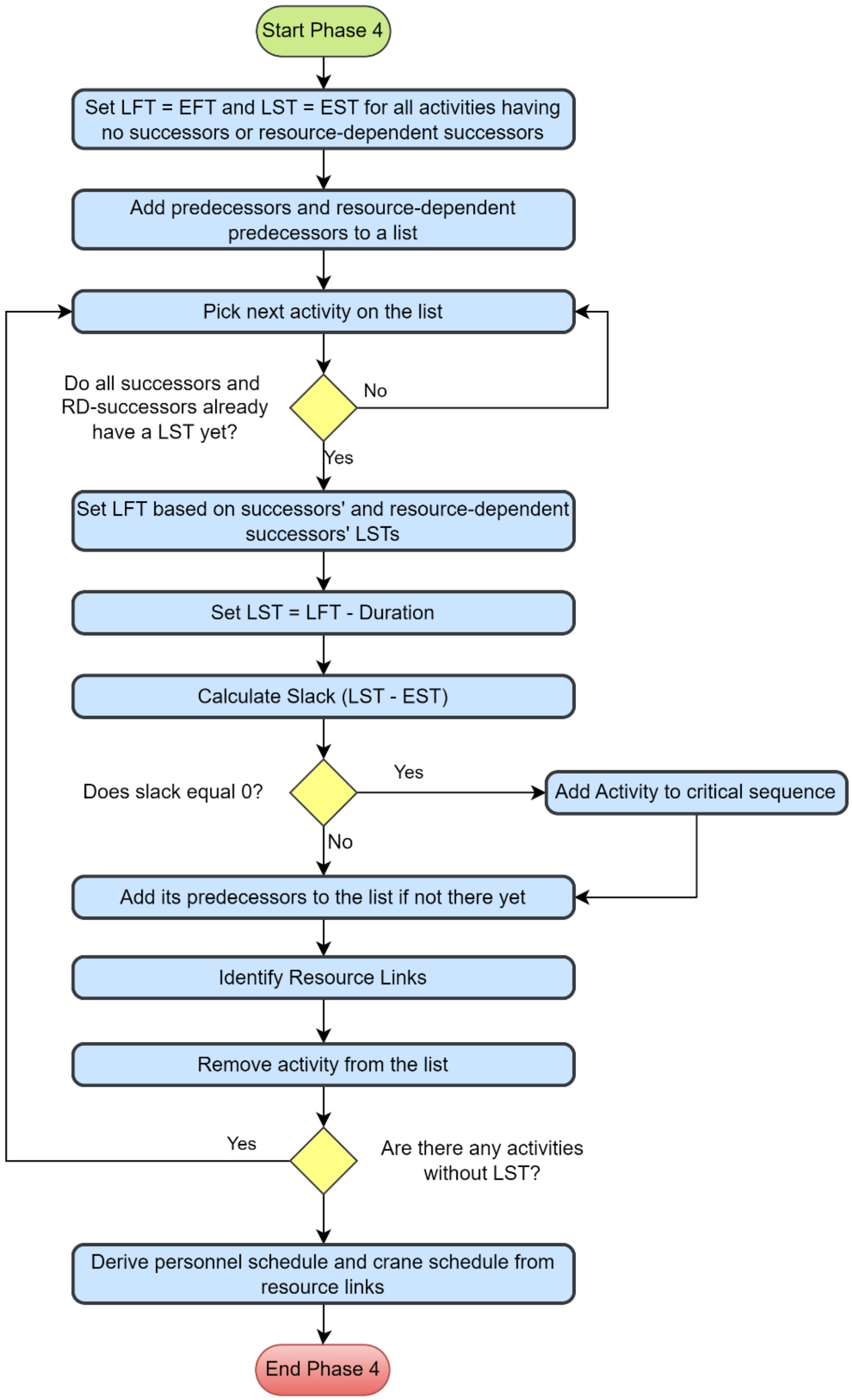


Figure 4.5 - Phase 4 – Identifying Resource Links and Critical Sequence

Phase 4 follows the same logic as the backward pass from Phase 1, but now resource dependencies are taken into account. Doing a backward pass starts by setting all LFT equal to EFT and all LST equal to EST for all activities having no (fractional) (resource-dependent) successors. The next step is to add all (fractional) (resource-dependent) predecessors of these activities to a list and pick the first activity on the list. For that activity, a check is necessary about whether all (fractional) (resource-dependent) successors of this activity already have an LST. If this is not the case, the next activity on the list should be considered. If all (fractional) (resource-dependent) successors have an LST, the LFT should be set based on the (fractional) (resource-dependent) successors' LSTs and LFTs. This follows the same logic and uses the same equation as in Phase 1:

$$LFT_i = \min\left(DueDate_i, \quad \min_{j \in suc(i)}(LFT_j - (1 - SF_{ij}) * Duration_j)\right)$$

Where:

- LFT_i is the latest finish time of activity i
- $DueDate_i$ is the due date of activity i
- SF_{ij} is the successor fraction (successor percentage divided by 100) for activity i and (fractional) successor or resource-dependent successor activity j ($0 \leq SF_{ij} \leq 1$)
- LFT_j is the latest finish time of (fractional) successor or resource-dependent activity j
- $Duration_j$ is the duration of (fractional) successor or resource-dependent activity j
- $suc(i)$ is the set of all (fractional) successor and resource-dependent successors of activity i

The next step is to calculate LST by subtracting the duration from the LFT. Then, the slack can be calculated by subtracting EST from LST. If the slack equals zero, the activity should be added to the critical sequence[7]. Next, the (fractional) predecessors and resource-dependent predecessors should be added to the list if these are not there yet. Then, resource links can be identified by checking what resource has been assigned to what activity at what moment. For each resource, a timetable can be created showing to what activity the resource is assigned at what moment. Between the subsequent activities, a link can be set indicating to what activity a resource is assigned next.

After identifying the resource links, the current activity can be removed from the list. If any activities remain without an LST, the same steps should be followed for the next activity on the list. If all activities do have an LST, a personnel schedule and a crane schedule can be derived from the resource links. When the personnel and crane schedules have been derived, Phase 4 is finished.

4.4.6 Phase 5 – Maximise Employee Utilisation

Phase 5 assigns activities to idle employees. Having idle employees is undesired since the executor does not want to tell them to stay home or to let them do nothing. For that reason, this phase aims to find activities idle employees can already start with. This has two main advantages. First of all, it gives work to unscheduled idle employees. Moreover, activities already started have a higher chance of finishing on time, increasing the likelihood of meeting the delivery date.

For example, it could be that on day 5, two employees are idle. This could be because all activities available to start require at least three employees. However, two employees can already do some work probably. By assigning these employees already to that activity, the activity is more likely to be finished on time causing a higher chance of finishing the project on time.

This phase tells the executor what day employees are idle and what they can do with the idle time. In Phase 5, altering the schedule is avoided for two reasons. First of all, it makes the method and output

[7] As explained in Section 3.5.4, when resource capacities are considered the term critical path becomes the therm critical sequence

more complicated. Moreover, it makes the schedule tight causing a delayed activity to immediately require rescheduling, so there are no small buffers anymore. Figure 4.6 shows the steps of Phase 5.

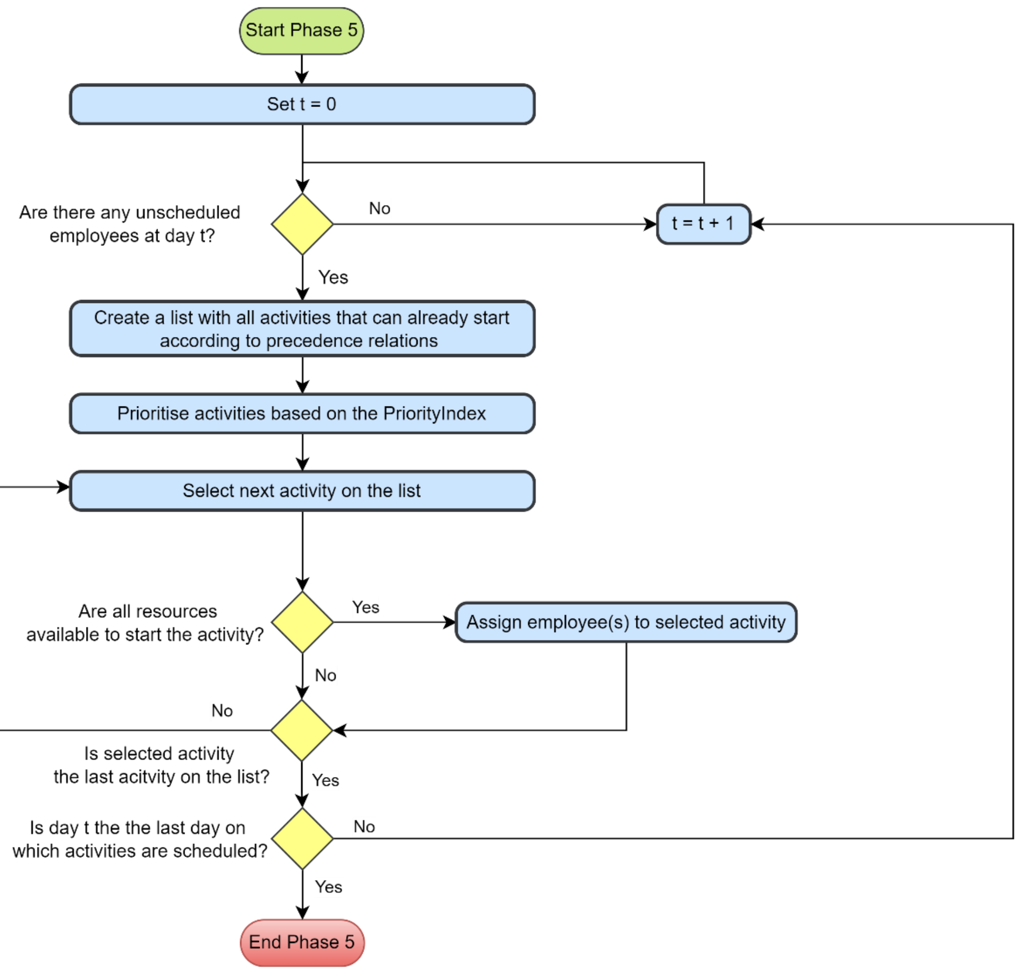


Figure 4.6 - Phase 5 – Maximise Employee Utilisation

Phase 5 starts by setting the day counter t to zero. Then for day t, it should be checked whether there are any unscheduled employees at day t. If this is not the case, t should be incremented and the same check should be done for the next day. This process should be repeated until a day has been found at which there are any unscheduled employees. Then, a list should be created with all activities that can start according to their precedence relations. Then these activities should be prioritised based on the priority index. Prioritising happens based on a priority index to ensure that uncertain activities have employees assigned earlier than their earliest start time. This should increase the likelihood of timely project completion. This priority index is the same as the priority index in Phase 2 with the only difference that it considers the slack of the current schedule rather than the slack from the schedule in Phase 1:

$$PriorityIndex = Max\{UncertaintyFactor * (DaysonList - Slack\ best\ schedule), 0\}$$

The next step is selecting the first activity on the list and determining whether all required resources, excluding employees, are available. Then, it must be verified whether the selected activity is the last activity on the list. If it is not, the process continues with the next activity. If the activity is the last activity on the list, it is necessary to check if day t is the final day. If it is not the final day, t should be incremented by 1. If it is the last day, Phase 5 ends.

4.5 Conclusion

Chapter 4 developed the Multi-Objective Resource-Constrained Critical Path Method (MORC CPM) for creating operational schedules. This chapter started with a problem description and a list of the necessary inputs for the MORC CPM, followed by the assumptions and simplifications made. After that, the MORC CPM consisting of five phases was developed. The MORC CPM aims to systematically create a workable operational schedule to apply in utility construction projects.

In Phase 1, the general CPM which does not take capacity restrictions into account has been applied. This phase laid the foundation for the subsequent phases and serves as an unrealistic upper bound. In Phase 2, a constructive heuristic makes the schedule from Phase 1 feasible using a Resource-Constrained Scheduling (RCS) forward pass. Phase 3 then aims to improve this schedule by applying a simulated annealing heuristic. Phase 4 identifies the critical sequence of activities and the resource links by a backward pass. This allows deriving the personnel and crane schedules from the operational schedule. Phase 5 focuses on assigning employees to activities on days with less than 100% employee utilisation. This is to make sure no employees are unoccupied and it increases the chance the project is finished on time. After the method has been developed, this chapter explains the tool that has been build to apply the MORC CPM.

This systematic approach enables executors to create effective and realistic operational schedules for utility construction projects with minimal effort.

5 Schedule Revision Method Development

This chapter develops a method for revising schedules after a disruption. Section 5.1 gives a problem description being the foundation for the development of the method. Section 5.2 discusses the assumptions and simplifications for this method. Subsequently, Section 5.3 develops the Schedule Revision Method (SRM).

5.1 Problem Description

When the executor creates an operational schedule, he is aware of the potential disruptions that can impact the execution of it. Disruptions can have minor impacts that can be solved easily but in most cases, the current schedule is not feasible anymore. To make the schedule feasible again, it should be revised. Disruptions can include unworkable weather, equipment failures, material shortages, unforeseen site conditions, and more. These disruptions can have negative impacts on the overall plan progression and impact delivery dates.

A delay of one activity may prevent its successors from starting on their scheduled dates. Moreover, extended use of resources because of delays can cause bottlenecks since these resources are not released on time for other scheduled activities. So, the delay of one activity can impact a lot of activities. This requires a standardised method to minimise the impact of the disruption on the schedule and project completion.

Currently, executors revise schedules based on individual experience and intuition resulting in different levels of efficiency between different executors. Some executors may shift the entire schedule by several days or weeks, while other executors find more creative solutions such as reordering activities.

Another challenge when revising schedules is to minimise the changes to the schedule. Extensive changes to the schedule can have a lot of impact on stakeholders like subcontractors, clients and suppliers since they have to change their plans and schedules as well, leading to dissatisfaction and potential conflicts.

The combination of limiting the impact on the project duration and the necessity to not change a lot requires a systematic approach to revising schedules. The key objectives of the method in this chapter are:

1. Minimise the impact the disruptions have on the project duration
2. Minimise the rescheduled number of activities

The method should also consider that it may not always be possible to reschedule certain activities. Reasons for this could be that the activity is scheduled shortly or that the subcontractor is not able to find another time slot to execute the activity. Another reason could be that materials require much storage space making it undesired to change their starting time since the storage space is then occupied longer. There could be a lot more reasons for an activity being not allowed to be rescheduled. The method should be able to take this into account.

5.2 Assumptions and Simplifications

To create a practical and efficient method for schedule revision, certain assumptions and simplifications are necessary:

- Once an activity, not being the disrupted activity, has already started on the day of the disruption, it cannot be rescheduled.
- Splitting activities is not allowed. For example, if the duration of an activity is five days, it is not allowed to schedule the first three days and then the last two days ten days later.
- Activity durations are assumed to be deterministic.
- Subcontractor activities do not require additional work from Hegeman.
- Subcontractors are assumed to be always available when rescheduling. This means that they always agree with the revised schedule.
- All input information provided by the executor is considered accurate and reliable.
- All workers have the same skill level and can perform any task. If it is necessary to distinguish worker types, another resource type can be created.
- Setup time between activities is not considered so assumed to be 0.
- There can be only one disruption. In reality, there could be more disruptions simultaneously but the method cannot handle this.

5.3 Schedule Revision Method

When a disruption occurs, the executor can use the method this chapter develops as shown in Figure 5.1 to deal with the disruption systematically.

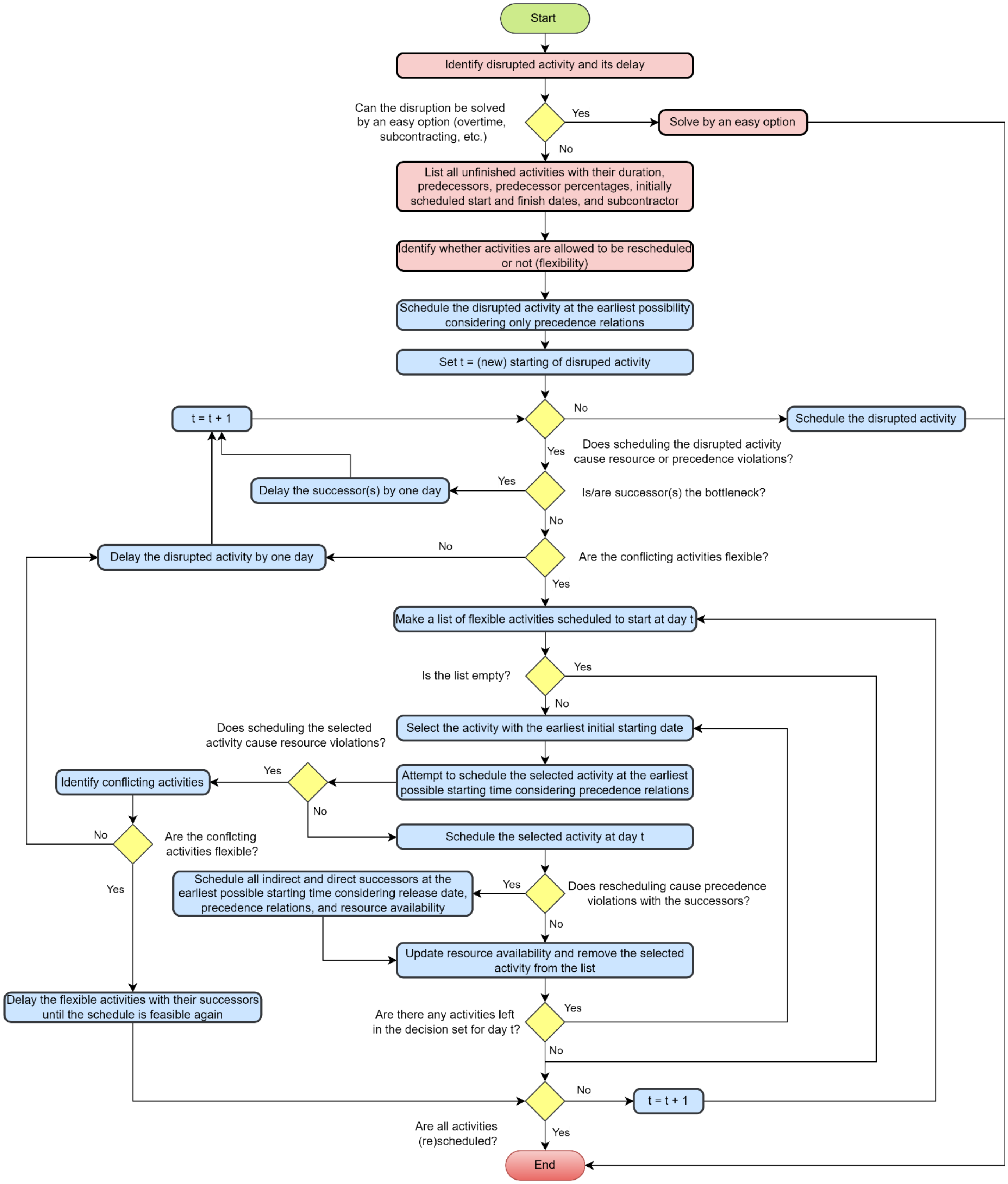


Figure 5.1 - Schedule Revision Method

The method begins with identifying the disrupted activity and assessing its impact. The executor then determines whether the disruption can be resolved by one of the easy options:

- Letting employees work overtime
- Getting additional employees for some days
- Subcontracting activities
- Increasing resource capacity for some days
- Expediting delivery

If the disruption can be resolved by one of these solutions, the executor should implement it. If not, the executor must list all activities with their duration, predecessors, predecessor percentages, initially scheduled start and finish dates, and subcontractor. The executor also needs to identify which activities can be rescheduled and which cannot. Up to this point, all steps are to be performed by the executor. From this moment on, all steps are part of the algorithm.

The algorithm first schedules the disrupted activity at the earliest possible day considering only precedence relations. This approach initially ignores resource capacities to avoid pushing the disrupted activity to the end of the project, which could cause a lot of precedence violations and revise the schedule drastically.

After scheduling the disrupted activity, the algorithm checks if scheduling this causes any resource or precedence violations. If there are no violations, the disrupted activity is scheduled without further action. If violations occur, the algorithm determines if the successors of the disrupted activity are the bottleneck.

If one or multiple successors are the bottleneck, they are delayed by one day and the check for resource and precedence violations should be repeated. If the successors are not the bottleneck, the algorithm checks if the conflicting activities are flexible or not.

If the disrupted activities are not flexible, the disrupted activity should be delayed by one day since these have no priority over unflexible activities. If conflicting activities are flexible, the algorithm creates a list of all flexible activities scheduled to start on day t. If this list is empty, the day counter should be incremented by one and the process repeats until the list is not empty. When the list is not empty anymore, the activity with the earliest starting date is selected.

The selected activity is then scheduled at the earliest possible starting time, considering only precedence relations. If this does not cause resource violations, the activity is scheduled for day t. Then, the algorithm checks if there are any precedence violations with the successors of the scheduled activity. If there are precedence violations, all direct and indirect successors are scheduled at the earliest possible starting time considering precedence relations and resource availability. Then, the resource availability should be updated. After that, the next activity on the list should be considered if there are still activities left.

If attempting to schedule the disrupted activity causes resource violations, the conflicting activities should be identified and considered whether they are flexible or not. If they are not flexible, the disrupted activity should be delayed. If they are flexible, the flexible activities with their successors should be delayed until the schedule is feasible again.

The order of priority for handling activities is:

1. **Unflexible activities**: These activities are scheduled first since moving them is undesired.
2. **Disrupted activity**: To limit the impact on the initial schedule, the disrupted activity should not be delayed too far. This also increases the chance of precedence violations if the disrupted activity has successors.
3. **Direct and Indirect successors of disrupted activity**: To maintain the flow of the project and to limit the impact on the initial schedule, the direct and indirect successors should have priority over flexible activities.
4. **Flexible activities**: These activities have the least priority since executors prefer postponing these instead of the other activities.

5.4 Conclusion

This chapter developed a method to revise operational schedules after a disruption. The method can be applied in cases where simple solutions such as working overtime, assigning additional employees, subcontracting, renting more equipment, or expediting delivery are not possible. This method provides a step-by-step approach to rescheduling activities while taking into account flexibility and resource availability. By a day-by-day consideration approach, the SRM minimises the impact of disruptions on the operational schedule without creating an entirely new schedule deviating a lot from the current schedule.

6 Implementation

This chapter discusses the implementation of both methods developed in Chapters 4 and 5 into a tool. Section 6.1 discusses the implementation of the Schedule Creation Tool with the MORC CPM developed in Chapter 4. Section 6.2 discusses the implementation of the Schedule Revision Tool developed in Chapter 5.

6.1 Schedule Creation Tool

One of the main objectives of this research is to generate a method executors can use to create an operational schedule. To apply the method, a tool is needed. Furthermore, the development of a tool in which the method is implemented allows for experiments and analysis to test the quality of the method.

Python is the programming language executing the method. Before Python can be used to generate a schedule, input information should be filled in somewhere. The executors can store the input information in Excel which makes the tool recognisable and easy to use. Then, the Python code reads the input information and applies MORC CPM to the input information. After running the code, a dashboard visualises all relevant outputs including the schedule and the utilisation. Figure 6.1 represents the process of the tool.

Figure 6.1 - Tool Process

The executor who uses the tool should fill in four sheets with information shown in Appendix A. In the first sheet, the executor should fill in the name of the activity, the duration, the release date, the due date, the uncertainty factor, whether a delivery is required, and the delivery time for these deliveries. The last two things to fill in do not influence the method but filing them in allows the tool to give the executor the delivery date.

In the second sheet, the executor should fill in the predecessors with corresponding predecessor percentages. The tool allows for at most nine predecessors per activity.

In the third sheet, the executor should fill in the resource capacities per day. The executor can fill in at most fifteen different resource types. In reality, executors only consider at most four resource types when scheduling.

In the fourth sheet, the executor should fill in the resource requirements. Per activity, the executor should indicate how many units of one resource are necessary.

After filling in the input in Excel, the executor can let the Python code run which is the last manual action the executor has to take. The Python code reads the name and the path to the workbook and reads all required input information from the Excel file. Then, each phase of the MORC CPM is systematically executed by the tool. When all steps are performed, the Python code manipulates the information such that it can be represented in a dashboard. For this dashboard, a URL has been created to show it. An example of this dashboard is shown in Appendix B.

6.2 Schedule Revision Tool

This section discusses the implementation of the schedule Revision Tool. The purpose of this Schedule Revision Tool is to simplify the process of revising operational schedules. The tool assists executors in revising schedules using the method developed in Chapter 5.

As for the Schedule Creation Tool, Python is the programming language executing the method. The process from input to output works in the same way as for the Schedule Creation Tool as shown in Figure 6.1. Several Excel sheets should be filled in by the executor such that the tool can generate a schedule. These sheets with example information can be found in Appendix C. Then the Python code executes the method and shows the output schedule with relevant outputs in a dashboard. An example of the dashboard can be found in Appendix D.

The executor needs to fill in five Excel sheets before the tool can execute the method. In the first sheet, the executor must store the activities with relevant their relevant information. For each activity, the executor must indicate whether rescheduling is allowed, whether the activity already started, the days still to be scheduled, the subcontractor, the initial start and end dates, and the uncertainty factor.

The second, third, and fourth sheets are the same as for the Schedule Creation Tool. In the second sheet, the executor should fill in the predecessors with corresponding predecessor percentages. In the third sheet, the executor should fill in the resource capacities per day. In the fourth sheet, the executor can fill in what activities require what resources and how much of them.

In the fifth sheet, the executor should fill in the disruption together with its disruption date and the impact of the disruption which is the number of days the activity is delayed.

6.3 Conclusion

This chapter detailed the implementation of the scheduling methods into two practical tools. The Schedule Creation Tool applies the MORC CPM developed in Chapter 4 while the Schedule Revision Tool applies the Schedule Revision Method developed in Chapter 5.

Both tools use Python for executing the scheduling methods with Excel serving as an interface for the input information. The Python code generates dashboards to show the schedules with corresponding information such as start and finish times, durations, and utilisation.

7 Experimental Analysis – Schedule Creation Method

This chapter conducts experiments to evaluate the performance of the MORC CPM, with a primary focus on the simulated annealing heuristic. The emphasis is on assessing the improvements made compared to the constructive heuristic since the simulated annealing component is crucial for the efficiency and reliability of the method. The experiments are designed to demonstrate that the MORC CPM can enhance scheduling efficiency and reliability. First, Section 7.1 outlines the experimental design for this chapter. Then, Section 7.2 designs the problem instances. Next, Section 7.3 determines the initial settings. Section 7.4 tunes the parameters from the initial settings. Lastly, Section 7.5 performs several different sensitivity analyses including varying the number of activities, the employee capacity, and the weights of the objective function.

7.1 Experimental Design

This section explains the experimental design of this chapter and serves as a guide through the chapter. Table 7.1 shows the experiment steps to be performed in this chapter. Each experiment step has a specific goal.

	Experimental Design Steps	**Goal**
1	Define Problem Instances	Create and generate different scenarios regarding the level of connectivity
2	Determine Initial Settings	Establish a starting point for tuning the parameters
3	Parameter Tuning	Optimise parameters for best performance for the different scenarios
4	Sensitivity Analysis – Varying the Number of Activities	Test and understand the scalability of the MORC CPM
5	Sensitivity Analysis – Varying the Employee Capacity	Test the impact of different employee capacities on the objective function and the running time.
6	Sensitivity Analysis – Varying Weights	Understand the impact of different parts of the objective function.

Table 7.1 - Experimental Design – Schedule Creation Method

Step 1 defines four different problem instances with varying levels of connectivity. The level of connectivity refers to the number of precedence relations. Step 2 determines the initial settings for the simulated annealing parameters. These parameters include the initial temperature, the Markov chain length, the decrease factor, and the stopping temperature. Step 3 tunes the parameters such that there is a good balance between the quality of the solutions and the running time.

Steps 4 to 6 perform several sensitivity analyses to test what impact several parameters and settings have on the MORC CPM. Step 4 tests various number of activities to test the scalability of the MORC CPM. Step 5 tests different employee capacities to assess what this does with the objective value and the running time. Finally, Step 6 varies the weights of the objective function terms to understand their impact.

7.2 Problem Instances

The performance of the MORC CPM may depend on the level of connectivity from the problem instance. To test this, this section creates four different problem instances with different levels of connectivity. The input information for these problem instances should reflect realistic project scenarios while allowing for controlled experimentation. The input information for these instances is based on typical project characteristics observed in similar construction projects from Hegeman. All problem instances include 40 activities with the following properties:

- The durations of the properties are drawn from a Uniform distribution: D ~ U(1, 5).
- All problem instances have eight subcontractors all performing a different number of activities. The number of activities per subcontractor are drawn from a Uniform distribution: N ~ U(1, 3).
- The release date for all activities is set to 0, and the due date is set to 500 since this is a large number. This approach ensures that scheduling is not constrained by release and due dates, allowing for an assessment of the performance of the method on other factors.
- The uncertainty factor is drawn from a Uniform distribution: UF ~ U(1, 5).

Each problem instance includes three different resource types with realistic capacities for an average Hegeman project:

- **Employees** – Capacity: 6
- **Crane** – Capacity: 1
- **Telehandler** – Capacity: 1

The capacity is representative of an average Hegeman project. Regarding the resource assignments, the assumptions are as follows:

- The Employee requirements for Hegeman activities[8] are drawn from a Uniform distribution E ~ U(1, 4).
- 20% of the activities require a crane. These activities are randomly selected.
- 10% of the activities require a telehandler. These activities are randomly selected.

Now that all shared activities and resource properties have been defined, four different scenarios regarding connectivity can be generated:

- **No Connectivity**: There are no precedence relations. While this is not a realistic scenario in real projects, it serves as an extreme case to show the behaviour of the method when no activities are dependent on each other.
- **Low Connectivity**: There are five clusters [9]with two or three connected activities per cluster. Each activity has at most one predecessor.
- **Medium Connectivity**: There are seven clusters with four or five connected activities per cluster. Each activity has at most two predecessors.
- **High Connectivity**: There are two clusters with twenty connected activities each. Each activity has at most three predecessors

The clusters can be found in Appendix E. Section 7.3 tests whether more connectivity leads to higher running time. Clusters are formed through the following steps:

1. **Calculate the number of activities that should not be part of a cluster:** Determine how many activities should be excluded from clustering.
2. **Random Selection:** Randomly select activities that will not be part of any cluster.

[8] Activities performed by subcontractors do not require employees of Hegeman as explained in Section 4.2

[9] In this context, clusters refer to groups of activities linked by precedence relations. These activities can be directly or indirectly connected.

3. **Forming Clusters**: After identifying activities that are not part of any cluster, create clusters by grouping the adjacent activities sequentially from low to high. For example, Activity A1 and A4 are randomly selected to not be part of any cluster, and then A2-A3-A5 form a cluster. A6-A7-A8 form another cluster. This process should be continued until all clusters have been formed.

The predecessor percentages vary between 50%, 75% and 100%, with half of the predecessor percentages being 100%, 25% being 50% and 25% being 75%. This distribution ensures a realistic representation of activity dependencies as a large fraction of activities require full completion of their predecessors. Table 7.2 shows the problem instances.

	Number of clusters (excl single activity clusters)	Cluster size (excl single activity clusters)	Number of clusters (incl. single activity clusters)
No connectivity	0	-	40
Low Connectivity	5	2-3	32
Medium Connectivity	7	4-5	15
High Connectivity	2	20	2

Table 7.2 - Problem Instances

7.3 Initial Settings for Simulated Annealing

This section focuses on determining the initial settings for the simulated annealing parameters. The initial settings serve as a starting point for the parameter tuning in Section 7.4. The following parameters require initial settings:

- Initial Temperature ($\mathbf{T_0}$)
- Markov Chain Length (**L**)
- Temperature Decrease Factor (**α**)
- Stopping Temperature ($\mathbf{T_{Stop}}$)

For simplicity, the weights for all three terms from the objective function are set to 1/3 in the initial settings. Section 7.5.3 experiments with the weights for the three parts of the objective function.

To determine the initial settings, it is important to understand the approximate size of the objective values since accepting worse solutions depends both on the current temperature and on the objective values of the current and neighbour solutions. In simulated annealing, all settings are interdependent. This means that adjusting the value of one parameter can impact the effectiveness of other settings. Therefore, careful consideration is required when choosing these parameters to ensure effective intensification and diversification.

Given that the objective values in this problem lie around 1, an initial temperature of 10 is used. This temperature is sufficiently high to explore the solution space effectively but is not so high that it accepts all solutions for a long time, which would happen if the temperature was set to for example 1000.

A Markov chain length of 100 is appropriate to start with. According to Aarts et al. (1997), the decrease factor normally lies between 0.9 and 0.98. To explore intensively in the beginning, a decrease factor of 0.95 is a good starting point. For the same reason as choosing a relatively low initial temperature, a low stopping temperature fits this research. To allow extensive intensification at the end, the stopping temperature is set to 0.001. So the initial settings to be used are:

- Initial Temperature (**T_0**) = 10
- Markov Chain Length (**L**) = 100
- Temperature Decrease Factor (**α**) = 0.95
- Stopping Temperature (**T_{Stop}**) = 0.001

The results will be compared based on various metrics, including objective value, project duration, utilisation, and resource variability. Table 7.3 shows the results of the four experiments with the initial settings. Graphs with the corresponding utilisation per resource per day are shown in Appendix F.

	Objective Value - Constr. Heur.	**Objective Value – Impr. Heur.**	**Project Duration - Constr. Heuristic**	**Project Duration – Impr. Heur.**	**Utilisation - Constr. Heuristic**	**Utilisation – Impr. Heuristic**	**Res. Var. - Constr. Heur.**	**Res. Var. – Impr. Heur.**	**Run time**
No connectivity (IT = 10, ML = 100)	0.9078	0.8220	71	64	76.94%	85,35%	0.7233	0.6632	218 seconds
Low connectivity (IT = 10, ML = 100)	0.9171	0.7886	77	65	70.94%	84.04%	0.7513	0.6776	283 seconds
Medium connectivity (IT = 1000, ML = 100)	0.9221	0.7803	78	65	70.03%	84.04%	0.7663	0.6741	670 seconds
High Connectivity (IT = 10, ML 100)	0.9444	0.7136	96	69	56.90%	79.17%	0.8333	0.7034	1,762 sec

Table 7.3 - Experiments Outcomes - Initial Settings

For the No Connectivity problem instance, all parts of the objective function have improved compared to Phase 2. It took 218 seconds to fulfil the simulated annealing heuristic. The low and medium connectivity experiments have comparable schedules after Phase 3, although the constructive heuristic from Phase 2 gave a less efficient schedule in the Medium Connectivity problem instance. This is because more (fractional) precedence relations have been introduced in this experiment. The high connectivity experiment has a comparable schedule as well, however, the constructive heuristic performed worse in this case. The constructive heuristic performs worse because it has to deal with a lot of precedence relations in the high connectivity experiment. This worse solution allows for a lot of possible improvements.

As expected, the running time increases substantially when more precedence relations have been defined. This is because the insertion and smart insertion operators require much time to make a solution feasible by shifting successors. The running time for the No Connectivity experiment is around three and a half minutes and the running time for the High Connectivity experiment is almost thirty minutes.

All experiments with the initial settings have 18,000 iterations. The left part of Figure 7.1 tracks the best objective value over the iterations for the Low Connectivity problem instance. Other experiments have graphs with a similar progression of the objective value which can be found in Appendix G. In the first couple of iterations, the objective value improves quickly by exploration. Accepting almost all solutions as new current solutions rapidly finds other promising neighbourhoods. After that, the objective value improves for the last time around 12,500 iterations. In the last 5,000 iterations, no better solutions have been found despite the extensive exploitation.

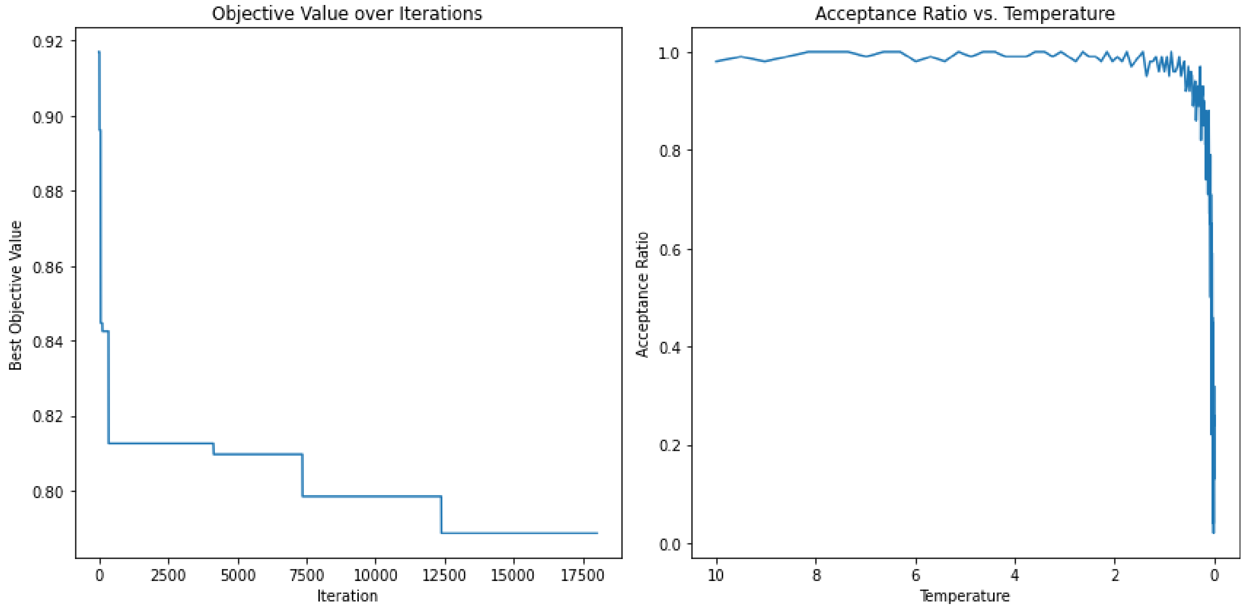


Figure 7.1 - Objective Value over Iterations (left) and Acceptance Ratio vs Temperature (right)

Next to the objective value, it is useful to track what fraction of neighbour solutions are accepted as the new current solution at which temperature. Figure 7.1 shows the acceptance ratio per Markov chain for certain temperatures. On the x-axis, the temperature is shown. On the y-axis, the acceptance ratio is shown. The Acceptance ratio reflects the number of neighbour solutions that are accepted in one Markov chain. Until a temperature of around two, almost all solutions are accepted as current solutions. This is also the part where most improvements are made. So, exploration gives a lot of new current solutions. Then, in the end, fewer neighbour solutions are accepted. But there less improvements are made.

To conclude, the initial settings seem to perform well but, for no problem instance, an improvement has been found in the last 5500 iterations. Therefore, the stopping temperature seems to be too low. Moreover, since in the beginning, the most improvements are found, more time could be invested here, and the decrease factor may seem to be too low.

7.4 Parameter tuning

This section aims to find better settings for the simulated annealing heuristic, balancing the objective value and running time. Section 7.4.1 designs the experimental setup based on the outcomes of the experiments with the initial settings. Section 7.4.2 analyses the results of the experiments. Lastly, Section 7.4.3 refines the experiments based on the outcomes and analyses them.

7.4.1 Experimental Setup - Parameter Tuning

Since the intensification in each experiment with the initial setting goes on for too long, the running time could be decreased while not getting worse solutions by increasing the stopping temperature. For that reason, all experiments get a stopping temperature of 0.01 from now on.

Based on the experiments in Section 7.3, it is difficult to draw conclusions about the initial temperature. The experiments with the initial settings quickly find improved solutions but it is unknown what happens when the initial temperature is higher or lower. Therefore, it can be valuable to test starting temperatures of 5 and 20.

It is also difficult to say what impact the Markov chain length has on the outcomes. This can be tested by lowering the Markov chain length to 50 and increasing it to 200.

Moreover, the impact of the value for the decrease factor is unknown at this stage. Therefore, it can be valuable to tune the value of the decrease factor. As discussed in Section 7.3, common values for the decrease factor are between 0.9 and 0.98. Therefore, both extremes will be tested. Based on these observations, six experiments can be designed as shown in Table 7.4.

	T_0	L	α	T_{Stop}
IT1	5	100	0.95	0.01
IT2	20	100	0.95	0.01
ML1	10	50	0.95	0.01
ML2	10	200	0.95	0.01
DF1	10	100	0.9	0.01
DF2	10	100	0.98	0.01

Table 7.4 - Experimental Setup – Parameter Tuning

To avoid complexity in this chapter, parameter tuning in this section will be conducted only on the Medium Connectivity problem instance. The outcomes of the experiments are measured against their objective values, project durations, utilisations, and resource variabilities. Moreover, the running time and the number of iterations were measured to analyse the time the experiments took to run. The outcomes of the experiments are shown in Table 7.5[10].

[10] The outcomes of the constructive heuristic are not shown since these are the same for all experiments.

	Objective Value – Impr. Heuristic	Project Duration – Impr. Heuristic	Utilisation – Impr. Heuristic	Res. Variability – Impr. Heuristic	Running time/Iterations	Number of iterations
Medium Connectivity – IT1	0.7970	66	82.77%	0.6987	570 seconds	12,300
Medium Connectivity – IT2	0.7936	66	82.77%	0.6886	653 seconds	14,900
Medium Connectivity – ML1	0.7950	66	82.77%	0.6926	290 seconds	6,750
Medium Connectivity – ML2	0.7830	65	84.04%	0.6825	1,169 seconds	27,000
Medium Connectivity – DF1	0.8153	68	80.03%	0.7023	285 seconds	6,600
Medium Connectivity – DF2	0.7803	65	84.04%	0.6741	1,539 seconds	34,200
Medium connectivity – Initial Settings	0.7803	65	84.04%	0.6741	670 seconds	18,000

Table 7.5 - Experiments Outcomes – Parameter Tuning

7.4.2 Analysis – Parameter Tuning

This section analyses the experiments conducted for parameter tuning based on important parameters including initial temperature, Markov chain length and decrease factor. Due to the interdependencies of these parameters, it is important to consider how changes in one parameter may influence the others.

Initial Temperature and Markov Chain Length

The experiments show that increasing the initial temperature from 5 to 20 slightly improves the objective value. However, the improvement is minimal. The running time is slightly higher when using an initial temperature of 20 but this difference is small (570 seconds versus 653 seconds). Because the differences are small, it is difficult to draw valid conclusions here, so leaving the initial temperature at 10 seems to be the right balance between running time and the objective value.

The choice of Markov chain length also impacts running time. A Markov chain length of 200 takes approximately four times longer than a chain length of 50 (1169 versus. 290 seconds). On the other hand, the objective value is slightly better for a Markov chain length of 200 since four times as many neighbour solutions are generated. The objective value for a Markov chain length of 50 is 0.7950 while the objective value for a Markov chain length of 200 is 0.7830. This difference is relatively small but decreasing the project duration by one day seems to be worth having a longer running time.

When considering the interaction between the initial temperature and the Markov chain length, an observation is that a higher initial temperature combined with a longer Markov chain length can enhance the exploration of the solution space. This combination allows the algorithm to explore a broad range of solutions at the beginning of the algorithm reducing the chance of ending up in a local optimum quickly. However, this enhanced exploration should be weighed against the running time. Both a Markov chain length and a high initial temperature increase the running time. The benefits of this improved exploration do not outweigh the increased running time. This suggests that an initial temperature of 10 and a Markov chain length of 100 are good settings.

Decrease Factor Compared to Other Parameters

Performing experiments with the decrease factor gives useful insights regarding the performance of the cooling scheme. A decrease factor of 0.9 gave an objective value of 0.8153 while a decrease factor of 0.98 gave the best outcome of all experiments with an objective value of 0.7803. The running time is more than five times higher when the decrease factor is 0.98 compared to when it is 0.9. However, because the objective value is much better when having a high decrease factor, it is worth the running time. Increasing the decrease factor even more can be considered in the next tuning experiments.

Figure 7.2 illustrates the objective values over time for decrease factors of 0.9 and 0.98. The reason, why the experiment with a decrease factor of 0.9 is performing less, is because it gets stuck in a local optimum after a bit more than 1000 iterations. A decrease factor of 0.98 leads to better solutions even after 33,000 iterations indicating that a high decrease factor both increases the quality of diversification and intensification compared to a decreased factor of 0.9.

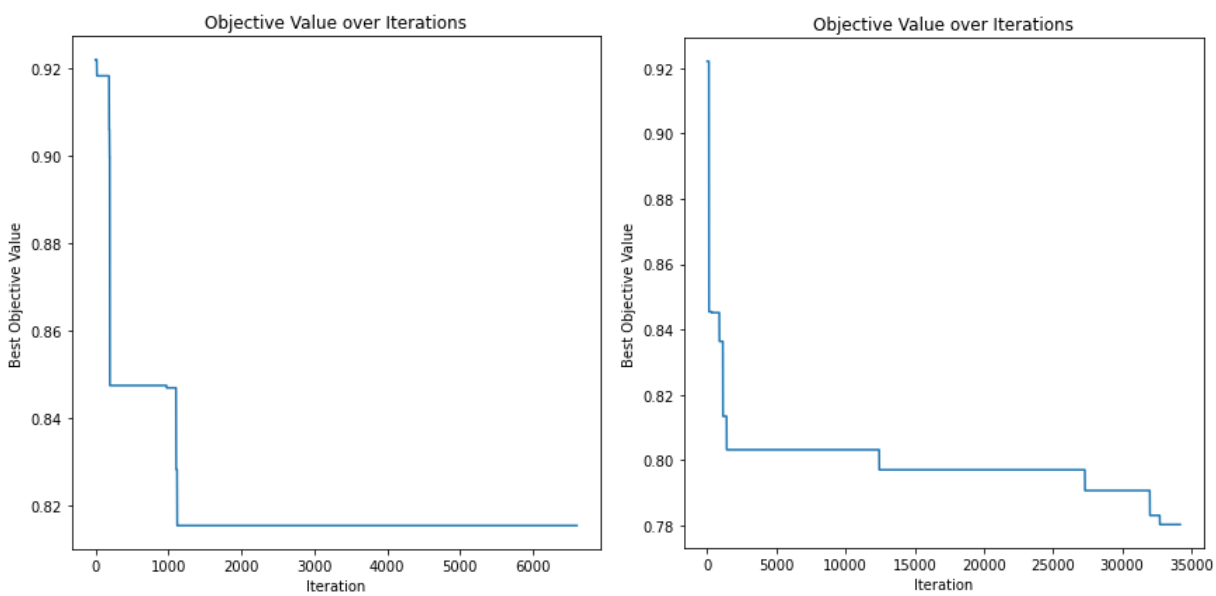


Figure 7.2 - Objective Value over Iterations - DF1 (0.9 - left) vs DF2 (0.98 - right)

While adjustments to the initial temperature and Markov chain length also impact the performance of the algorithm, the decrease factor has the most effect on the quality of the solutions. Considering these findings, further tuning experiments should focus on exploring even higher decrease factors to determine if additional improvements can be achieved.

General Conclusion for Follow-Up Experiments

In general, the simulated annealing heuristic seems to profit from a high degree of diversification at the beginning. High diversification can be done in three different ways: high initial temperature, high decrease factor and a high Markov chain length. For now, the high decrease factor seems to have the best performance, followed by a high Markov chain length. A high initial temperature seems to have the lowest impact. For that reason, follow-up experiments will be done with a higher decrease factor and a higher Markov chain length.

7.4.3 Refined Experiments – Parameter Tuning

This section refines the experiments analysed in Section 7.3 aiming to tune the parameters of the simulated annealing heuristic even more. As concluded in the previous sections, the initial and stopping temperature values are effective and efficient now. However, the Markov chain length and the decrease factor require additional experiments to get an efficient and effective value. This introduces four new experiments shown in Table 7.6.

Experiment	T_0	L	α	T_{Stop}
DF-Refine1	10	100	0.97	0.01
DF-Refine2	10	100	0.99	0.01
ML-Refine1	10	150	0.98	0.01
ML-Refine2	10	250	0.98	0.01

Table 7.6 - Experimental Setup – Parameter Tuning – Refined Experiments

The outcomes of the refined experiments are shown in Table 7.7.

Experiment	Objective Value – Impr. Heuristic	Project Duration – Impr. Heuristic	Utilisation – Impr. Heuristic	Res. Variability – Impr. Heuristic	Running time/Iterations	Number of iterations
Medium Connectivity – DF-Refine1	0.7950	66	82.77%	0.6926	1,029 seconds	22,700
Medium Connectivity – DF-Refine2	0.7803	65	84.04%	0.6741	3,253 seconds	68,800
Medium Connectivity – ML-Refine1	0.7776	65	84.04%	0.6660	2,173 seconds	51,300
Medium Connectivity – ML-Refine2	0.7709	64	85.35%	0.6717	4,040 seconds	85,500

Table 7.7 - Experiments Outcomes – Parameter Tuning – Refined Experiments

Decrease Factor

The experiments with decrease factors of 0.98 and 0.99 resulted in the same schedules. However, the experiment with 0.98 requires 1539 seconds and 34,200 iterations, while the experiment with a decrease factor of 0.99 requires 3,253 seconds and 68,800 iterations. In the experiment with a decrease factor of 0.97, all three parts of the objective value are worse than for higher decrease factors. Although the experiment requires only 1029 seconds and 22,700 iterations, the worse objective value causes a higher decrease factor to be preferred. Since decrease factors of 0.98 and 0.99 give the same schedule, a decrease factor of 0.98 seems to be the best choice because of the lower running time.

Markov Chain Length

The refined experiments for tuning the Markov chain length show that longer Markov chains improve the objective value but increase the running time as well. The experiment with a Markov chain length of 150 has an objective value of 0.7776 obtained in 2,173 seconds in 51,300 iterations. The experiment with a Markov chain length of 250 resulted in the best objective value thus far being 0.7709. However, it does this in 4,040 seconds and 85,500 iterations which is a large increase in computation time. Compared to the initial experiment with a Markov chain length of 200, both refined experiments perform a bit better. However, since all objective values lie close to each other but the running times differ a lot, a Markov chain length of 150 seems to be a good balance between the quality of the objective value and the running time.

After tuning all relevant parameters, the following settings seem to be a good balance between solution quality and running time:

- Initial Temperature ($\mathbf{T_0}$) = 10
- Markov Chain Length (**L**) = 150
- Temperature Decrease Factor ($\boldsymbol{\alpha}$) = 0.98
- Stopping Temperature ($\mathbf{T_{Stop}}$) = 0.01

The graphs with resource utilisation of all parameter tuning experiments are shown in Appendix H. The corresponding charts showing the objective value over iterations and the acceptance ratio versus the temperature are shown in Appendix I.

7.5 Sensitivity Analysis

This section performs several sensitivity analyses to provide deeper insights into the outcomes and performance of the simulated annealing heuristic with different parameter values. The sensitivity analysis is divided into four sections all testing different parameter types. Section 7.5.1 tests what happens when the number of activities differs between experiments. Section 7.5.2 tests different capacities for the resources. Lastly, Section 7.5.3 varies the weights for the different parts of the objective function.

7.5.1 Varying the Number of Activities

The goal of this sensitivity analysis is to assess the impact of increasing the number of activities on the performance of the simulated annealing heuristic. Since it is challenging to apply the same degree of connectivity to scenarios with different numbers of activities, this sensitivity analysis will be applied to the No Connectivity scenario. The sensitivity analyses use the parameter values from the parameter tuning from Section 7.4. This section evaluates the scalability and impact on the output when varying the number of activities. The number of activities to be tested are 10, 20, 40, 60, 80, and 100.

The data sets look like the following:

- 10 Activities: First ten activities of No Connectivity problem instance with 40 activities
- 20 Activities: First twenty activities of No Connectivity problem instance with 40 activities
- 60 Activities: Fourty activities of No Connectivity problem instance with 40 activities and the first 20 activities twice
- 80 Activities: Twice the activities of No Connectivity problem instance with 40 activities.
- 100 Activities: Twice the activities of No Connectivity problem instance with 40 activities and the first twenty activities three times.

Table 7.8 shows the outcomes of the Experiments.

Experiment	Objective Value – Constr. Heuristic	Objective Value – Impr. Heuristic	Project Duration – Constr. Heuristic	Project Duration – Impr. Heuristic	Utilisation – Constr. Heuristic	Utilisation – Impr. Heuristic	Res. Varia-bility – Constr. Heuristic	Res. Varia-bility – Impr. Heuristic	Run time[11]
No connectivity – 10 Activities	0.8924	0.8838	17	17	78.68%	78.68%	0.6773	0.6514	156 sec
No connectivity – 20 Activities	0.8573	0.8238	31	30	83.06%	85.83%	0.5718	0.5358	275 sec
No connectivity – 40 Activities	0.9078	0.8085	71	63	76.94%	86.71%	0.7233	0.6509	425 sec
No connectivity – 60 Activities	0.9078	0.8212	71	64	76.94%	85.35%	0.7233	0.6607	761 sec
No connectivity – 80 Activities	0.8906	0.8869	128	128	85.35%	85.35%	0.6717	0.6607	1,693 sec
No connectivity – 100 Activities	0.8933	0.8714	164	160	82.32%	84.38%	0.6801	0.6631	2,494 sec

Table 7.8 - Experiments Outcomes – Sensitivity Analysis – Varying the Number of Activities

Based on the different experiments, several observations can be made. The first observation is that increasing the number of activities leads to a large increase in running time. While having twenty activities only requires 275 seconds to perform the simulated annealing heuristic, it requires 2,494 seconds with 100 activities. Although the number of activities is five times as high, the running time is almost ten times as high. When applying the number of activities to the highly connected problem instances, the running time would probably increase even more and go to impractical running times. The High Connectivity instance already requires with 40 activities 8-9 times as much time as the No Connectivity problem instance. This indicates the heuristic has some scalability issues.

Since the dataset differs because of the different number of activities, it is challenging to compare the objective values of the different experiments. For the No Connectivity problem instance where there are no precedence relations between the activities, in the experiments with 10, 20, 80, and 100 activities, the constructive heuristic is performing well. This suggests that in scenarios with few dependencies, the heuristic can quickly find good solutions even for a high number of activities.

Since the No Connectivity scenario has no dependencies between activities, all experiments end up in an optimum relatively quickly. All experiments run for 51,300 iterations, but no experiment finds a new best objective value after 21,000 iterations. Four of the six experiments do not find better solutions after 5,000 iterations. The reason for this is probably that all possible solutions have already been generated before

[11] The number of Iterations all equal 51,300. Since this is equal, it is not shown in the Table

because of the relatively small complexity of the data. Thus, the objective values found seem to be global optima.

To conclude, this sensitivity analysis indicates that the simulated annealing heuristic is scalable but requires more running time with an increasing number of activities. The results also show that the heuristic reaches a global optimum quickly for No Connectivity scenarios. This shows that the simulated annealing heuristic in this research is efficient for low-problem instances. However, previous sections have shown that this does not hold for more connected environments. Scenarios with low connectivity can probably benefit from another cooling scheme decreasing the running time.

The graphs with resource utilisation of these experiments can be found in Appendix J. The charts with the objective value over iterations and the acceptance ratio versus the temperature are shown in Appendix K.

7.5.2 Varying Employee Capacity

This section evaluates how different capacities for the resource type "Employees" affect the performance of the simulated annealing heuristic. To ensure accurate tests, the other resource type's capacities are kept constant at 1.

In all previous experiments, the capacity for the "Employees" resource type has been kept constant at 6. This analysis varies the employee capacity with 4, 5, 6, and 7 employees available. The experiments are performed on the Medium Connectivity problem instance with 40 activities. The results of the experiments are shown in Table 7.9.

	Objective Value – Constr. Heuristic	Objective Value – Impr. Heuristic	Project Duration – Constr. Heuristic	Project Duration – Impr. Heuristic	Utilisation – Constr. Heuristic	Utilisation – Impr. Heuristic	Res. Varia-bility – Constr. Heuristic	Res. Varia-bility – Impr. Heuristic	Run time
Employee Capacity: 4	0.9236	0.7956	117	99	62.25%	73.57%	0.7707	0.6946	3,966 sec
Employee Capacity: 5	0.9159	0.8165	90	79	69.37%	79.02%	0.7478	0.6938	2,798 sec
Employee Capacity: 6	0.9221	0.7776	78	65	70.03%	84.04%	0.7663	0.6660	2,173 seconds
Employee Capacity: 7	0.9119	0.8061	64	56	75.86%	86.71%	0.7358	0.6683	1,934 sec

Table 7.9 - Experiments Outcomes – Sensitivity Analysis – Varying Employee Capacity

Based on the experiments, several observations can be made. As expected, increasing the capacity decreases the project duration. The utilisation increases as well which is an interesting observation. There could be two reasons for this increased utilisation. The first reason is that having one idle employee has less impact on the total utilisation when the capacity is higher. For example, one idle employee out of four is 25% and one idle employee out of seven is less than 15%. The second reason is that the utilisation of other resources increases because the same amount has been used but in a shorter time frame. So in case of having higher employee capacity, the other resources have less idle time.

Regarding the objective values, it is not fair to draw valid conclusions because the experiments are directly comparable due to the different capacities used.

When employee capacity increases, there is higher flexibility in assigning activities to different employees. The increase in flexibility leads to lower resource variability for higher capacities.

When capacity is higher, the running time decreases a lot. This is because more flexibility allows for faster solution generation. When an activity is selected to be inserted, more flexibility allows for quicker adjustments to make the schedule feasible again due to fewer resource conflicts. Moreover, the project duration is shorter so the heuristic has to check fewer days for potential resource conflicts.

To conclude, this sensitivity analysis indicates that increasing employee capacity improves project efficiency by reducing the project duration and increasing the utilisation. Additionally, resource variability decreases. However, the impact of adding additional capacity diminishes after a certain point. More can be gained from increasing the capacity from 4 to 5 than from 6 to 7. At a certain point, utilisation may decrease because other resources become the bottleneck and adding more employees would only cause the utilisation to decrease.

The resource utilisation graphs of the experiments from this sensitivity analysis can be found in Appendix J. The objective value over iterations and the acceptance ratio versus the temperature charts are shown in Appendix K.

7.5.3 Varying Weights

This sensitivity analysis aims to assess the impact of differing weights in the objective function. As explained in Section 4.1, the objective function consists of three parts: project duration, resource utilisation, and resource variability. In all previous experiments, all parts had equal weights of 1/3 contributing equally to the objective value. This sensitivity analysis varies the weights to analyse their impact on the solution. All experiments are applied to the Medium Connectivity problem instance. The first experiment is taken over from section 7.5.2 since all weights being 1/3 have already been tested there. Experiments 2,3, and 4 all have one weight equal to one and the other weights being 0. The results of the experiments are shown in Table 7.10.

	Objective Value – Constr. Heuristic	**Objective Value – Impr. Heuristic**	**Project Duration – Constr. Heuristic**	**Project Duration – Impr. Heuristic**	**Utilisation – Constr. Heuristic**	**Utilisation – Impr. Heuristic**	**Res. Variability – Constr. Heuristic**	**Res. Variability – Impr. Heuristic**	**Running time**
W1 = 1/3 W2 = 1/3 W3 = 1/3	0.9221	0.7776	78	65	70.03%	84.04%	0.7663	0.6660	2,173 seconds
W1 = 1 W2 = 0 W3 = 0	1.0	0.8333	78	65	70.03%	84.04%	0.7663	0.6703	2425 sec
W1 = 0 W2 = 1 W3 = 0	1.0	0.8462	78	66	70.03%	82.77%	0.7663	0.6843	2343 sec
W1 = 0 W2 = 0 W3 = 1	0.7663	0.6682	78	65	70.03%	84.04%	0.7663	0.6682	2489 sec

Table 7.10 - Experiments Outcomes – Sensitivity Analysis – Varying Weights

The objective values are not directly comparable because they consist of different elements. However, the elements themselves are comparable. Comparing the initial experiment with all weights equal to 1/3 with the experiment focused on optimising the project duration shows that the project duration and utilisation are the same. However, by also optimising the resource variability, the first experiment gives lower resource variability. This suggests that it is beneficial to give nonzero weights to all experiments since all values are the same or better in that case.

Comparing the first experiment to the experiment focused on optimising the utilisation, the experiment optimising the utilisation performs slightly worse in terms of resource variability and project duration. However, it seems to be the case that the third experiment ended up in a local optimum making it difficult to draw valid conclusions.

Comparing the first experiment to the fourth experiment aiming to optimise resource variability, the project duration and utilisation are consistent across both experiments. However, the resource variability is slightly better in the first experiment. This indicates that assigning a weight of only 1/3 to the resource variability term already has enough impact on the resource variability.

To conclude, all experiments have comparable outcomes. This indicates there is a high correlation between the different terms. This causes optimising one term to have positive effects on the values of the other terms.

7.6 Conclusion

This chapter executed multiple experiments with the simulated annealing part of the MORC CPM aiming to demonstrate its effectiveness in enhancing scheduling efficiency and reliability. First, four problem instances were created. Next, initial settings for the simulated annealing parameters have been determined and are systematically tuned afterwards. After that, sensitivity analyses have been performed on varying the number of activities, employee capacity, and weights in the objective function.

At the beginning of this chapter, four different problem instances have been defined with varying levels of connectivity. This allowed the experiments to show that the more the activities are connected, the more the solution space is restricted causing lower objective values.

After defining the problem instances, initial settings for the simulated annealing heuristic have been generated which are tuned thereafter such that they balance the solution quality and the running time. Important insights from these sections include:

- A relatively high decrease factor of 0.98 improved solution quality with the cost of increased running time but is worth the additional running time.
- A Markov chain length of 150 is a good balance between running time and solution quality.
- An initial temperature of 10 and a stopping temperature of 0.01 give good performance.

After tuning the parameters, three different sensitivity analyses have been performed.

- The first sensitivity analysis tests data sets with different numbers of activities. The main finding here is that the running time increases exponentially when increasing the number of activities. This gives issues with scalability when combining it with highly connected problem instances.
- The second sensitivity analysis varied the employee capacity. Increasing the employee capacity reduced project duration and improved resource utilisation. However, after a certain point, adding

more employees makes the other resources the limiting resource. This showed that adding more employees has a negative effect on utilisation.
- The third sensitivity analysis tested all parts of the objective function. The high correlation between the terms indicated that optimising one term positively influenced the other terms.

The experiments collectively demonstrate that the MORC CPM is efficient and reliable. The method is reliable since it consistently produces high-quality schedules and maintains stability across various scenarios. The method maintained stability in performance metrics and showed that it can handle a large variety of precedence relations. The findings suggest that the MORC CPM can improve scheduling efficiency and reliability, making it a valuable tool for project management.

8 Experimental Analysis – Schedule Revision Method

This chapter evaluates the effectiveness of the Schedule Revision Method (SRM) developed in Chapter 5. Section 8.1 outlines the experimental design. Section 8.2 introduces the problem instances for testing the method. Section 8.3 executes the experiments. Section 8.4 discusses the results in detail.

8.1 Experimental Design

This section outlines the experimental design for evaluating the effectiveness of the SRM. The primary objective of the SRM is to minimise the impact of the disruptions while minimising the changes to the current schedule. The experiments aim to demonstrate that the SRM is more effective than the manual approaches currently used by executors. Figure 8.1 summarises the experimental steps and their goals.

	Experimental Design Steps	**Goal**
1	Define Problem Instances	Create different scenarios to show the ability of the method to revise different disruptions
2	Disruption Scenarios	Create various realistic disruption scenarios to do experiments with.
3	Experiments	Apply the SRM to the disruptions created in Step 2 on the problem instances from Step 1
4	Results and Discussion	Analyse the results of the experiments based on the primary objectives

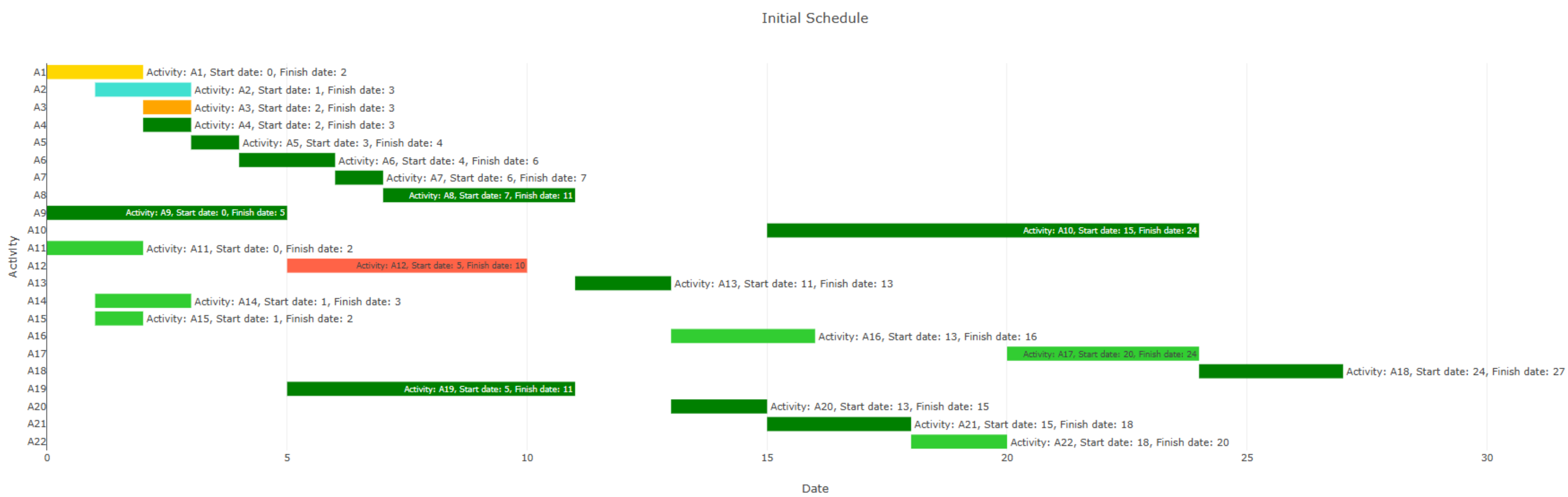
Table 8.1 - Experimental Design - Schedule Revision Method

Step 1 creates the problem instances to test the SRM. Step 2 designs realistic disruption scenarios. Step 3 applies the disruption scenarios from Step 2 to the problem instances of Step 1. Lastly, Step 4 discusses the results based on the primary objectives of the SRM. These primary objectives include the number of rescheduled activities, project duration, resource utilisation, and resource variability.

8.2 Problem Instances

This section introduces the problem instances for evaluating the effectiveness of the SRM. The problem instances are derived from a real-case scenario provided by an executor from Hegeman representing typical disruptions in construction projects [12]. Figure 8.1 shows the current project schedule. The initial project duration is 27 days. Figure 8.2 shows the corresponding resource utilisation graphs. The Excel sheets with the input information are provided in Appendix L.

[12] The problem instances are derived from a real-case scenario, using only selected parts to maintain clarity and manageability in the chapter. As explained in Section 2.4, the operational schedule is divided into different parts corresponding to locations at the construction site. The problem instances use two of these parts with some small modifications.



Legend: Subcontractors

- Hegeman
- Sub E
- Sub C
- Sub A
- Sub D
- Sub B

Figure 8.1 - Current Operational Schedule

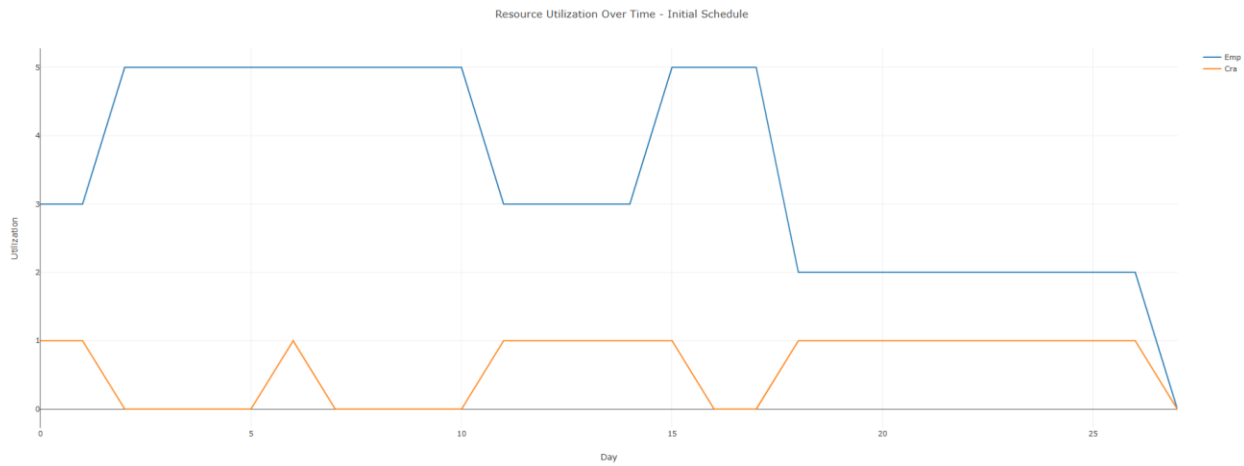


Figure 8.2 - Current Project Schedule - Resource Utilisation

To test the SRM, several realistic disruption scenarios are created, as detailed by the executor. These represent potential issues that could occur during the project impacting the schedule. The disruption scenarios are the problem instances which are shown in Table 8.2.

	Impacted Activity	Disruption Explanation	Day of disruption	Impact
1	A7	Defect Prefab stairs necessitating reproduction. This results in a delay of 8 days.	Day 6	8 Days
2	A2	Adjusting and finishing the HSB walls on the second floor takes two days longer than expected The subcontractor had a double schedule causing they start 2 days later	Day 5	2 Days

Table 8.2 - Problem Instances – Schedule Revision Method

These disruptions are designed to evaluate the ability of the SRM to adapt the schedule while minimising the impact on the primary objectives. Each disruption represents a common issue in construction projects.

8.3 Experiments

This section presents the experiments aiming to evaluate the effectiveness of the SRM and discusses the results. When using the SRM, it is crucial to carefully input the necessary information. This information includes whether rescheduling is allowed, whether an activity has already started, the number of days required to complete the activity, and the initial start and finish times. The schedule should only be revised from the day of the disruption. This means that completed activities should not be considered and activities already started may not be interrupted. Only activities that have not yet started should be considered for rescheduling.

In these experiments, all activities are allowed to be rescheduled for two reasons. The first reason is to show that the method can find better solutions than simply delaying the whole schedule. The second reason is to show that only a few activities are rescheduled even when all activities are flexible.

Problem Instance 1

In Problem Instance 1, the disrupted activity is Activity A7 which is delayed by 8 days (from day 6 to day 14). Although all activities were allowed to be rescheduled, the experiment with Problem Instance 1 only rescheduled Activity A16. Activity A16 was delayed by two days because delaying Activity A7 caused a crane capacity violation on day 14. The delay did not affect the project duration showing the ability of the method to revise schedules effectively. Figure 8.3 shows the revised schedule which can be compared to the schedule in Figure 8.1.

Since the delay of Activity A7 did not affect the project duration, it did not affect the total utilisation. The same number of employees are scheduled within the same timeframe. The resource utilisation per day is shown in Figure 8.4.

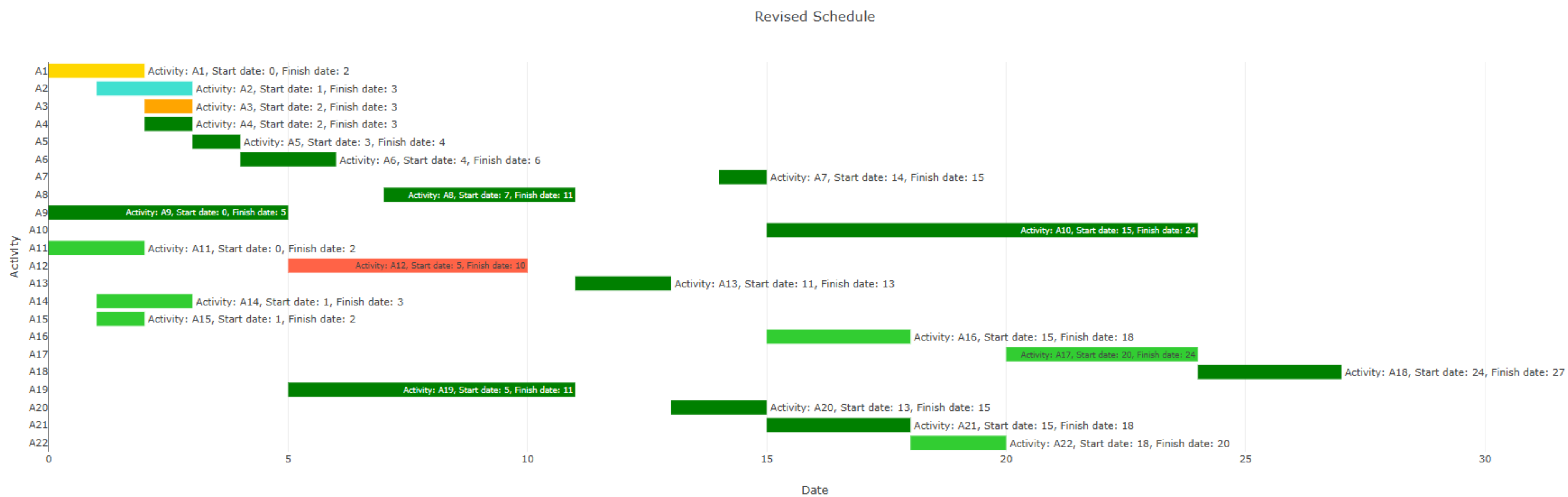


Figure 8.3 - Revised Operational Schedule - Problem Instance 1

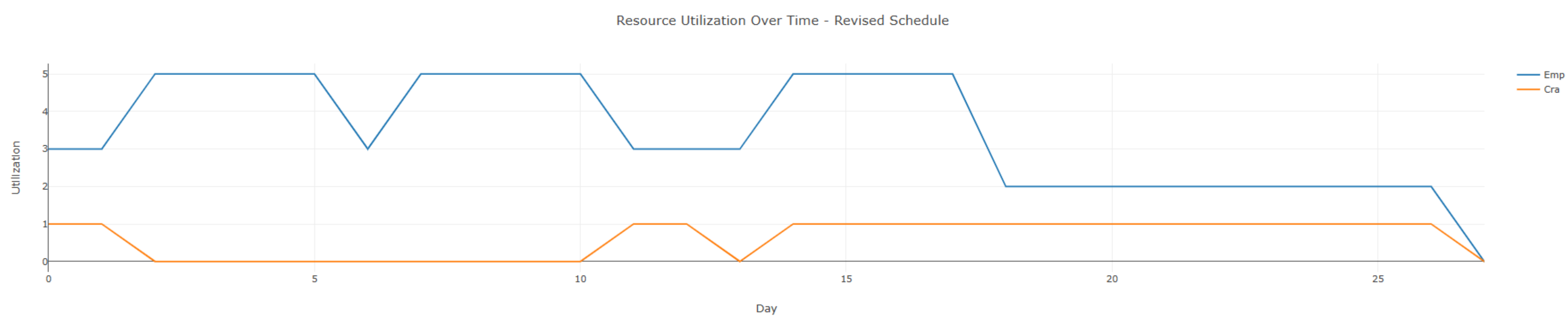


Figure 8.4 - Resource Utilisation - Problem Instance 1

Comparing the outcomes to a manual approach of delaying the entire schedule by 8 days, the SRM saved 8 days of project duration.

Problem Instance 2

In Problem Instance 2, Activity A2 was delayed by two days. Activity A2 was initially scheduled at day 1[13]. This is early in the scheduling period causing that delaying the activity may require a lot of rescheduling. Activity A2 has a duration of two days and has been delayed from day 1 to day 3.

When an executor decides to delay the whole schedule, the project would have been delayed by two days. However, the SRM only gives a project delay of one day. Activity A2 has four (in)direct predecessors (A5, A6, A7, and A8) impacted by this delay since they were initially scheduled immediately after their predecessors were finished. Figure 8.5 shows the revised schedule and the initial schedule is in Figure 8.1.

Figure 8.6 shows the corresponding utilisation per day. Comparing this to the initial resource utilisation from Figure 8.2, the resource utilisation is lower shortly after the disruption. This is due to the delay of the activities to later in the period.

[13] Day 1 is the second day since there is also an Day 0.

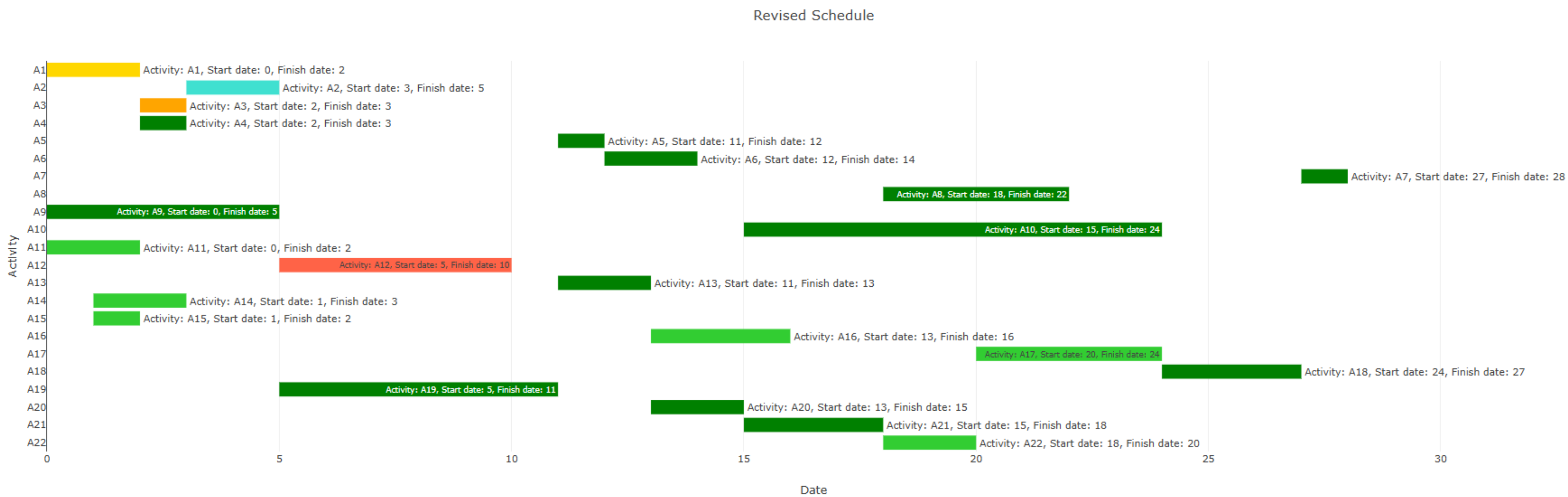


Figure 8.5 - Revised Operational Schedule - Problem Instance 2

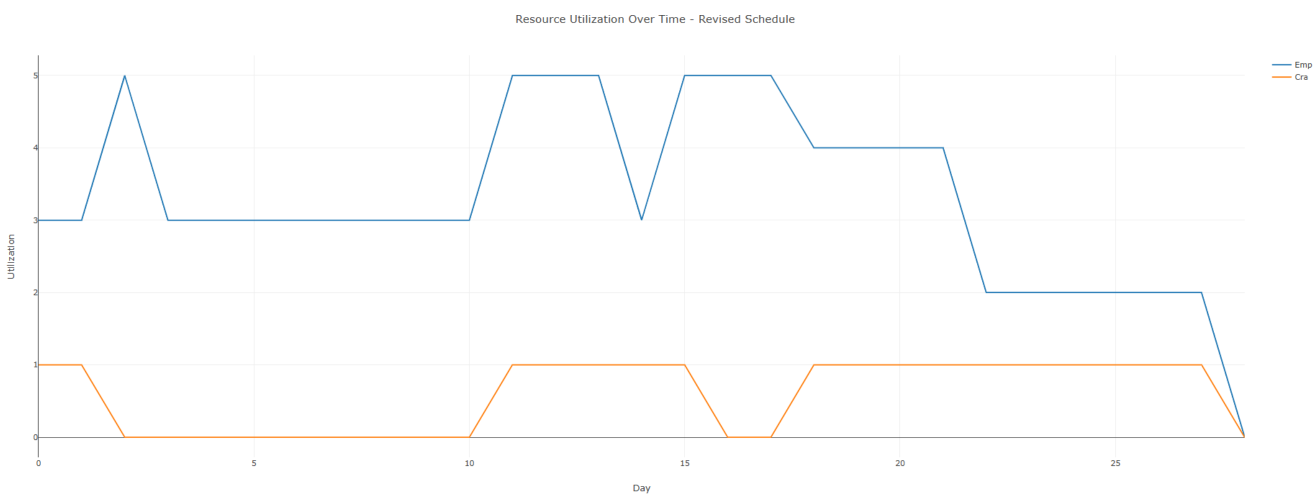


Figure 8.6 - Resource Utilisation - Problem Instance 2

This experiment shows that the method is also able to deal with activities initially scheduled early in the scheduling period.

8.4 Discussion and Conclusion

The experiments showed that the SRM is an efficient and reliable method for revising operational schedules. The SRM consistently provides better results than delaying the whole schedule. Additionally, the experiments have shown that only a fraction of activities need to be rescheduled. These activities mainly include the activities that cause precedence violations or resource violations.

The efficiency of the results of the SRM is highlighted by its ability to minimise project delays even when disruptions occur early in the schedule. The reliability of the SRM is demonstrated by consistently finding good solutions for different problem instances.

A limitation of this chapter is that it is difficult to test the method to other practical revision methods than delaying all activities. This is because most executors work ad hoc and have no detailed strategy for this. It namely depends on the nature of the disruption and what they do with it.

This chapter has shown that the SRM is an effective method capable of revising operational schedules with minimal impact and rescheduling only a few activities. The SRM can assist executors in their way of working by saving time and coming up with efficient rescheduling options.

9 Conclusions and Recommendations

This chapter draws conclusions and provides recommendations based on this research. Section 9.1 answers the main research question and discusses the main findings. Section 9.2 provides recommendations for CAPE Groep, Hegeman, and ECOLOGIC. Lastly, Section 9.3 discusses the limitations of the research and provides directions for future research.

9.1 Conclusion

This research aimed to tackle the core problem "Executors lack generic methods for creating and revising operational schedules in utility construction projects, negatively influencing the efficiency and reliability of project delivery". Addressing this core problem should positively impact the action problem "Frequent occurrence of delayed delivery dates". This section answers the main research question:

Main RQ: *How should the methods for creating and revising operational schedules be designed such that they are practically applicable to the users?*

Currently, executors create and revise operational schedules using their own way of working. A project leader provides an overall plan, and the executor creates operational schedules focusing on the short term. Creating an efficient operational schedule positively impacts project progression.

Schedule Creation Method

For creating operational schedules, this research developed the Multi-Objective Resource-Constrained Critical Path Method (MORC CPM). The MORC CPM balances project duration, resource utilisation, and resource variability. The MORC CPM consists of five phases that develop efficient, accurate and adaptable schedules:

1. **Phase 1**: Creates an unfeasible schedule without capacity constraints, serving as a foundation for subsequent phases.
2. **Phase 2**: Applies a constructive heuristic with a resource-constrained forward pass, developed in this research, to make the schedule feasible
3. **Phase 3**: Uses a simulated annealing heuristic to improve the schedule by using insertion and smart insertion operators.
4. **Phase 4**: Identifies resource links and the critical sequence. Using the resource links, this phase derives personnel and crane schedules.
5. **Phase 5**: Assigns activities to idle employees on days with unoccupied employees. These employees can do some (preparing) work for activities not having all resources available. The executor benefits from this in that there are no idle employees and it increases the chance the activity is finished on time.

To assist Hegeman, a tool has been developed with the MORC CPM integrated into it. This tool allows for experiments with the method. The experimental analysis with Phase 3 of the MORC CPM involved tuning parameters and performing sensitivity analyses to understand the impact of these parameters. Key observations include:

- Parameter tuning balanced both running time and solution quality with the following settings for the simulated annealing:
    - Initial Temperature (T0) = 10
    - Markov Chain Length (L) = 150
    - Temperature Decrease Factor (α) = 0.98
    - Stopping Temperature (TStop) = 0.01
- Sensitivity analyses showed:
    - The running time increases exponentially with the number of activities, indicating scalability issues for large, highly connected problem instances.
    - Increasing the employee capacity reduces project duration and improves resource utilisation. However, beyond a certain point, adding more employees makes other resources the limiting factor, which negatively affects overall utilisation.
    - High correlation between different terms of the objective function means optimising one term positively influences the others. So, when optimising only the resource variability, the project duration decreases, and the resource utilisation increases.
    - Phase 3 of the MORC CPM effectively improves operational schedules in terms of project duration, resource utilisation, and resource variability.

It is important to note that the findings of these experiments apply to the data sets used. This does not necessarily mean that they apply to other data sets.

Schedule Revision Method

In construction projects, disruptions are common which requires rescheduling. Currently, executors at Hegeman revise operational schedules based on their experience, with no standardised methodology. This research developed a Schedule Revision Method (SRM) to assist executors in handling disruptions with minimal changes and impact on project duration.

The SRM focuses on rescheduling activities after a disruption occurs to make the schedule feasible again. The SRM aims to reschedule activities allowed to be rescheduled with minimal changes and impact on the project duration. Key observations from the SRM experiments include:

- The SRM has shown its ability to adapt schedules to disruptions by selectively rescheduling to make the schedule feasible again. This minimises the impact on the operational schedules and the project duration demonstrating its efficiency.
- The SRM has shown that it performed well across different scenarios, demonstrating its reliability.

Executors at Hegeman will benefit from these structured methods and tools provide guidance in their work of creating and revising operational schedules. These methods enhance efficiency and reliability and therefore have a positive impact on the project progression.

9.2 Recommendations

This section provides Hegeman, CAPE Groep, and ECOLOGIC with recommendations based on this research. The recommendations for Hegeman are:

- **Test and implement the Schedule Creation Tool**: Test the Schedule Creation Tool with the MORC CPM on a single project to evaluate its performance and identify any limitations. If successful, incorporate the tool into the standard way of working for all executors. If limitations are identified, iteratively improve the tool.
- **Test and implement the Schedule Revision Tool:** Test the Schedule Revision Tool with the SRM for various disruptions on a single project and assess its effectiveness and reliability. If it works as intended, use the tool for more projects and integrate in the standard way of working for all executors. If problems occur, try to iteratively solve the problems to improve the tool.
- **Research integration of the tools**: Research the integration of the scheduling methods and tools company-wide in Hegeman and identify all potential problems for implementation.
- **Provide training for the executors**: Provide training programs for executors on using the methods and tools.
- **Continuously improve the methods using a feedback mechanism:** Establish a feedback mechanism to collect feedback, experiences and suggestions from the executors. Use this to continuously refine and improve the methods and tools.
- **Apply simulated annealing settings**: Implement the optimised settings for the simulated annealing heuristic, and test the effectiveness in real-world situations.

The recommendation for CAPE Groep is to **consider building a generic, industry-wide tool** with the methods and tools from this research as a basis.

The recommendations for ECOLOGIC are:

- **Digital Twin integration**: Research the implementation of these scheduling methods in the digital twin to be developed.
- **Research improving scheduling methods**: Conduct research on improving the scheduling methods.
- **Develop tools for creating overall plans**: Do further research on whether comparable methods can be developed for creating an overall plan.

9.3 Limitations and Future Research

This section outlines the limitations of the research and identifies possibilities for future research.

The first limitation is that the MORC CPM has been tested on test data instead of real data. This test data may not fully represent all possible real problem instances. Future research can be done on how these methods perform with real-world data. Moreover, future research can investigate how the settings of the simulated annealing heuristic react to different data.

The second limitation is that the methods and tools rely on several assumptions and simplifications. These assumptions and simplifications are valid but represent a simplified version of reality. However, the assumptions and simplifications are designed such that the methods still give valid results. Future research can be done on making the method more realistic by having fewer assumptions and simplifications.

Another limitation is that the methods and tools developed in this research have not been tested in real-world environments. So, it is unknown how executors react to using these methods and tools and whether they have the intended outcomes.

While uncertainty factors have been introduced in the constructive heuristic, the improvement heuristic does not take into account these uncertainty factors. The uncertainty factors have some impact on the final schedule since the constructive schedule influences the final schedule but this impact is limited. Future research can be done on how to integrate these uncertainty factors in the simulated annealing heuristic.

The SRM developed in this research can handle only one disruption at the time. Further research and development can investigate whether it is possible to improve the SRM such that it can handle multiple disruptions.

Since creating an overall plan and an operational schedule have things in common, further research can explore how to adapt the MORC CPM for developing overall plans.

From the ECOLOGIC perspective, future research can be done on how to implement these methods into a digital twin. On top of that, the introduction of Artificial Intelligence in creating and revising schedules can be researched. Furthermore, ECOLOGIC can research collaboration with stakeholders in creating the operational schedule with the digital twin.

Bibliography

Aarts, E. H. L., Korst, J. H. M., & Laarhoven, P. J. M. van. (1997). Simulated annealing. In *Local search in combinatorial optimization* (Vol. 187, Issue 210, pp. 91–120). Wiley-Interscience.

Abbasi, B., Shadrokh, S., & Arkat, J. (2006). Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied Mathematics and Computation*, *180*(1), 146–152.

Abdolshah, M. (2014). A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*, *5*(4), 253–286.

Abeyasinghe, L., Greenwood, D. J., & Johansen, D. E. (2001). An efficient method for scheduling construction projects with resource constraints. *International Journal of Project Management*, *19*(1), 29–45. www.elsevier.com/locate/ijproman

Aghimien, D., Aigbavboa, C., Oke, A., & Setati, M. (2018). Challenges of Front-End Loading In Construction Project Delivery. In *Proceedings of the Fourth Australasia and South-East Asia Structural Engineering and Construction Conference*.

Al Nasseri, H., & Aulin, R. (2016). Understanding management roles and organisational behaviours in planning and scheduling based on construction projects in Oman. *Journal of Construction in Developing Countries*, *21*(1), 1–18.

Al-Emad, N., Rahman, I. A., Nagapan, S., & Gamil, Y. (2017). Ranking of Delay Factors for Makkah's Construction Industry. In *MATEC Web of Conferences* (Vol. 103, p. 03001). EDP Sciences.

Alfieri, A., Tolio, T., & Urgo, M. (2011). A project scheduling approach to production planning with feeding precedence relations. *International Journal of Production Research*, *49*(4), 995–1020.

Atin, S., & Lubis, R. (2019). Implementation of Critical Path Method in Project Planning and Scheduling. *IOP Conference Series: Materials Science and Engineering (Vol. 662, No.2, p. 022031)*, *662*(2).

Baydoun, G., Haït, A., Pellerin, R., Clément, B., & Bouvignies, G. (2016). A rough-cut capacity planning model with overlapping. *OR Spectrum*, *38*(2), 335–364.

Bishnoi, N. (2018). Critical Path Method (CPM): A Coordinating Tool. *International Research Journal of Management Science & Technology*, *9*(1), 459–467. http://www.irjmst.com

Bokor, O., Kocsis, T., & Szenik, G. (2011). New tools in project scheduling. Challenges of the construction project planning. *Budownictwo Inzynieria Srodowiska*, *2*(3), 214–221. http://en.wikipedia.org

Bowers, J. A. (1995). Criticality in Resource Constrained Networks. *Source: The Journal of the Operational Research Society*, *46*(1), 80–91. https://www.jstor.org/stable/2583838

Choi, J. (2012). A Knowledge Representation Model for Formalizing Subcontractors' Scheduling Process. *Journal of Asian Architecture and Building Engineering*, *11*(2), 275–282.

Cynthia, O. U. (2020). Implementation of Project Evaluation and Review Technique (PERT) and Critical Path Method (CPM): A Comparative Study. *International Journal of Industrial and Operations Research*, *3*(004).

De Boer, R. (1998). *Resource-constrained multi-project management : a hierarchical decision support system*. University of Twente.

Elsayegh, A., & El-Adaway, I. H. (2021). Holistic Study and Analysis of Factors Affecting Collaborative Planning in Construction. *Journal of Construction Engineering and Management*, *147*(4).

Ghaffari, M., & Emsley, M. W. (2015). Current status and future potential of the research on Critical Chain Project Management. *Surveys in Operations Research and Management Science*, *20*(2), 43–54.

Han, X., Dong, Y., Yue, L., & Xu, Q. (2019). State Transition Simulated Annealing Algorithm for Discrete-Continuous Optimization Problems. *IEEE Access*, *7*, 44391–44403. https://doi.org/10.1109/ACCESS.2019.2908961

Hans, E. W., Herroelen, W., Leus, R., & Wullink, G. (2007). A hierarchical approach to multi-project planning under uncertainty. *Omega*, *35*(5), 563–577.

Hartmann, S., & Briskorn, D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. In *European Journal of Operational Research* (Vol. 297, Issue 1, pp. 1–14). Elsevier B.V.

Heerkens, H., & Van Winden, A. (2017). *Solving Managerial Problems Systematically* (1st ed.). Noordhoff Uitgevers.

Henderson, D., Jacobson, S. H., & Johnson, A. W. (2006). The Theory and Practice of Simulated Annealing. In *Handbook of Metaheuristics* (pp. 287–319). Kluwer Academic Publishers.

Kim, K. (2020). Generalized resource-constrained critical path method to improve sustainability in construction project scheduling. *Sustainability*, *12*(21).

Kim, K., & de la Garza, J. M. (2003). Phantom Float. *Journal of Construction Engineering and Management*, *129*(5), 507–517.

Kim, K., & de la Garza, J. M. (2005). Evaluation of the Resource-Constrained Critical Path Method Algorithms. *Journal of Construction Engineering and Management*, *131*(5), 522–532.

Kim, N., Moon, H., Son, B., & Park, M. (2021). Smart and robust critical chain scheduling for construction projects using buffer extraction. *International Journal of Sustainable Building Technology and Urban Development*, *12*(4), 394–409.

König, M., & Beißert, U. (2009). Construction Scheduling Optimization by Simulated Annealing. *Proc. of the 26th Annual International Symposium on Automation and Robotics in Construction*.

Kreis, D., Gibson, B., Jasper, J., Van Dyke, C., Wallace, C., Catchings, R., & McCormack, S. (2019). *Critical Path for Project Development*.

Leach, L. P. (1999). Critical Chain Project Management Improves Project Performance. *Project Management Journal*, *30*(2), 39–51.

Lin, S. W., & Yu, V. F. (2012). A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, *217*(1), 94–107.

Lu, M., Lam, H. C., & Dai, F. (2008). Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization. *Automation in Construction*, *17*(6), 670–681.

Lu, M., & Li, H. (2003). Resource-Activity Critical-Path Method for Construction Planning. *Journal of Construction Engineering and Management*, *129*(4), 412–420.

Majumder, S., Majumder, S., & Biswas, D. (2022). Impact of effective construction planning in project performance improvement. *Quality and Quantity*, *56*(4), 2253–2264.

Naber, A., & Kolisch, R. (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, *239*(2), 335–348.

Nawaz, A., Su, X., & Nasir, I. M. (2021). BIM Adoption and Its Impact on Planning and Scheduling Influencing Mega Plan Projects- (CPEC-) Quantitative Approach. *Complexity*, *2021*, 1–9.

Nkasu, M. M. (1994). COMSARS: a computer-sequencing approach to multiresource-constrained scheduling Part 1: Deterministic networks. *International Journal of Project Management*, *12*(3), 183–192.

Ökmen, Ö. (2013). A procedure for critical path method-based scheduling in linear construction projects. *Journal of the South African Institution of Civil Engineering*, *55*(2), 12–20.

Pantouvakis, J.-P., & Manoliadis, O. G. (2006). A Practical Approach to Resource-Constrained Project Scheduling. *Operations Research*, *6*(3), 299–309.

Rader, D. J. (2010). Simulated Annealing. In *Deterministic operations research: models and methods in linear optimization* (pp. 652–657). John Wiley & Sons.

Saad, M., Baba, S., & Amoudi, O. (2015). A suggested solution to improve the traditional construction planning approach. *Jordan Journal of Civil Engineering*, *9*(2).

Sears, S. K., Sears, G. A., & Clough, R. H. (2008). *Construction Project Management: A Practical Guide to Field Construction Management* (5th ed.). John Wiley & Sons, Inc.

Seppänen, O., & Kenley, R. (2005). Performance Measurement Using Location-based Status Data. *Proceedings of International Group for Lean Construction (IGLC)-13*, 263–269.

Sheikhkhoshkar, M., Bril El-Haouzi, H., Aubry, A., & Hamzeh, F. (2023). Functionality as a Key Concept for Integrated Project Planning and Scheduling Methods. *Journal of Construction Engineering and Management*, *149*(7).

Tang, P., Mukherjee, A., & Onder, N. (2013). Construction schedule simulation for improved project planning: Activity criticality index assessment. In *2013 Winter Simulation Conference (WSC)* (pp. 3237–3248). IEEE.

Tritschler, M., Naber, A., & Kolisch, R. (2017). A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, *262*(1), 262–273.

Van der Horn, E., & Mahadevan, S. (2021). Digital Twin: Generalization, characterization and implementation. *Decision Support Systems*, *145*.

Van Laarhoven, P. J. M., Aarts, E. H. L., & Lenstra, J. K. (1992). Job Shop Scheduling by Simulated Annealing. In *Operations Research: Vol. 40 (1)* (Issue 1, pp. 113–125).

Viklund Tallgren, M., Roupé, M., Johansson, M., & Bosch-Sijtsema, P. (2020). BIM-tool development enhancing collaborative scheduling for pre-construction. *Journal of Information Technology in Construction (ITcon)*, *25*, 374–397.

Winch, G. M., & Kelsey, J. (2005). What do construction project planners do? *International Journal of Project Management*, *23*(2), 141–149.

Woodworth, B. M., & Shanahan, S. (1988). Identifying the critical sequence in a resource constrained project. *International Journal of Project Management*, *6*(2), 89–96.

Yarramsetti, S., & Kousalya, G. (2015). An optimized event based software project scheduling with uncertainty treatment. *ARPN Journal of Engineering and Applied Sciences*, *10*(6), 2613–2620.

Zwikael, O. (2009). Critical Planning Processes in Construction Projects. *Construction Innovation*, *9*(4), 372–387.

Appendix A – Schedule Creation Tool – Excel Input

Activities TAB

	Activity Description	Activity	Days	Subcontractor	Release Date	Due Date	UncertaintyFactor	Delivery?	Number of days call before		
1	Activity A1	A1	11		0	500	4	Nee		Number of Activities:	40
2	Activity A2	A2	8		0	500	1	Nee			
3	Activity A3	A3	3	Sub A	0	500	2	Nee			
4	Activity A4	A4	5		0	500	1	Nee			
5	Activity A5	A5	8	Sub B	0	500	5	Nee			
6	Activity A6	A6	4		0	500	4	Nee			
7	Activity A7	A7	5		0	500	2	Nee			
8	Activity A8	A8	7	Sub C	0	500	5	Nee			
9	Activity A9	A9	9		0	500	2	Nee			
10	Activity A10	A10	1		0	500	3	Nee			
11	Activity A11	A11	9	Sub D	0	500	3	Nee			
12	Activity A12	A12	5	Sub A	0	500	3	Nee			
13	Activity A13	A13	12	Sub D	0	500	1	Nee			
14	Activity A14	A14	4		0	500	2	Nee			
15	Activity A15	A15	8	Sub B	0	500	1	Nee			
16	Activity A16	A16	10		0	500	4	Nee			
17	Activity A17	A17	3	Sub E	0	500	3	Nee			
18	Activity A18	A18	8		0	500	1	Nee			
19	Activity A19	A19	12		0	500	1	Nee			
20	Activity A20	A20	6		0	500	4	Nee			
21	Activity A21	A21	10	Sub B	0	500	1	Nee			
22	Activity A22	A22	4		0	500	5	Nee			
23	Activity A23	A23	3	Sub F	0	500	3	Nee			
24	Activity A24	A24	3		0	500	2	Nee			
25	Activity A25	A25	1	Sub G	0	500	1	Nee			
26	Activity A26	A26	6		0	500	2	Nee			
27	Activity A27	A27	1		0	500	2	Nee			
28	Activity A28	A28	3	Sub F	0	500	1	Nee			
29	Activity A29	A29	7		0	500	1	Nee			
30	Activity A30	A30	10		0	500	2	Nee			
31	Activity A31	A31	7	Sub H	0	500	2	Nee			
32	Activity A32	A32	12		0	500	2	Nee			
33	Activity A33	A33	4	Sub F	0	500	3	Nee			
34	Activity A34	A34	12		0	500	2	Nee			
35	Activity A35	A35	2		0	500	5	Nee			
36	Activity A36	A36	5	Sub H	0	500	2	Nee			
37	Activity A37	A37	8		0	500	5	Nee			
38	Activity A38	A38	2	Sub H	0	500	2	Nee			
39	Activity A39	A39	10		0	500	1	Nee			

Activities – Predecessors TAB

Beschrijving Activity	Activiteit	Voorganger 1	Percentage Voorganger 1	Voorganger 2	Percentage Voorganger 2
Activity A1	A1				
Activity A2	A2				
Activity A3	A3				
Activity A4	A4	A1	100.00%	A2	75.00%
Activity A5	A5	A4	50.00%		
Activity A6	A6	A5	100.00%		
Activity A7	A7				
Activity A8	A8				
Activity A9	A9	A7	100.00%		
Activity A10	A10				
Activity A11	A11	A9	50.00%	A10	75.00%
Activity A12	A12				
Activity A13	A13	A12	100.00%		
Activity A14	A14	A13	50.00%		
Activity A15	A15				
Activity A16	A16	A14	75.00%	A15	50.00%
Activity A17	A17				
Activity A18	A18				
Activity A19	A19				
Activity A20	A20	A18	100.00%	A19	100.00%
Activity A21	A21	A20	75.00%		
Activity A22	A22				
Activity A23	A23				
Activity A24	A24				
Activity A25	A25				
Activity A26	A26	A23	100.00%		

Resources TAB

Resource	Afkorting Resource	Dag 1	Dag 2	Dag 3	Dag 4	Dag 5	Dag 6	Dag 7	Dag 8	Dag 9	Dag 10	Dag 11	Dag 12	Dag 13	Dag 14	Dag 15	Dag 16	Dag 17	Dag 18	Dag 19	Dag 20	Dag 21	Dag 22	Dag 23	Dag 24	Dag 25	Dag 26
Employees	Emp	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Crane	Cra	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Telehandler	Tel	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Resource Assignments TAB

Beschrijving Activity	Activity	Subcontractor	Resource 1	Resource Units 1	Resource 2	Resource Units 2	Resource 3	Resource Units 3	Resource 4	Resource Units 4
Activity A1	A1	0	Emp	1			Tel	1		
Activity A2	A2	0	Emp	4						
Activity A3	A3	Sub A			Cra	1				
Activity A4	A4	0	Emp	3						
Activity A5	A5	Sub B								
Activity A6	A6	0	Emp	1	Cra	1				
Activity A7	A7	0	Emp	3						
Activity A8	A8	Sub C								
Activity A9	A9	0	Emp	1						
Activity A10	A10	0	Emp	2			Tel	1		
Activity A11	A11	Sub D			Cra	1				
Activity A12	A12	Sub A								
Activity A13	A13	Sub D								
Activity A14	A14	0	Emp	2	Cra	1				
Activity A15	A15	Sub B								
Activity A16	A16	0	Emp	1						
Activity A17	A17	Sub E								
Activity A18	A18	0	Emp	1						
Activity A19	A19	0	Emp	4						
Activity A20	A20	0	Emp	1	Cra	1				

Appendix B – Schedule Creation Tool – Dashboard Output

Schedule Output

Appendix C – Schedule Revision Tool – Excel Input

Activities TAB

	Activity Description	Activity	Rescheduling allowed?	Already started?	Days to schedule	Subcontractor	Initial start date	Initial end date	UncertaintyFactor
1	Activity A1	A1	Ja	Nee	11		16	27	4
2	Activity A2	A2	Ja	Ja	0		0	8	1
3	Activity A3	A3	Ja	Ja	0	Sub A	0	3	2
4	Activity A4	A4	Ja	Nee	5		43	48	1
5	Activity A5	A5	Ja	Nee	8	Sub B	46	54	5
6	Activity A6	A6	Ja	Nee	4		46	50	4
7	Activity A7	A7	Ja	Ja	4		8	13	2
8	Activity A8	A8	Ja	Nee	7	Sub C	50	57	5
9	Activity A9	A9	Ja	Nee	9		19	28	2
10	Activity A10	A10	Ja	Nee	1		12	13	3
11	Activity A11	A11	Ja	Nee	9	Sub D	24	33	3
12	Activity A12	A12	Ja	Ja	0	Sub A	0	5	3
13	Activity A13	A13	Ja	Ja	0	Sub D	0	12	1
14	Activity A14	A14	Ja	Nee	4		41	45	2
15	Activity A15	A15	Ja	Ja	0	Sub B	0	8	1
16	Activity A16	A16	Ja	Nee	10		42	52	4
17	Activity A17	A17	Ja	Ja	0	Sub E	0	3	3
18	Activity A18	A18	Ja	Ja	7		8	16	1
19	Activity A19	A19	Ja	Nee	12		13	25	1
20	Activity A20	A20	Ja	Nee	6		13	19	4

Activities – Predecessors TAB

Activity Description	Activity	Predecessor 1	Predecessor Percentage 1	Predecessor 2	Predecessor Percentage 2	Predecessor 3	Predecessor Percentage 3
Activity A1	A1						
Activity A2	A2						
Activity A3	A3						
Activity A4	A4	A1	100.00%	A2	75.00%		
Activity A5	A5	A4	50.00%				
Activity A6	A6						
Activity A7	A7						
Activity A8	A8						
Activity A9	A9	A7	100.00%				
Activity A10	A10						
Activity A11	A11	A9	50.00%	A10	75.00%		
Activity A12	A12						
Activity A13	A13	A12	100.00%				
Activity A14	A14	A13	50.00%				
Activity A15	A15						
Activity A16	A16	A14	75.00%	A15	50.00%		
Activity A17	A17						
Activity A18	A18						
Activity A19	A19						
Activity A20	A20	A18	100.00%	A19	100.00%		

Resources TAB

Resource	Afkorting Resource	Dag 1	Dag 2	Dag 3	Dag 4	Dag 5	Dag 6	Dag 7	Dag 8	Dag 9	Dag 10	Dag 11	Dag 12	Dag 13	Dag 14	Dag 15	Dag 16	Dag 17	Dag 18	Dag 19	Dag 20	Dag 21	Dag 22	Dag 23	Dag 24	Dag 25	Dag 26
Employees	Emp	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Crane	Cra	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Telehandler	Tel	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

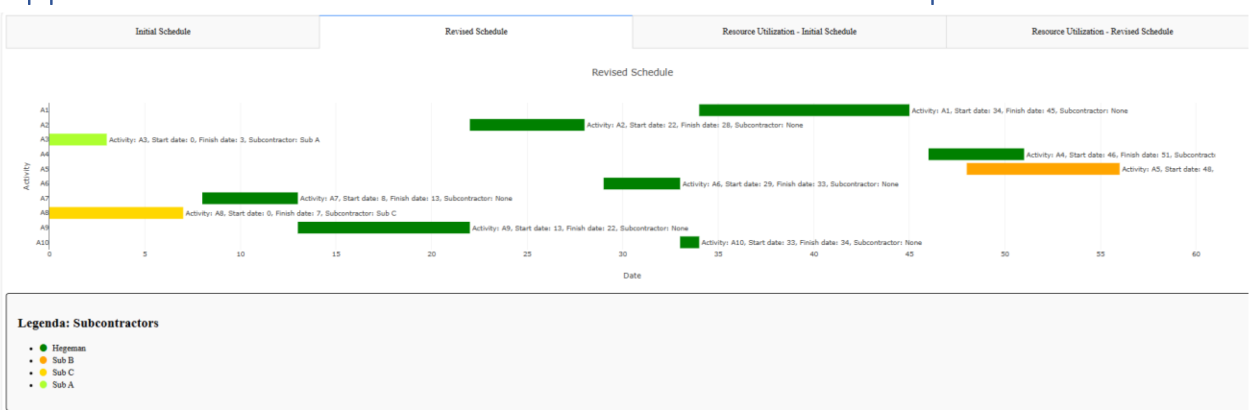
Resource Assignments TAB

Beschrijving Activity	Activity	Subcontractor	Resource 1	Resource Units 1	Resource 2	Resource Units 2	Resource 3	Resource Units 3	Resource 4	Resource Units 4
Activity A1	A1	0	Emp	1			Tel	1		
Activity A2	A2	0	Emp	4						
Activity A3	A3	Sub A			Cra	1				
Activity A4	A4	0	Emp	3						
Activity A5	A5	Sub B								
Activity A6	A6	0	Emp	1	Cra	1				
Activity A7	A7	0	Emp	3						
Activity A8	A8	Sub C								
Activity A9	A9	0	Emp	1						
Activity A10	A10	0	Emp	2			Tel	1		
Activity A11	A11	Sub D			Cra	1				
Activity A12	A12	Sub A								
Activity A13	A13	Sub D								
Activity A14	A14	0	Emp	2	Cra	1				
Activity A15	A15	Sub B								
Activity A16	A16	0	Emp	1						
Activity A17	A17	Sub E								
Activity A18	A18	0	Emp	1						
Activity A19	A19	0	Emp	4						
Activity A20	A20	0	Emp	1	Cra	1				

Disruptions TAB

Activity Name	Disruption Date	Impact Disruption
A2	4	50

Appendix D – Schedule Creation Tool – Dashboard Output



Appendix E – Clusters Problem Instances

Clusters Initial Settings – Low connectivity:

- **Cluster 1**: A7 → A9 → A11
- **Cluster 2:** A12 → A18
- **Cluster 3**: A22 → A26 → A30
- **Cluster 4**: A32 → A33
- **Cluster 5**: A35 → A38 → A39

Clusters Initial Settings – Medium Connectivity:

- **Cluster 1**:
    - A1
    - A2
    - A4 (Predecessors: A1 (100%), A2 (75%))
    - A5 (Predecessor: A4 (50%))
    - A6 (Predecessor: A5 (100%))
- **Cluster 2**:
    - A7
    - A9 (Predecessor: A7 (100%))
    - A10
    - A11 (Predecessors: A9 (50%) , A10 (75%))
- **Cluster 3**:
    - A12
    - A13 (Predecessor: A12 (100%))
    - A14 (Predecessor: A13 (50%))
    - A15
    - A16 (Predecessors: A14 (75%) , A15 (50%))
- **Cluster 4**:
    - A18
    - A19
    - A20 (Predecessors: A18 (100%) , A19 (100%))
    - A21 (Predecessor: A20 (75%))
- **Cluster 5**:
    - A23
    - A26 (Predecessor: A23 (100%))
    - A27 (Predecessor: A26 (100%))
    - A28 (Predecessor: A27 (75%))
    - A29 (Predecessor: A28 (50%))
- **Cluster 6**:
    - A30
    - A32
    - A33 (Predecessors: A30 (50%), A32 (100%))
    - A34 (Predecessor: A33 (100%))
- **Cluster 7**:

- A35
- A37
- A38 (Predecessors: A35 (75%), A37 (100%))
- A39 (Predecessor: A38 (50%))
- A40 (Predecessor: A39 (100%))

Clusters Initial Settings – High Connectivity

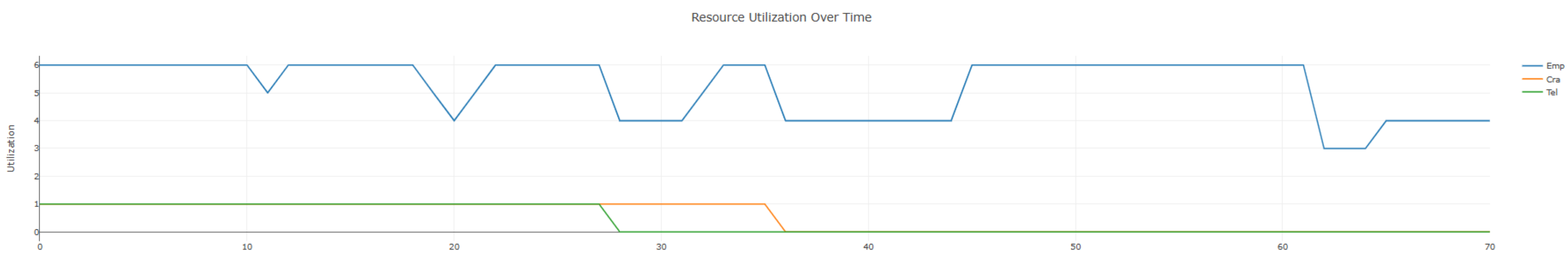
- **Cluster 1:**
  - A1
  - A2
  - A3
  - A4 (Predecessors: A1 (100%), A2 (75%))
  - A5 (Predecessors: A3 (50%), A4 (100%))
  - A6 (Predecessor: A5 (75%))
  - A7
  - A8
  - A9 (Predecessors: A7 (100%), A8 (50%))
  - A10
  - A11 (Predecessors: A9 (100%), A10 (75%))
  - A12
  - A13 (Predecessors: A12 (100%), A5 (50%))
  - A14 (Predecessors: A13 (50%), A6 (100%))
  - A15 (Predecessor: A14 (75%))
  - A16 (Predecessors: A14 (75%), A15 (50%))
  - A17
  - A18 (Predecessors: A16 (100%), A8 (50%))
  - A19 (Predecessors: A18 (75%), A17 (50%))
  - A20 (Predecessors: A18 (100%), A19 (50%), A9 (75%))
- Cluster 2:
  - A21
  - A22
  - A23 (Predecessors: A21 (100%), A22 (75%))
  - A24 (Predecessors: A23 (50%))
  - A25 (Predecessors: A24 (75%))
  - A26 (Predecessor: A23 (100%))
  - A27 (Predecessor: A26 (100%))
  - A28 (Predecessors: A27 (75%), A24 (50%))
  - A29 (Predecessors: A28 (50%), A25 (75%))
  - A30
  - A31
  - A32
  - A33 (Predecessors: A30 (50%), A32 (100%))
  - A34 (Predecessor: A33 (100%))
  - A35 (Predecessors: A29 (75%), A26 (50%))

- A36
- A37 (Predecessors: A35 (100%), A36 (50%))
- A38 (Predecessors: A35 (75%), A37 (100%))
- A39 (Predecessors: A38 (50%), A36 (75%))
- A40 (Predecessors: A39 (100%), A32 (50%))

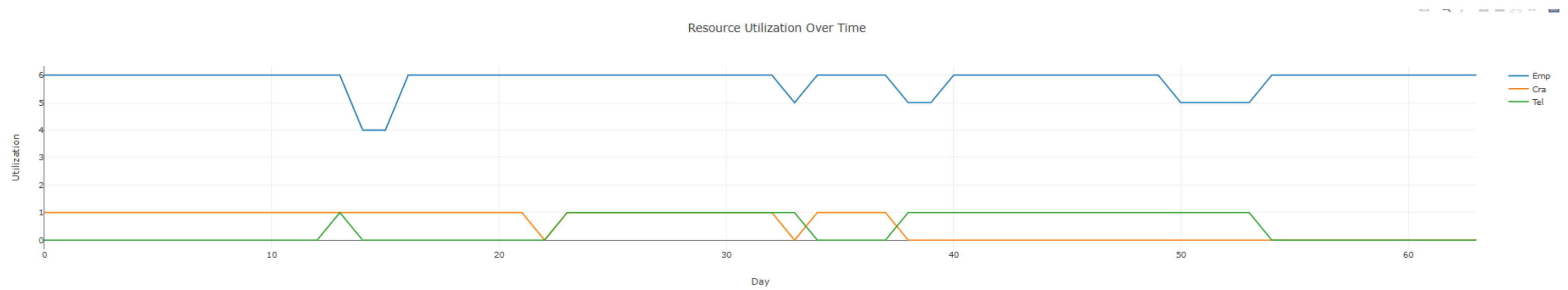
Appendix F – Resource Utilisation – Initial Settings

No connectivity (IT = 10, ML = 100)

Constructive Heuristic:

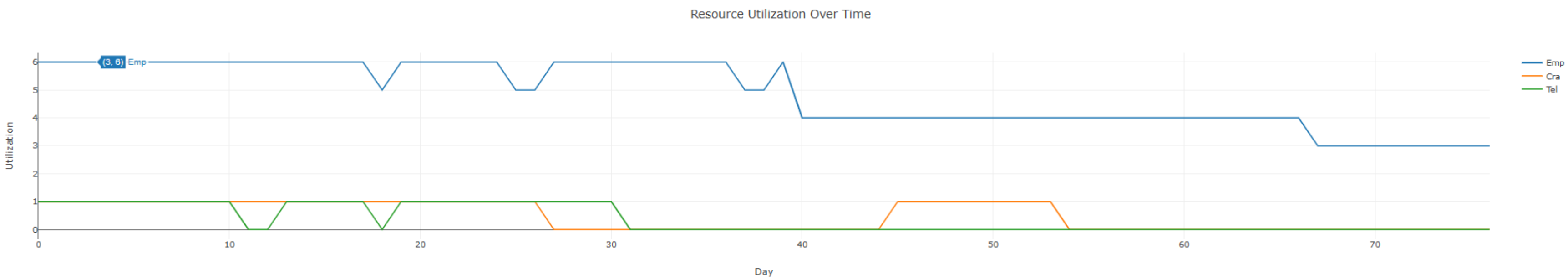


Improvement Heuristic:

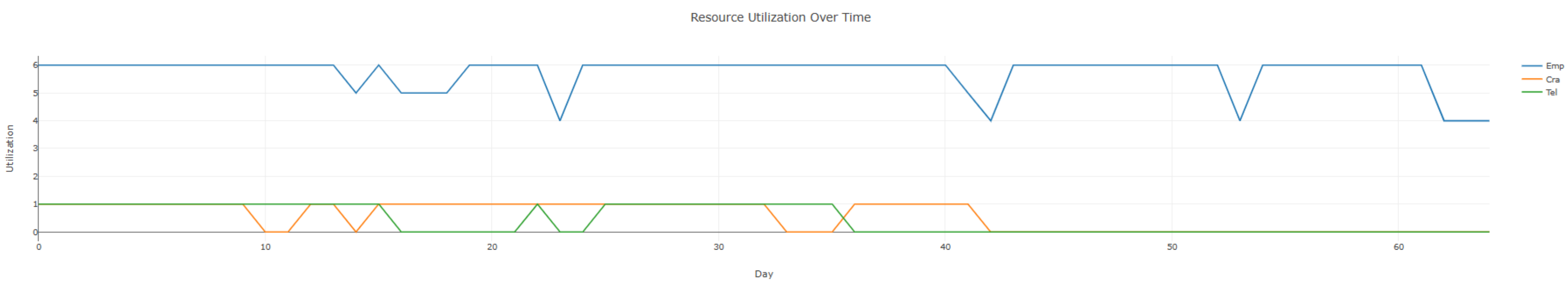


Low connectivity (IT = 10, ML = 100)

Constructive heuristic

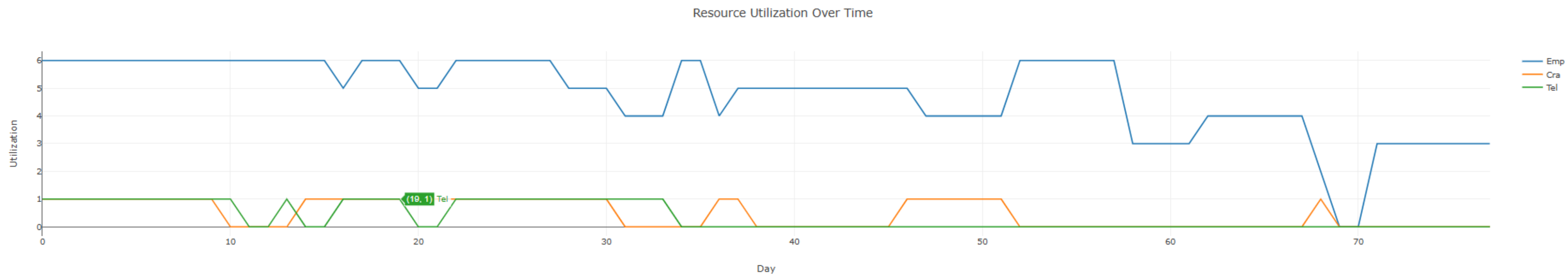


Improvement Heuristic

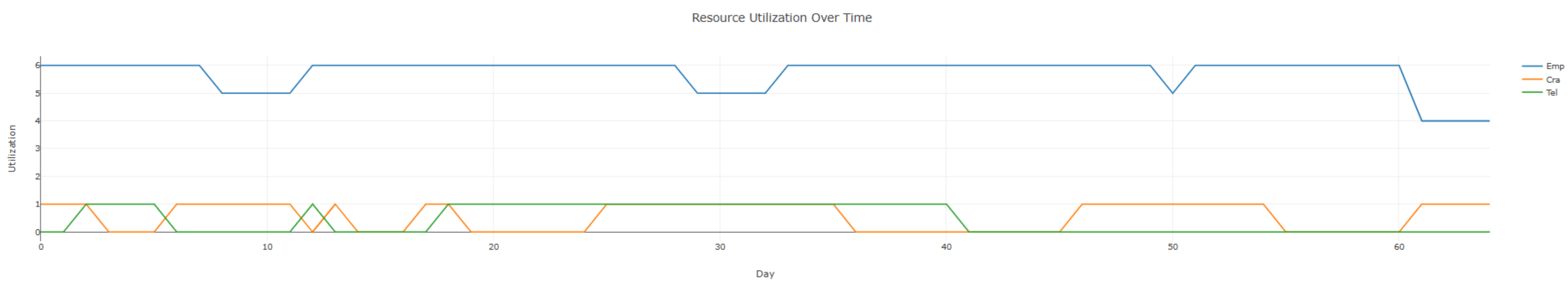


Medium Connectivity (IT = 10, ML = 100)

Constructive Heuristic

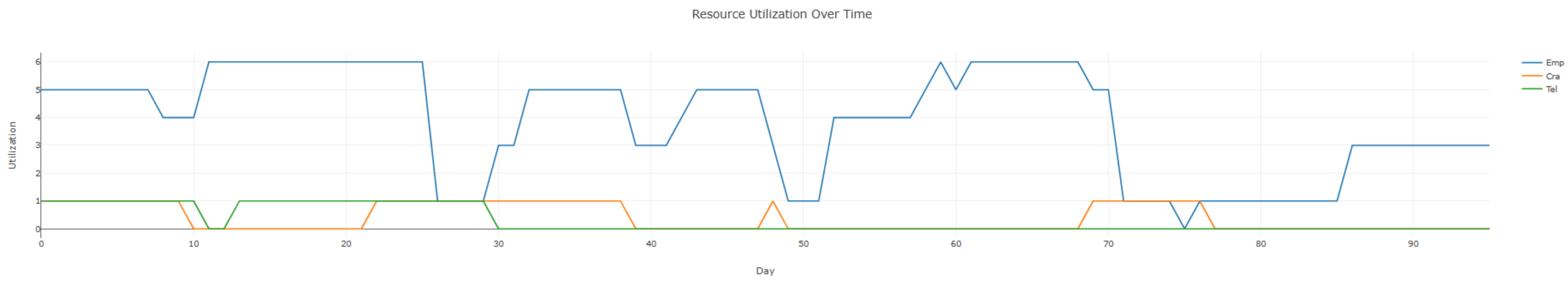


Improvement Heuristic

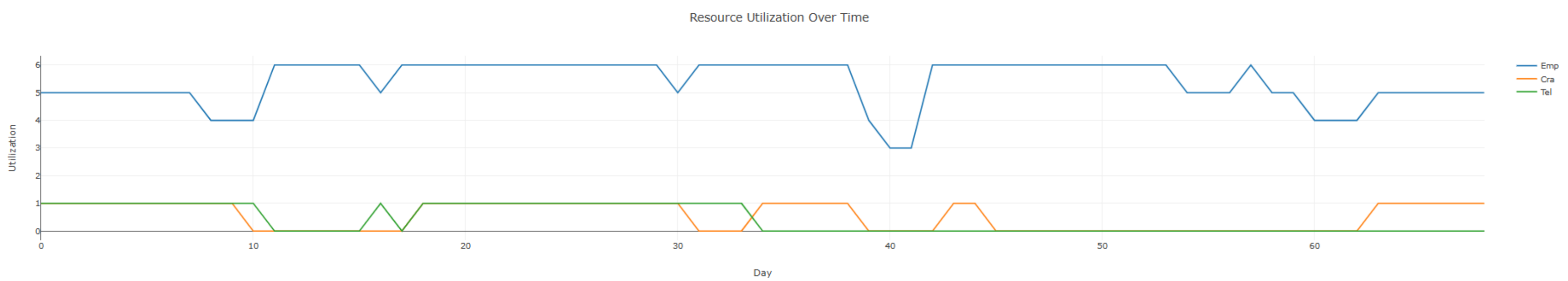


High Connectivity (IT = 10, ML = 100)

Constructive Heuristic

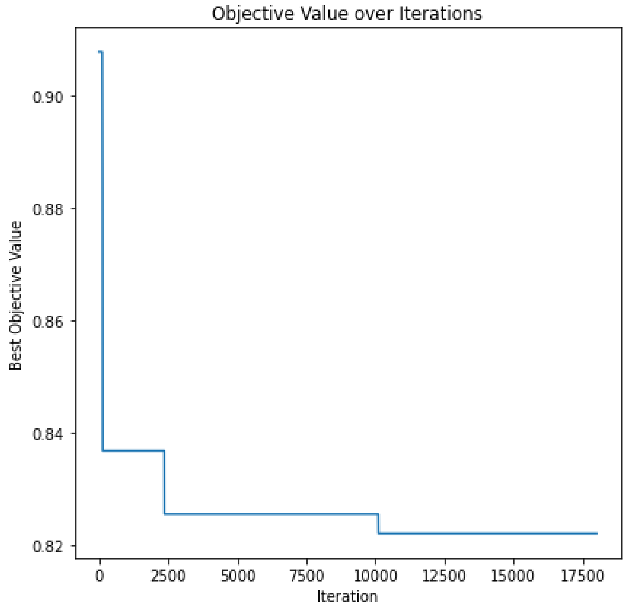


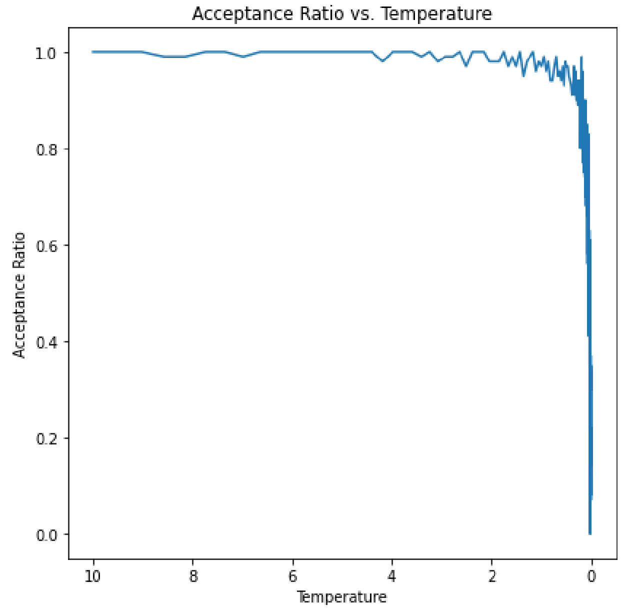
Improvement Heuristic



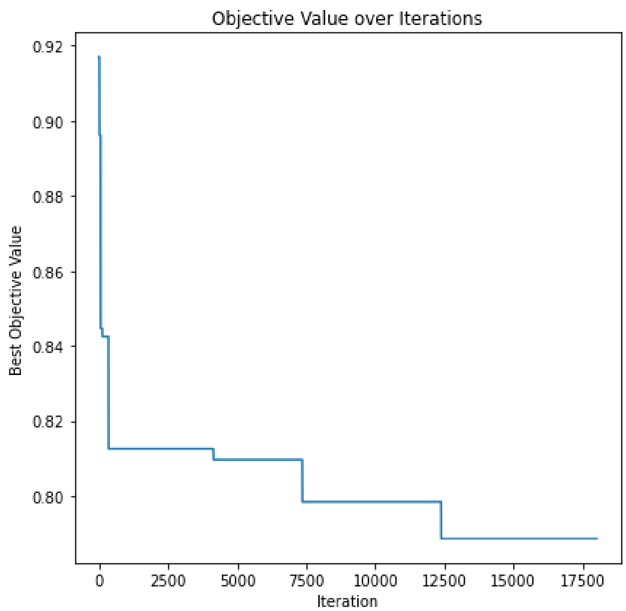
Appendix G – Objective Value over Iterations & Acceptance Ratio vs Temperature - Initial Settings

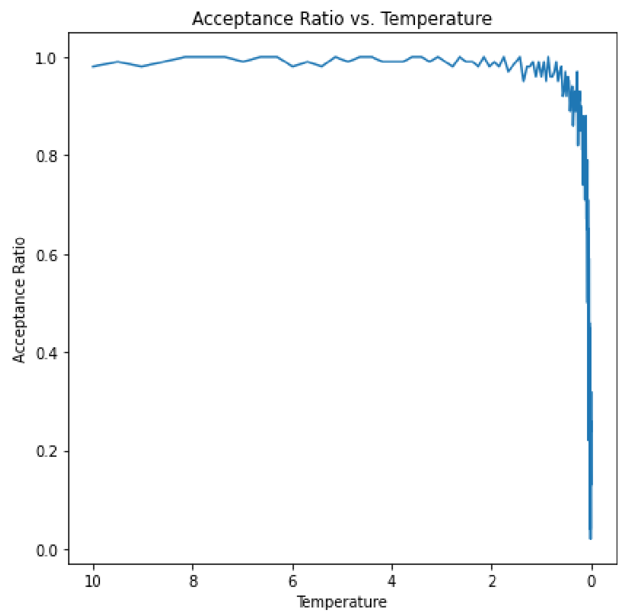
No connectivity



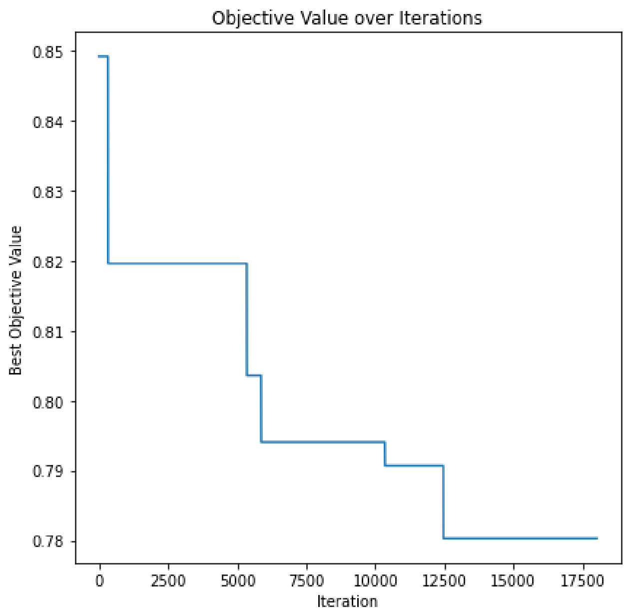


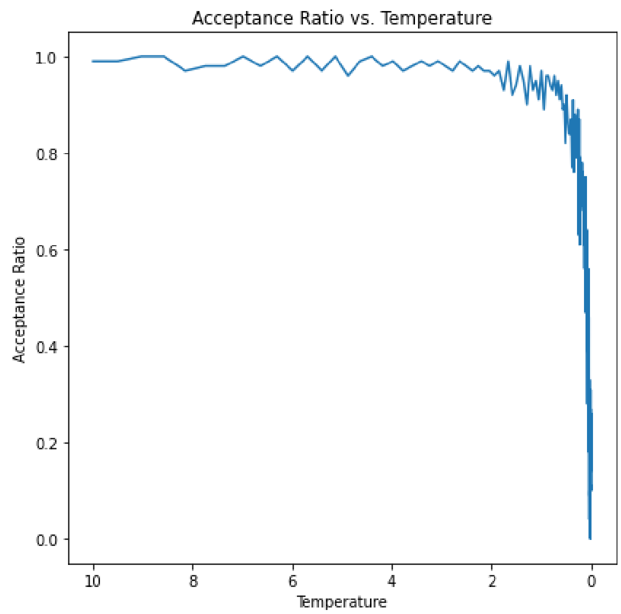
Low Connectivity



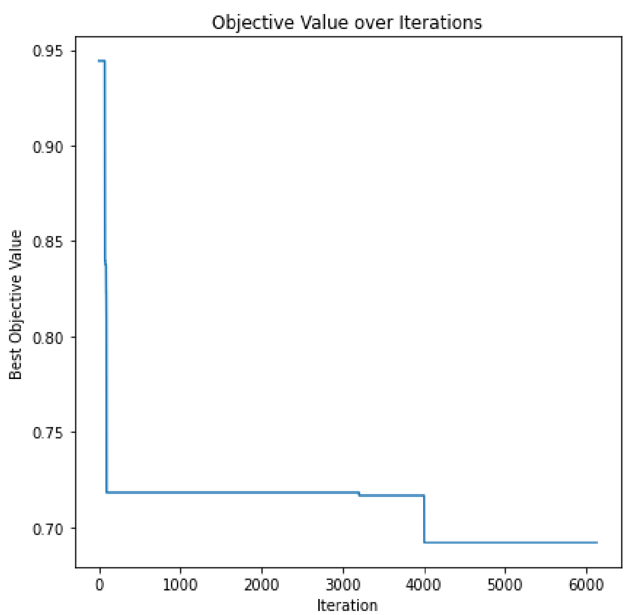


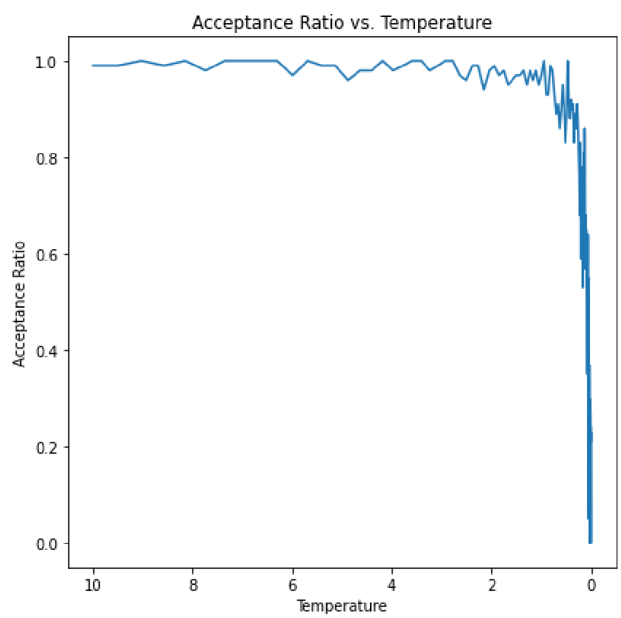
Medium Connectivity





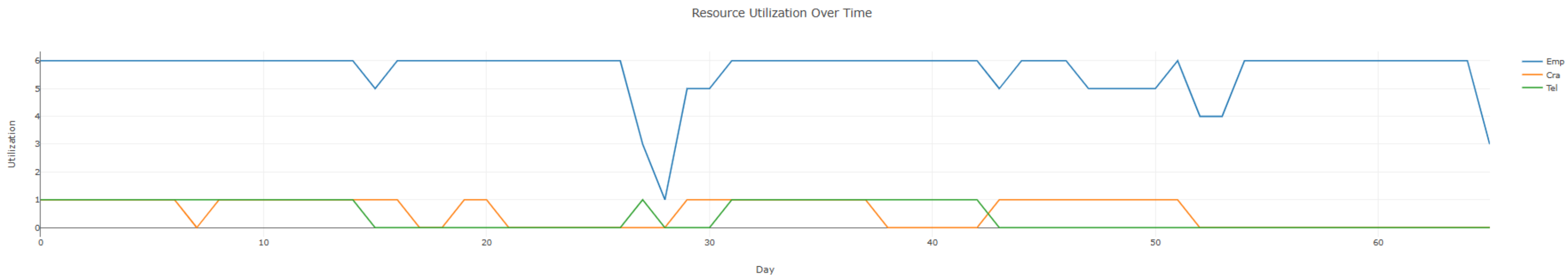
High Connectivity



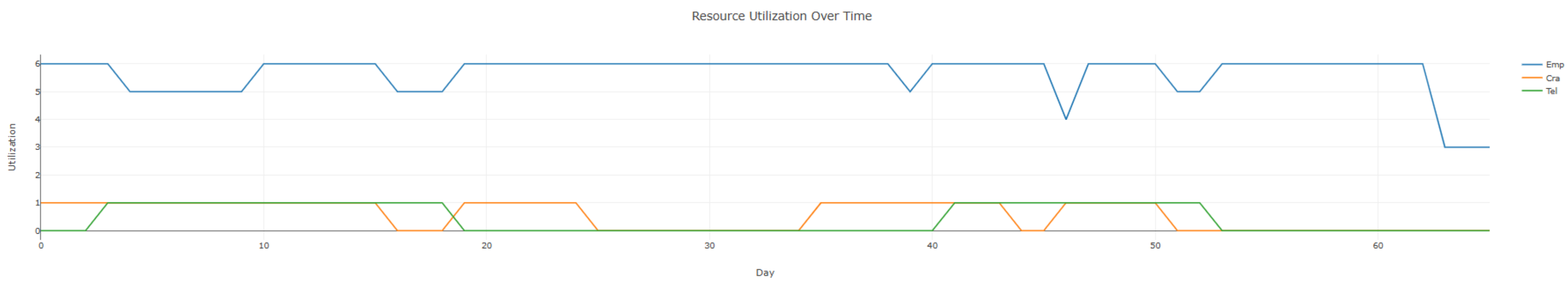


Appendix H- Resource Utilisation – Parameter Tuning

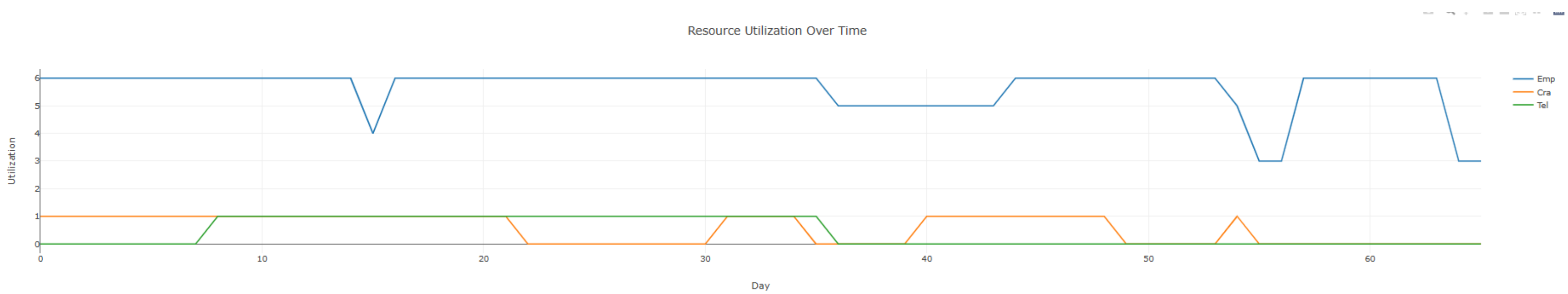
Medium Connectivity – IT1



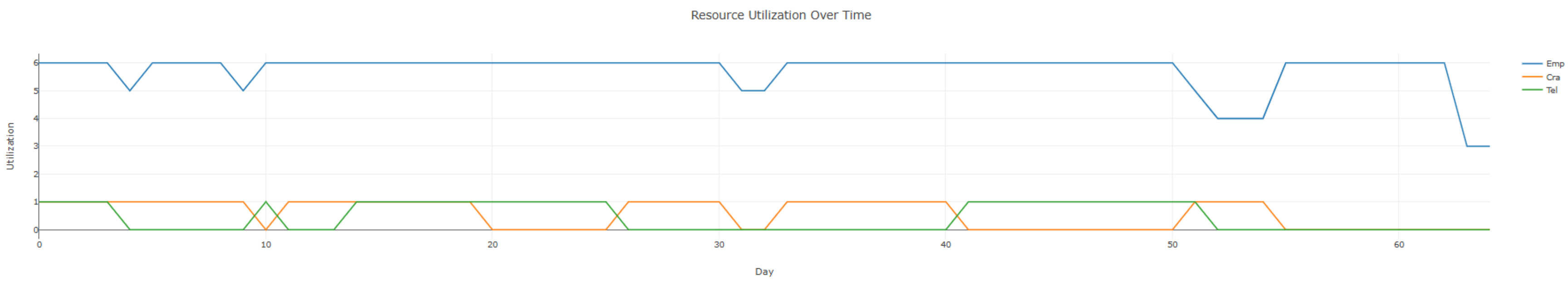
Medium Connectivity – IT2



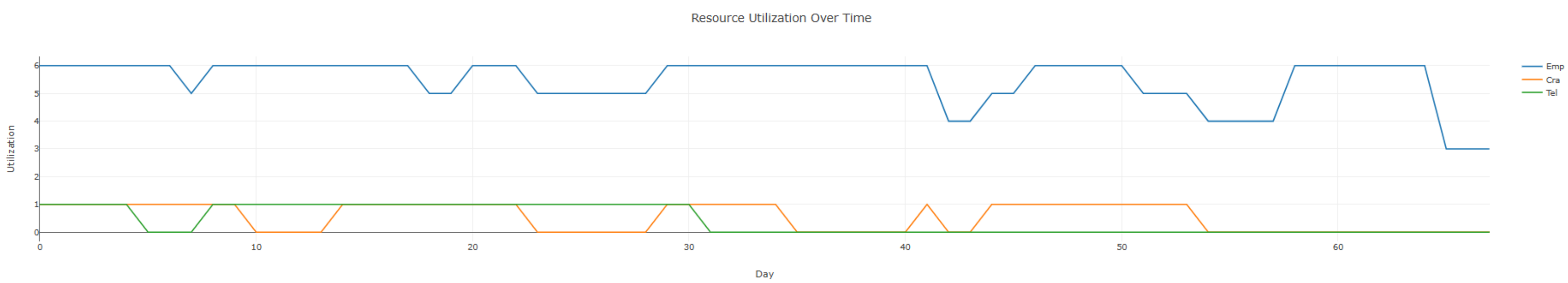
Medium Connectivity – ML1



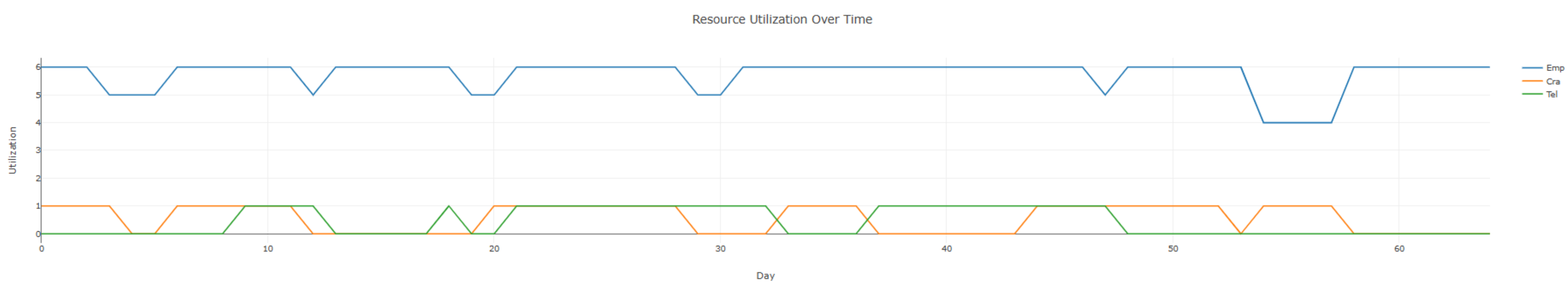
Medium Connectivity – ML2



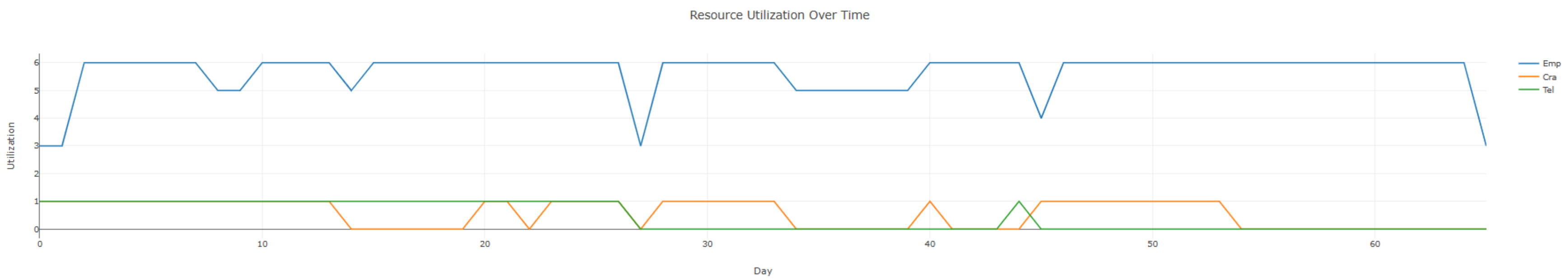
Medium Connectivity – DF1



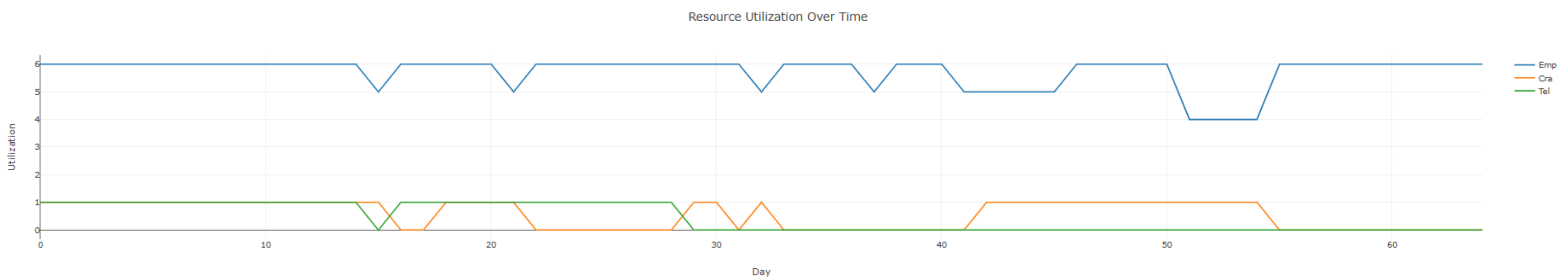
Medium Connectivity – DF2



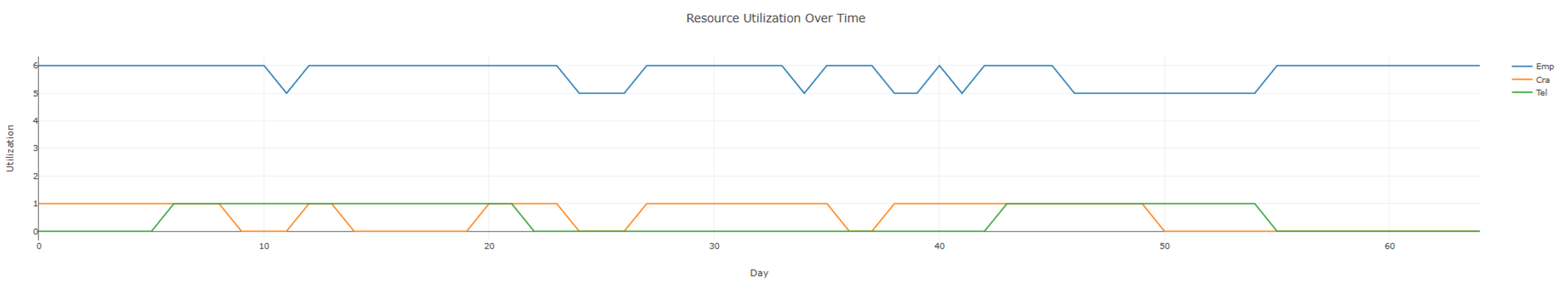
Medium Connectivity – DF-Refine 1



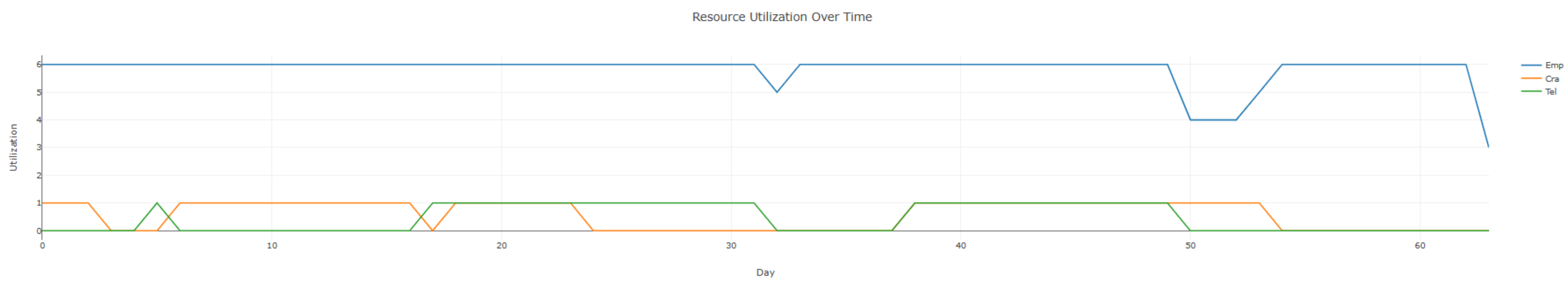
Medium Connectivity – DF-Refine 2



Medium Connectivity – ML-Refine 1

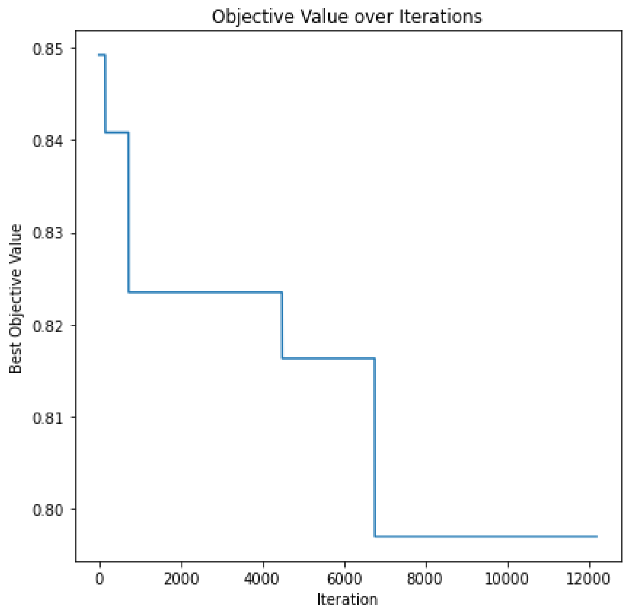


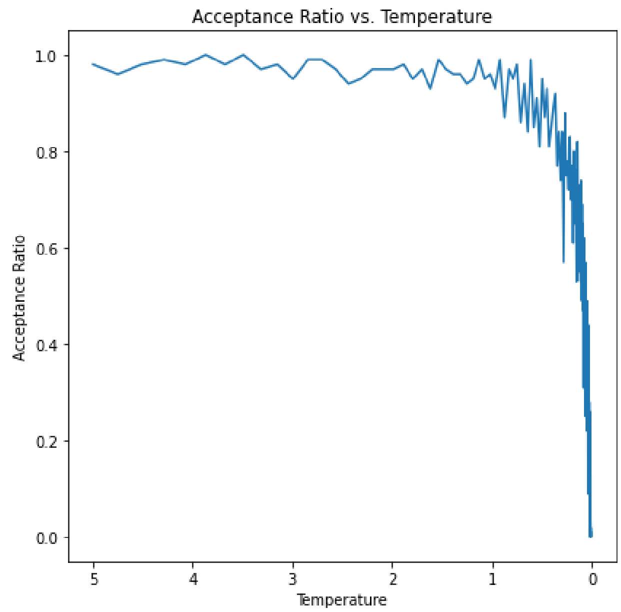
Medium Connectivity – ML-Refine 2



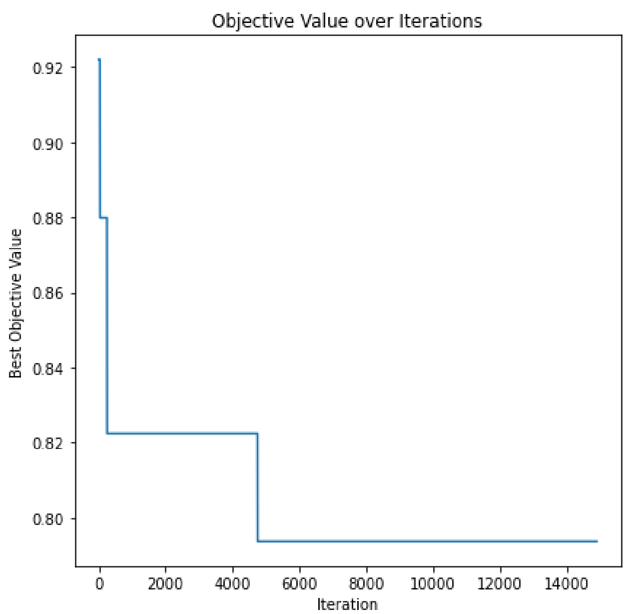
Appendix I - Objective Value over Iterations & Acceptance Ratio vs Temperature - Parameter Tuning

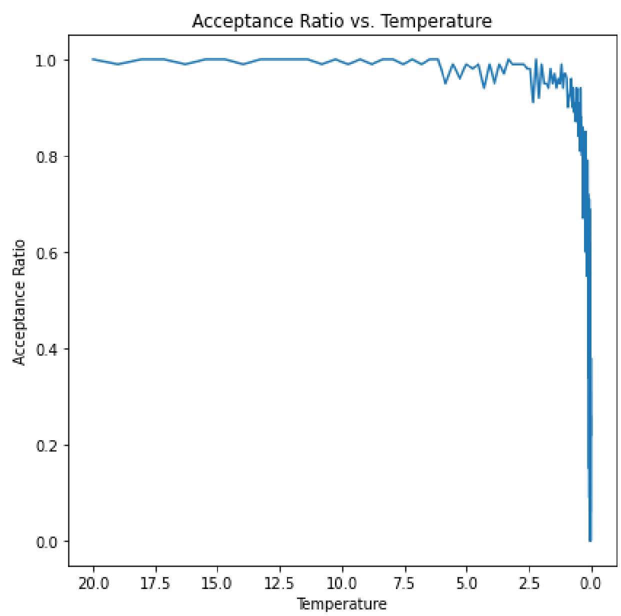
Medium Connectivity – IT1



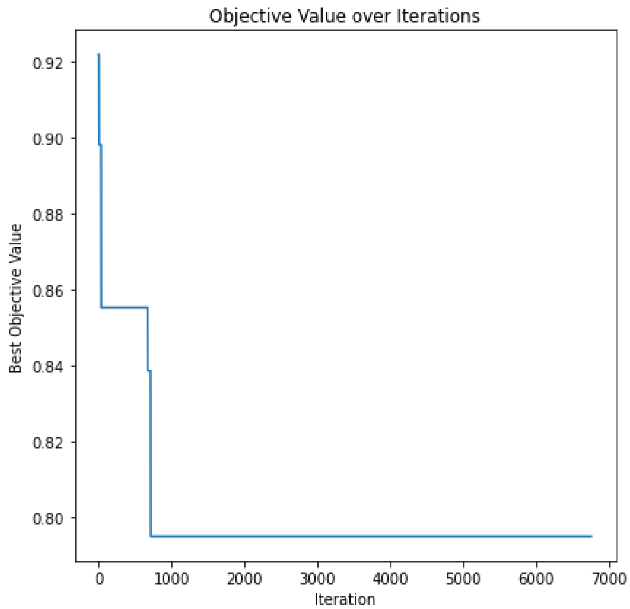


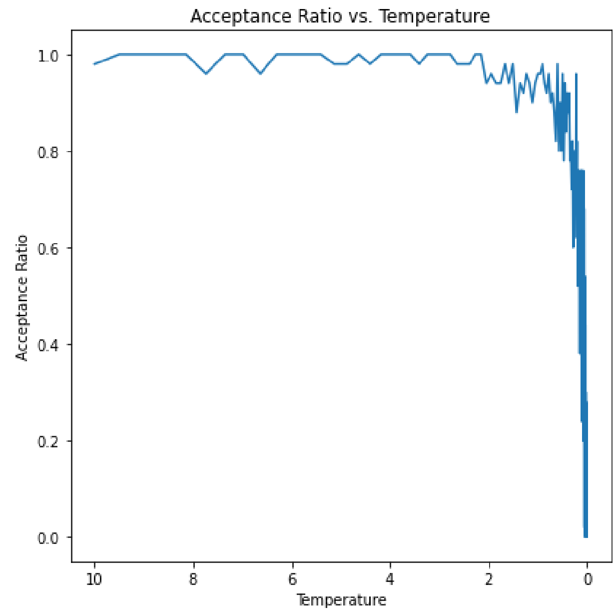
Medium Connectivity – IT2



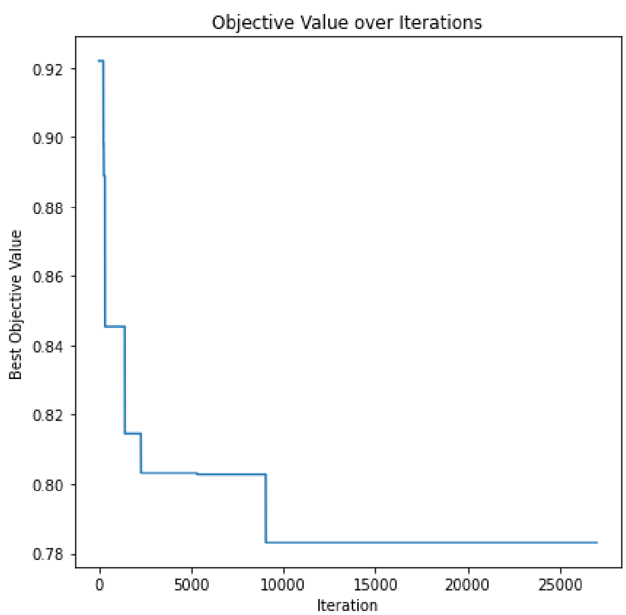


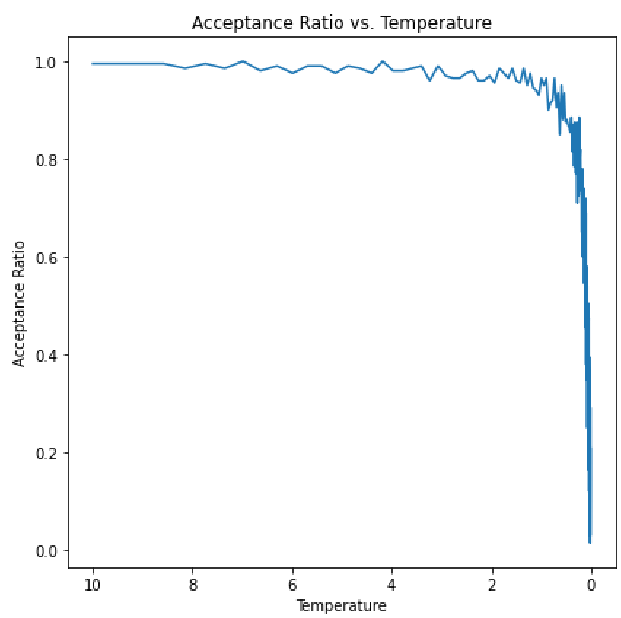
Medium Connectivity – ML1



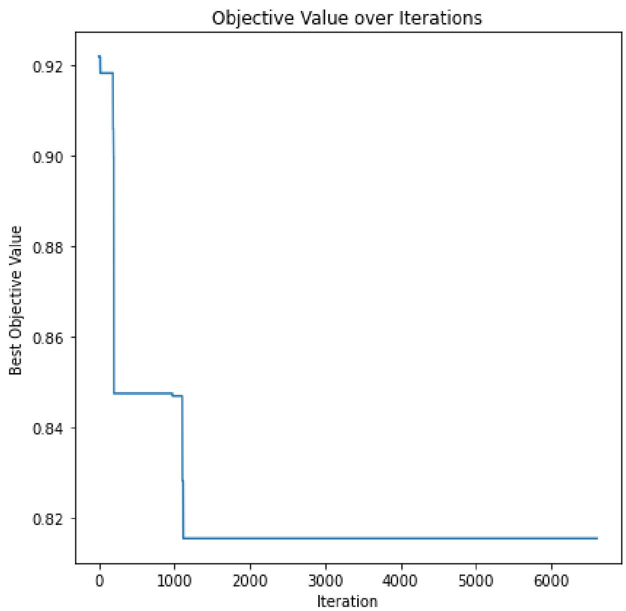


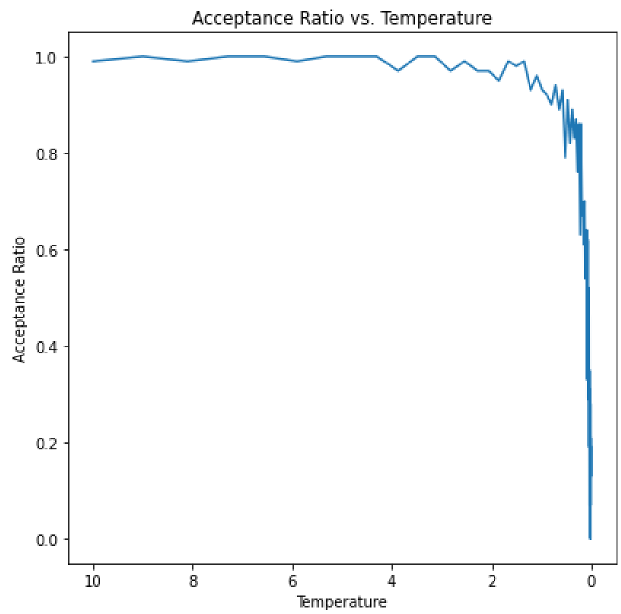
Medium Connectivity – ML2



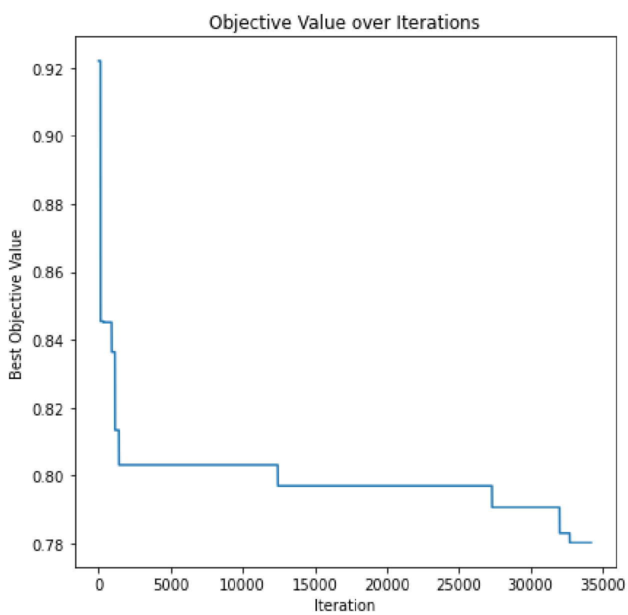


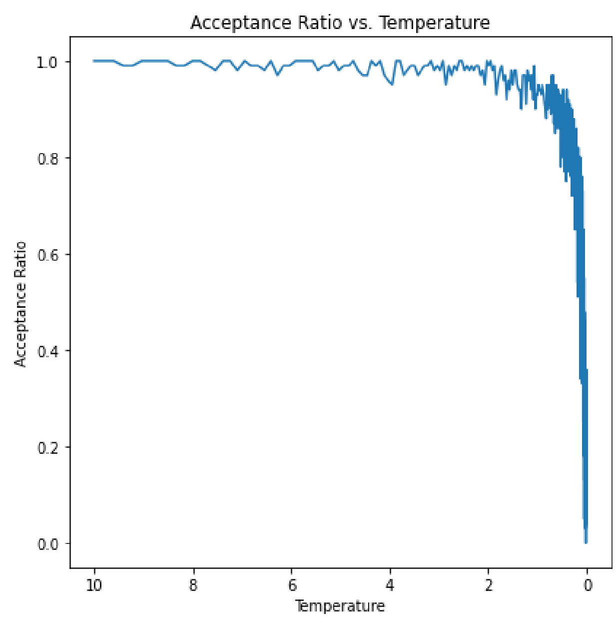
Medium Connectivity – DF1



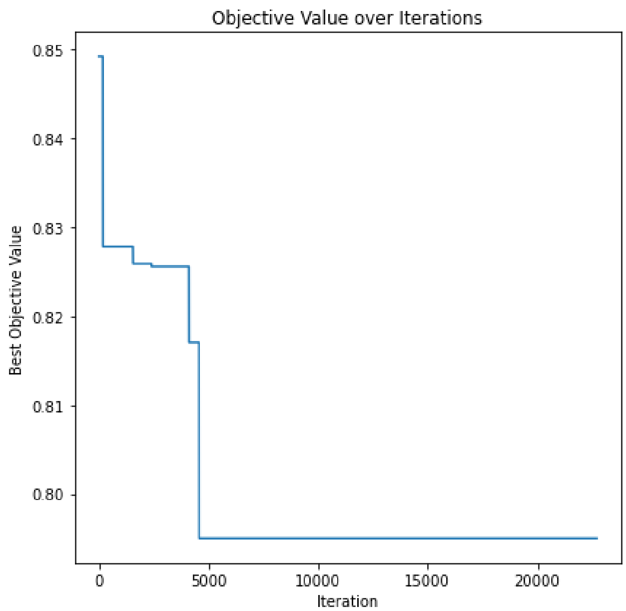


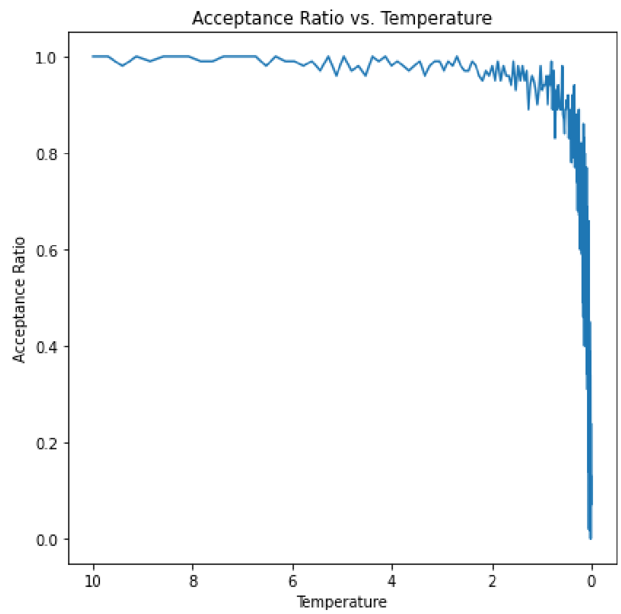
Medium Connectivity – DF2



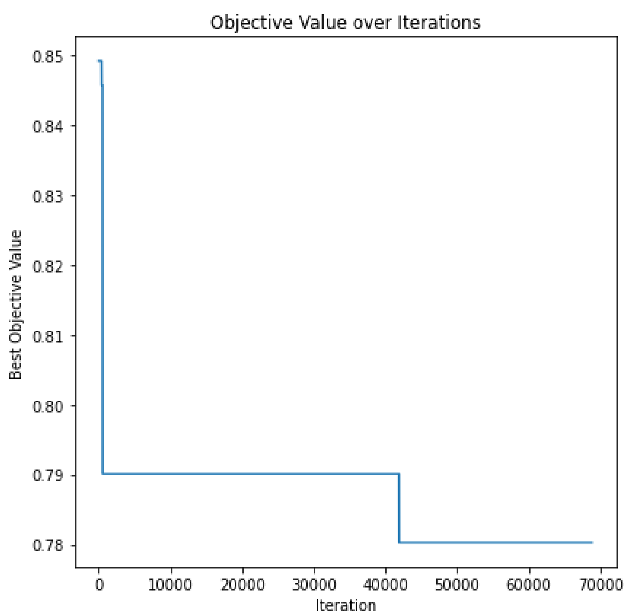


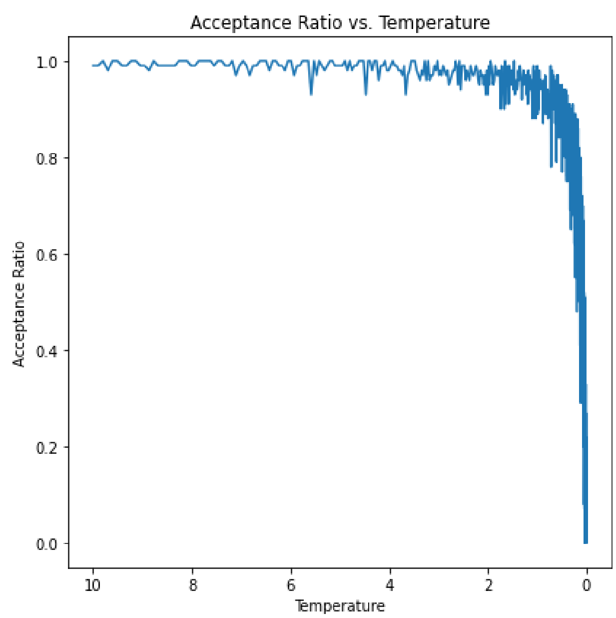
Medium Connectivity – DF-Refine 1



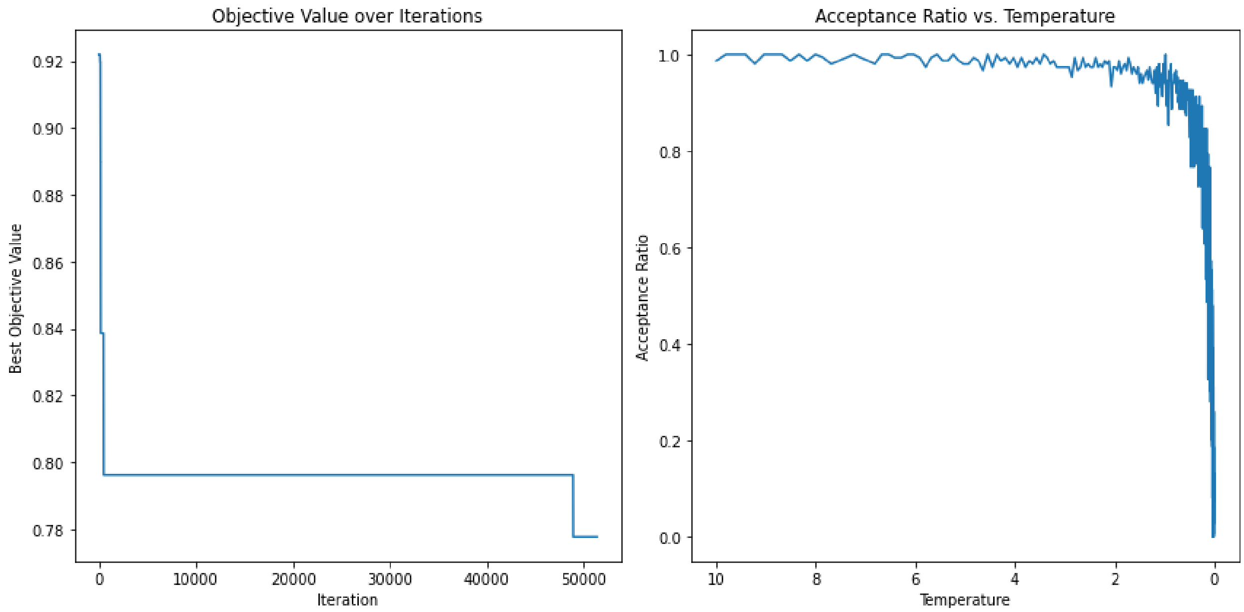


Medium Connectivity – DF-Refine 2

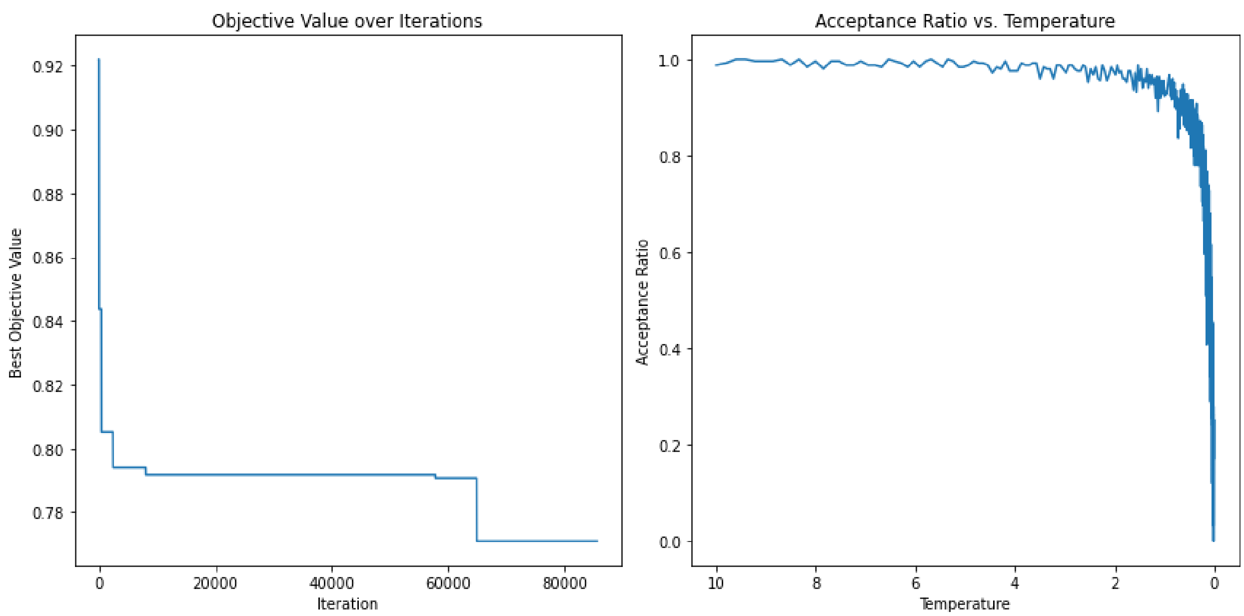




Medium Connectivity – ML-Refine 1

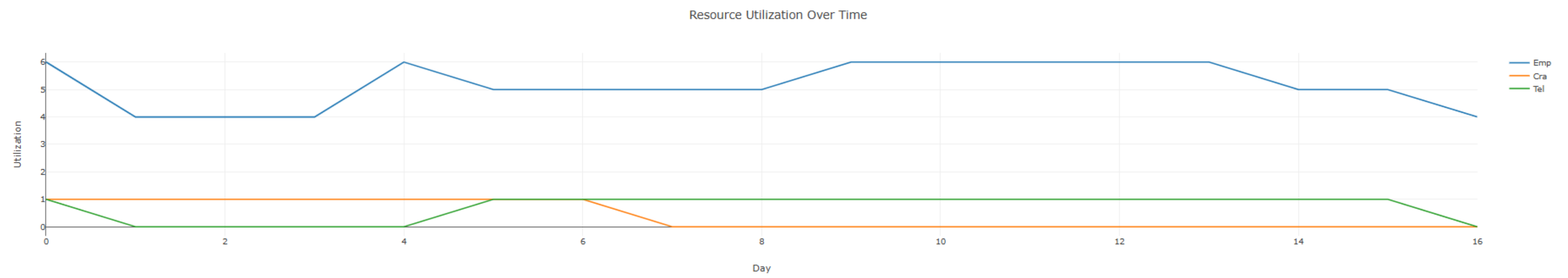


Medium Connectivity – ML-Refine 2

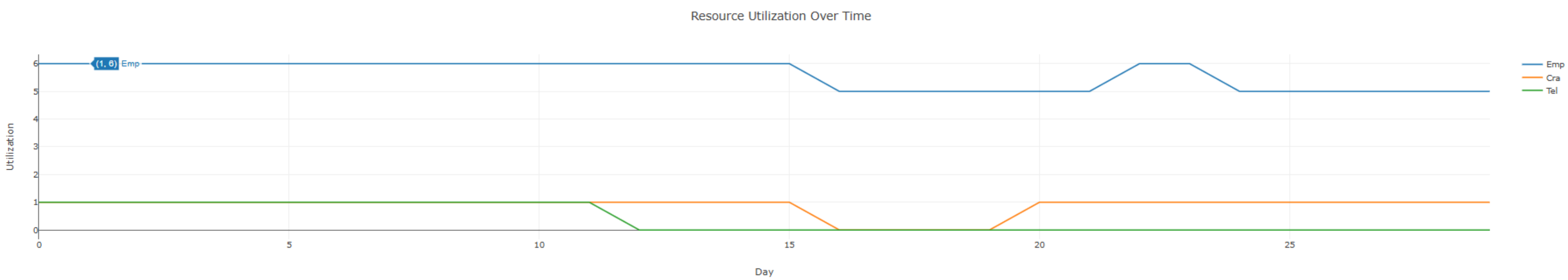


Appendix J - Resource Utilisation- Sensitivity Analysis - Varying the Number of Activities

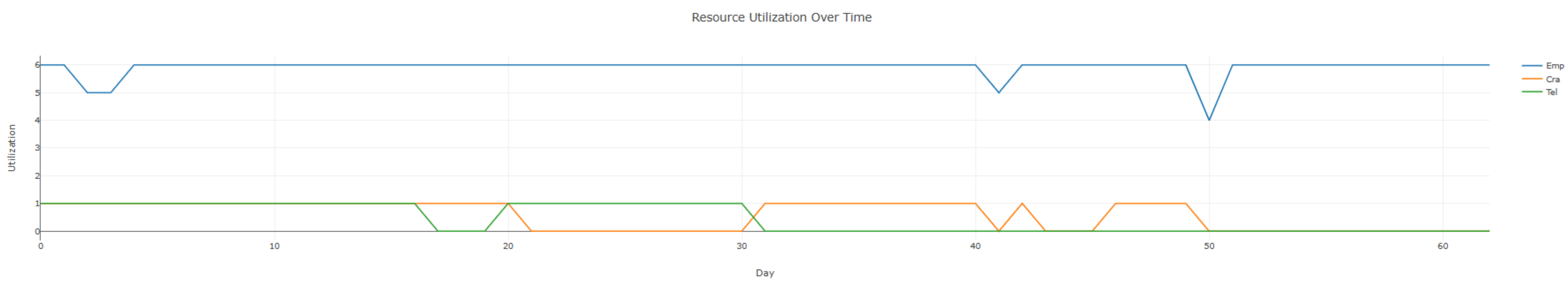
10 Activities



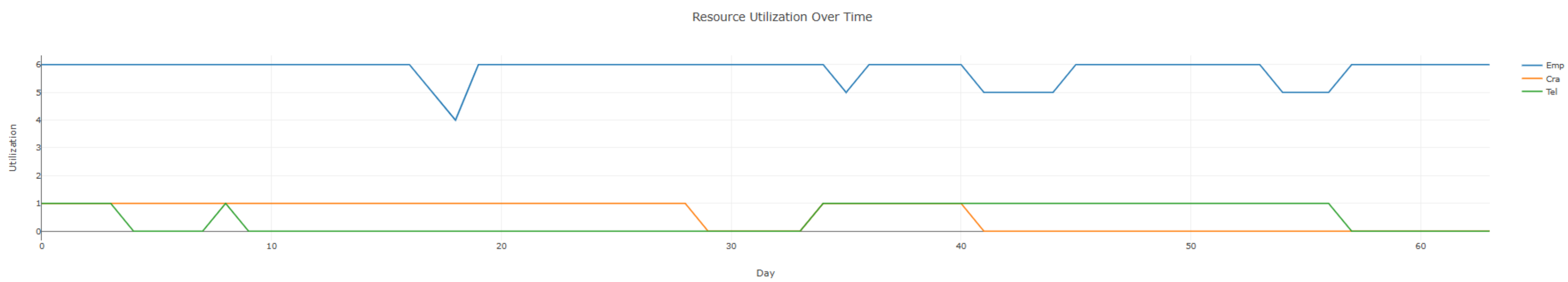
20 Activities



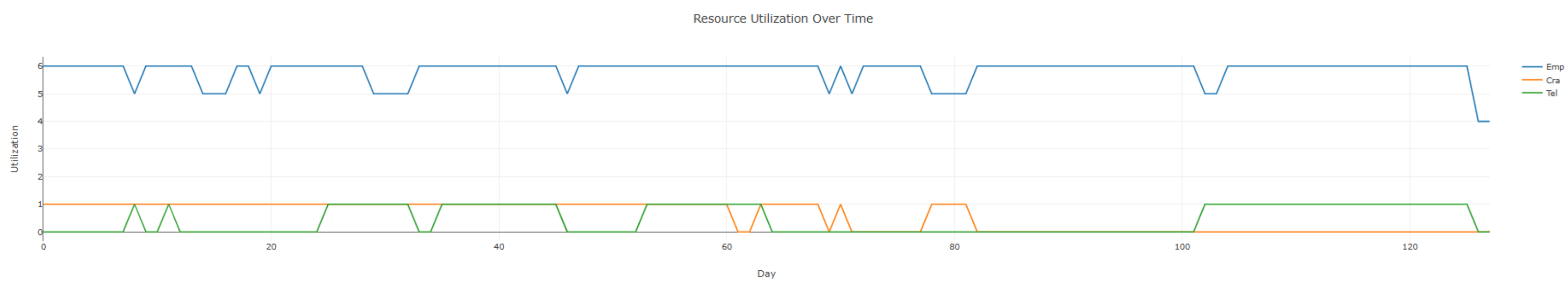
40 Activities



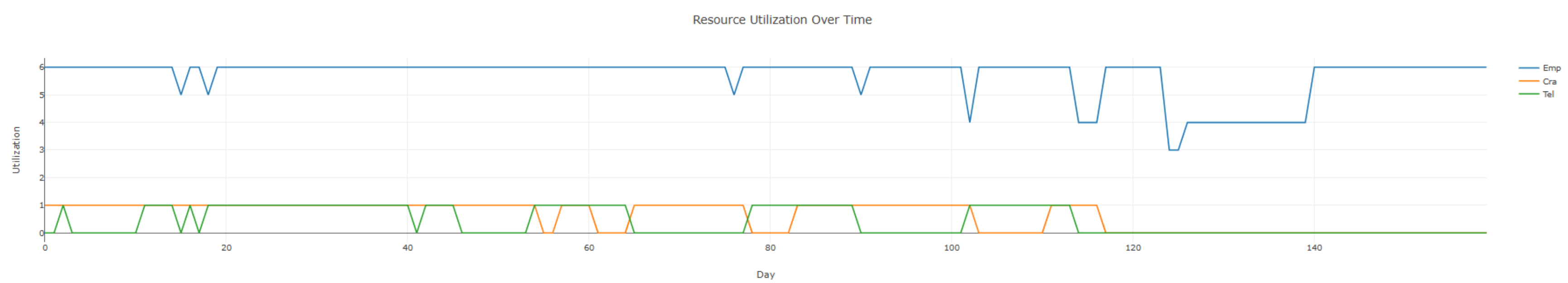
60 Activities



80 Activities

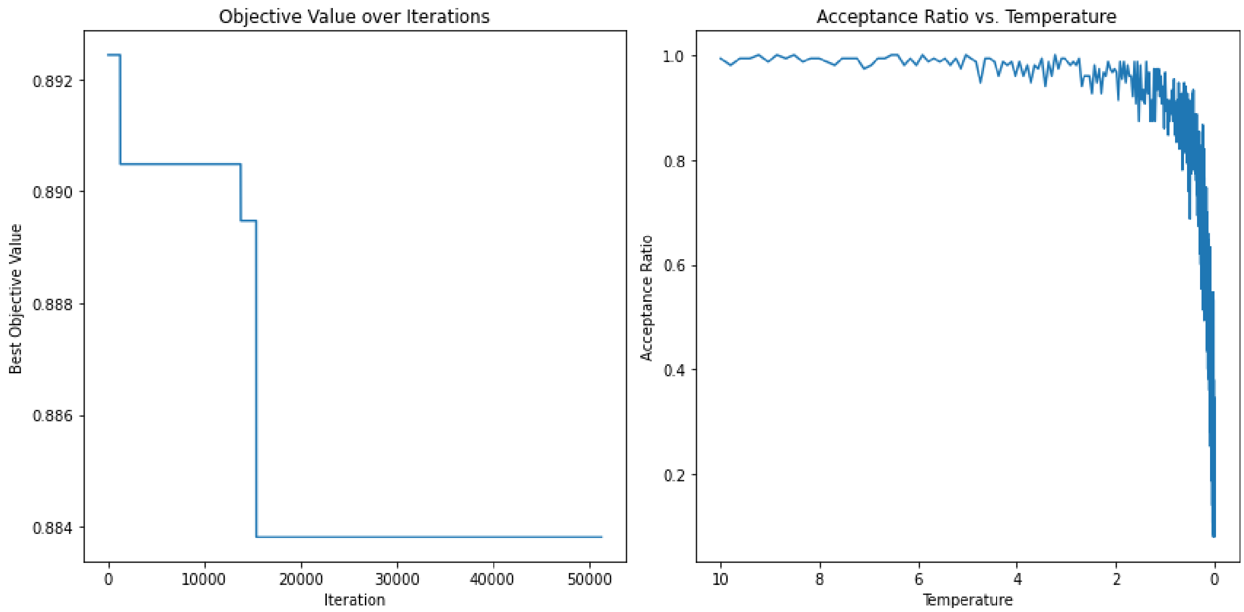


100 Activities

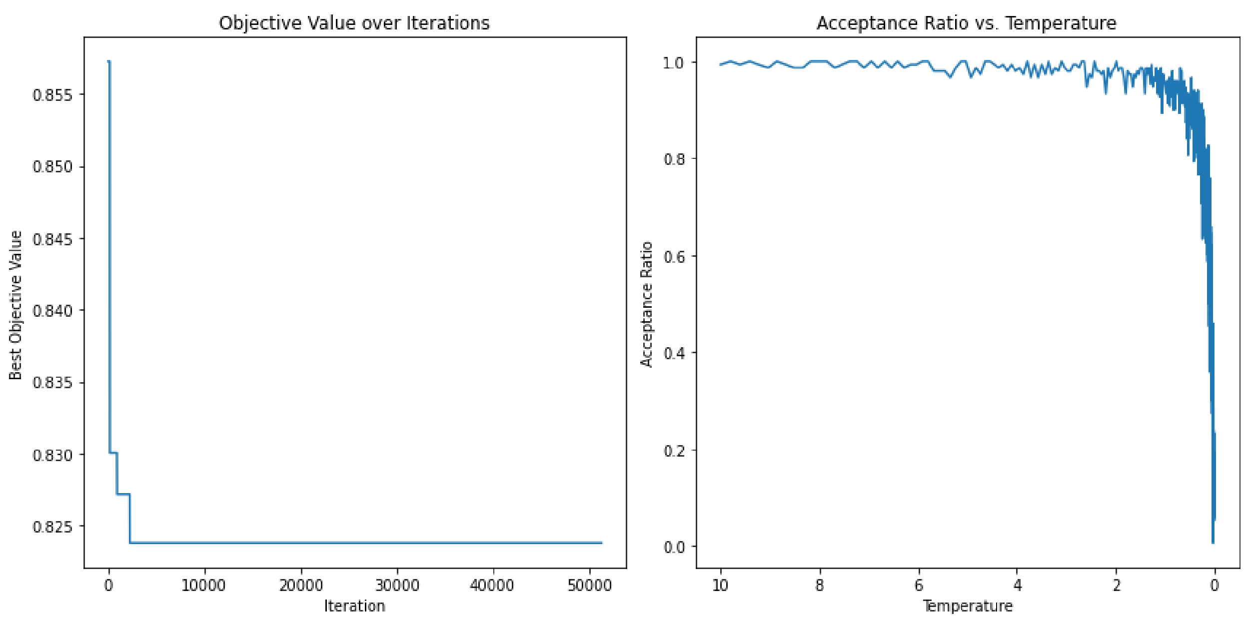


Appendix K - Objective Value over Iterations & Acceptance Ratio vs Temperature - Sensitivity Analysis - Varying the Number of Activities

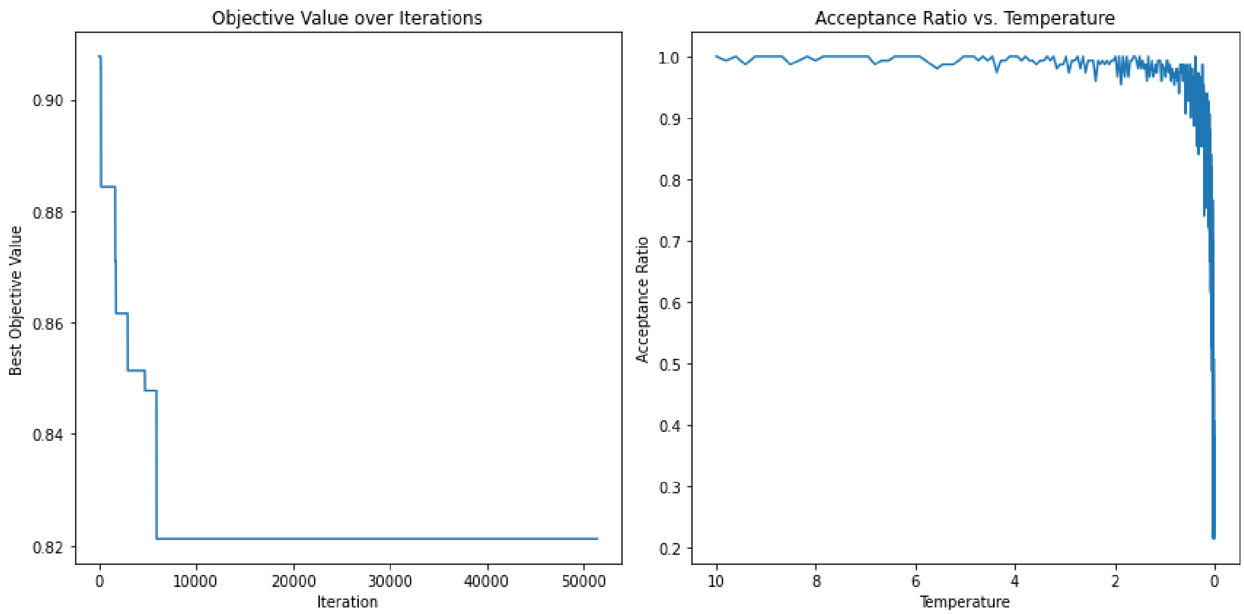
10 Activities



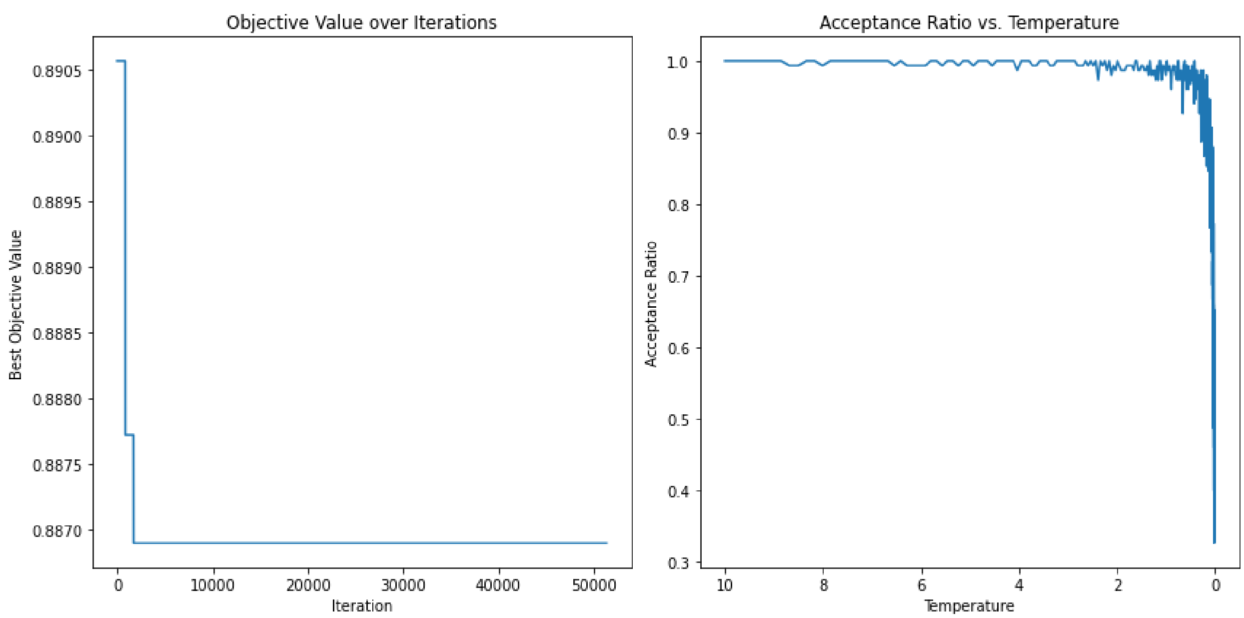
20 Activities



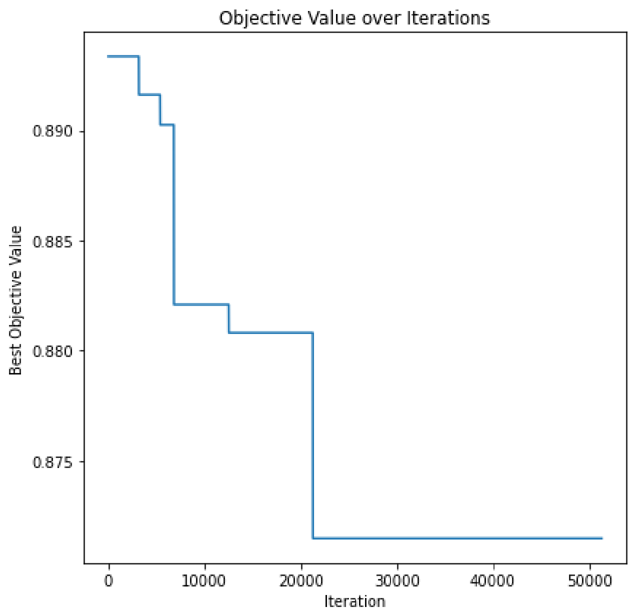
60 Activities

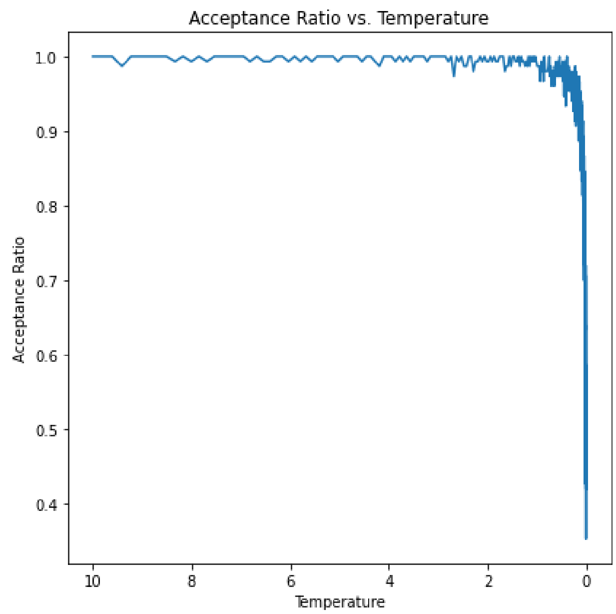


80 Activities



100 Activities





Appendix L - Excel sheets with the input information – Schedule Revision Method

Activities TAB

	Beschrijving Activiteit	Activiteit	Herplannen toegestaan?	Al begonnen?	Dagen nog te plannen	Onderaannemer	Initiële begindatum	Initiële einddatum
1	Aanbrengen druklaagwapening en wapening ihw betonvloeren	A1	Nee	Ja	2	Sub A	0	2
2	Aanbrengen installatiewerk in druklaag (Breman)	A2	Nee	Ja	2	Sub B	1	3
3	Controle wapening	A3	Nee	Ja	1	Sub C	2	3
4	Afzetten kolomvoetplaten tbv stort druklaag	A4	Nee	Ja	1		2	3
5	Storten druklaag	A5	Nee	Ja	1		3	4
6	Opschonen en verwijderen bekisting incl. maatvoering J-ankers.	A6	Nee	Ja	2		4	6
7	leggen prefab trappen	A7	Ja	Nee	1		6	7
8	Verwijderen tijdelijke ondersteuning compleet (bij druksterkte 14N/mm²)	A8	Ja	Nee	4		7	11
9	Afstellen/afwerken HSB wanden 2de verdieping	A9	Nee	Ja	5		0	5
10	Plaatsen en afstellen dakranden hoofdak incl dakranden atrium en trappenhuis.	A10	Ja	Nee	9		15	24
11	Montage staalconstructie sporthal incl. dakvlak	A11	Nee	Ja	2	Sub D	0	2
12	Aanbrengen veiligheidsnetten / randbeveiliging (sporthal en atrium)	A12	Nee	Ja	5	Sub E	5	10
13	Aanbrengen stalen dakplaten sporthal en atrium	A13	Ja	Nee	2		11	13
14	Montage staalconstructie atrium	A14	Nee	Ja	2	Sub D	1	3
15	Gelamineerde houten liggers atrium (Heko)/ plaatsing Hutten incl.	A15	Nee	Ja	1	Sub D	1	2
16	Plaatsing stalen hoofdtrap atrium incl. staalplaat-betonvloeren	A16	Ja	Nee	3	Sub D	13	16
17	montage staalconstructie dakopbouw incl.overig staal	A17	Ja	Nee	4	Sub D	20	24
18	Storten ihw gestorte betonvloer en staalplaatbetonvloer rondom hoofdtrap	A18	Ja	Nee	3		24	27
19	Plaatsen HSB wanden "sporthal" incl. dakranden - Deel 1	A19	Nee	Ja	6		5	11
20	Plaatsen HSB wanden "sporthal" incl. dakranden - Deel 2	A20	Ja	Nee	2		13	15
21	Afstellen HSB wanden "sporthal" incl dakranden	A21	Ja	Nee	3		15	18
22	Plaatsing trapbomen theathertrap en tribune trap	A22	Ja	Nee	2	Sub D	18	20

Activities – Predecessors TAB

Activity Description	Activity	Predecessor 1	Predecessor Percentage 1	Predecessor 2	Predecessor Percentage 2	Predecessor 3	Predecessor Percentage 3	P
Aanbrengen druklaagwapening en wapening ihw betonvloeren	A1							
Aanbrengen installatiewerk in druklaag (Breman)	A2	A1	50.00%					
Controle wapening	A3							
Afzetten kolomvoetplaten tbv stort druklaag	A4	A1	50.00%					
Storten druklaag	A5	A2	100.00%	A3	100.00%	A4	100.00%	
Opschonen en verwijderen bekisting incl. maatvoering J-ankers.	A6	A5	100.00%					
leggen prefab trappen	A7	A6	100.00%					
Verwijderen tijdelijke ondersteuning compleet (bij druksterkte 14N/mm²)	A8	A6	100.00%					
Afstellen/afwerken HSB wanden 2de verdieping	A9							
Plaatsen en afstellen dakranden hoofdak incl dakranden atrium en trappenhu	A10							
Montage staalconstructie sporthal incl. dakvlak	A11							
Aanbrengen veiligheidsnetten / randbeveiliging (sporthal en atrium)	A12	A11	50.00%					
Aanbrengen stalen dakplaten sporthal en atrium	A13	A12	100.00%					
Montage staalconstructie atrium	A14							
Gelamineerde houten liggers atrium (Heko)/ plaatsing Hutten incl.	A15							
Plaatsing stalen hoofdtrap atrium incl. staalplaat-betonvloeren	A16	A13	100.00%					
montage staalconstructie dakopbouw incl.overig staal	A17	A16	100.00%					
Storten ihw gestorte betonvloer en staalplaatbetonvloer rondom hoofdtrap	A18	A17	100.00%					
Plaatsen HSB wanden "sporthal" incl. dakranden - Deel 1	A19	A11	100.00%					
Plaatsen HSB wanden "sporthal" incl. dakranden - Deel 2	A20	A13	100.00%	A19	100.00%			
Afstellen HSB wanden "sporthal" incl dakranden	A21	A20	100.00%					
Plaatsing trapbomen theathertrap en tribune trap	A22	A21	100.00%					

Resources TAB

Resource	Afkorting Resource	Dag 1	Dag 2	Dag 3	Dag 4	Dag 5	Dag 6	Dag 7	Dag 8	Dag 9	Dag 10	Dag 11	Dag 12	Dag 13	Dag 14	Dag 15	Dag 16	Dag 17	Dag 18	Dag 19
Employees	Emp	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Crane	Cra	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Resource	Dag 20	Dag 21	Dag 22	Dag 23	Dag 24	Dag 25	Dag 26	Dag 27	Dag 28	Dag 29	Dag 30	Dag 31	Dag 32	Dag 33	Dag 34	Dag 35	Dag 36	Dag 37	Dag 38	D
Employees	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
Crane	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Resource Assignments TAB

Beschrijving Activity	Activity	Subcontractor	Resource 1	Resource Units 1	Resource 2	Resource U	Re
Aanbrengen druklaagwapening en wapening ihw betonvloeren	A1	Sub A					
Aanbrengen installatiewerk in druklaag (Breman)	A2	Sub B					
Controle wapening	A3	Sub C					
Afzetten kolomvoetplaten tbv stort druklaag	A4		Emp	2			
Storten druklaag	A5		Emp	2			
Opschonen en verwijderen bekisting incl. maatvoering J-ankers.	A6		Emp	2			
leggen prefab trappen	A7		Emp	2	Cra	1	
Verwijderen tijdelijke ondersteuning compleet (bij druksterkte 14N/mm²)	A8		Emp	2			
Afstellen/afwerken HSB wanden 2de verdieping	A9		Emp	3			
Plaatsen en afstellen dakranden hoofdak incl dakranden atrium en trappenhu	A10		Emp	2			
Montage staalconstructie sporthal incl. dakvlak	A11	Sub D			Cra	1	
Aanbrengen veiligheidsnetten / randbeveiliging (sporthal en atrium)	A12	Sub E					
Aanbrengen stalen dakplaten sporthal en atrium	A13		Emp	3	Cra	1	
Montage staalconstructie atrium	A14	Sub D					
Gelamineerde houten liggers atrium (Heko)/ plaatsing Hutten incl.	A15	Sub D					
Plaatsing stalen hoofdtrap atrium incl. staalplaat-betonvloeren	A16	Sub D			Cra	1	
montage staalconstructie dakopbouw incl.overig staal	A17	Sub D			Cra	1	
Storten ihw gestorte betonvloer en staalplaatbetonvloer rondom hoofdtrap	A18		Emp	2	Cra	1	
Plaatsen HSB wanden "sporthal" incl. dakranden - Deel 1	A19		Emp	3			
Plaatsen HSB wanden "sporthal" incl. dakranden - Deel 2	A20		Emp	3			
Afstellen HSB wanden "sporthal" incl dakranden	A21		Emp	3			
Plaatsing trapbomen theathertrap en tribune trap	A22	Sub D			Cra	1	

Disruption TAB

Activity Name	Disruption Date	Impact Disruption
A7	6	8