MSC. INDUSTRIAL ENGINEERING AND MANAGEMENT

OPTIMAL INSPECTION POLICIES FOR SEWER NETWORKS UNDER RESOURCE CONSTRAINTS USING PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

AUTHOR : ABEL KEIJZER (S1863169)

DATE : FEBRUARY 22, 2024

UNIVERSITY OF TWENTE.

Title	OPTIMAL INSPECTION POLICIES FOR SEWER NETWORKS UNDER RESOURCE CONSTRAINTS USING PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES
Date (DD-MM-YYYY)	*22-02-2024*
Main author	*Abel Keijzer* *MSc. Industrial Engineering and Management*
Supervisors	*Dr. E. Topan (BMS), University of Twente* *Dr. H. Chen (BMS), University of Twente* *L. A. Jimenez-Roa (FMT), University of Twente*
Number of pages	*65*
Contact	*Email: A.l.Keijzer@student.utwente.nl*
Educational facility	*University of Twente*
Department	*Behavioral, Management and Social Sciences*
Study program	*Industrial Engineering and Management*
Contact	*University of Twente* *Postbus 217* *7500AE Enschede* *(Overijssel), The Netherlands* *Tel. 0534 89 91 111*

This thesis was written as an assignment for graduation from the study program Industrial Engineering and Management at the University of Twente in The Netherlands.

UNIVERSITY OF TWENTE.

Acknowledgements

This report will mark the last graded project I will have produced as a student IEM at the University of Twente. It is the culmination of long hours and perseverance. This could not have been possible without the help of the people around me to support each step of the way. From those actively involved to those encouraging me without any knowledge of the content, I am thankful for all the support throughout the process.

A massive thank you to my supervisors for managing my energy towards a focused goal. It is through your effort that I can finally move to the next stage in my life. I want to first thank Lisandro and Engin for their patience and understanding. It has taken me much longer than I hoped to end where I am, and I appreciate the additional effort this has taken throughout these times. Furthermore, you have allowed me to understand the process of research much better. It is both with shame of my mistakes and pride of my growth that I look back at our earliest meetings, where I was unable to grasp the concepts and core values of research. Furthermore, I would like to thank Hao for his willingness to be a second supervisor, without knowing much about me, or much about the progress of the research at that time. It has been a pleasure to work with all of you, and I appreciate all the new skills I have learned throughout this project.

Finally, many thanks to my family and friends, who supported me when I suddenly had to move without notice. Additionally, our collective work sessions have made this possible. The motivation I received through our combined struggle has been invaluable.

Abstract

Inspection strategies are an important part of asset management. Efficient strategies and schedules can reduce the number of failures and improve the return on investment. Sewer networks use these strategies to optimize the expected state of the network and minimize the downtime. These networks are an important part of modern infrastructure and reduce risks of waterborne diseases(Nguyen & Seidu, 2022). However, the network has expanded in recent years without more allocation of budget (Obradović et al., 2023). Therefore, the scheduling strategies have had to be optimized further. Without a self-announcing failure, difficult degradation patterns, and a long-expected lifespan, optimizing the maintenance strategy has been difficult. This paper presents a model based on the theory of the partially observable Markov decision process (POMDP), where the maintenance schedule is optimized. The schedule considers a predefined horizon and includes both rehabilitation and inspection. The value of inspection is modeled through posterior probabilities of the expected state space. The model is solved using mixed integer linear programming (MILP). The resulting schedules of the model are compared to simpler scheduling strategies. The differences in expected costs are used to determine if this model is able to construct an appropriate and efficient maintenance schedule for a single pipe.

Management Summary

The research examines the problems of scheduling maintenance for a sewer network. The network is a vital part of infrastructure, and has allowed safe transportation of wastewater, greatly reducing the risks of waterborne illnesses spreading (Obradović et al., 2023). The reliance on this network is high, and the size has only increased in recent years, without the necessary increase in budget. Consequently, asset managers have had to improve maintenance efficiency for the network. However, a sewer network is located underground, where the condition of pipes is hidden until inspection. Furthermore, a failed pipe can remain hidden for an indeterminate amount of time, based on factors such as the soil composition, demand (Draude et al., 2022). These two factors have led to the following research question:

To what extent do budgetary constraints for maintenance affect the determination of optimal inspection intervals of inspection decisions for sewer pipes?

For answering this question, the following research questions have been defined:

1. *What information is required for modeling optimal inspection intervals and policies?*
2. *How can you evaluate the performance of inspection interval policies?*
3. *What methods can solve the model for optimal inspection policies?*
4. *How do inspection strategies change under different constraints and environments?*

Problem context

Pipes in the network degrade over time. This degradation is modeled through one of three main models: statistical, Markov chains, or machine learning (Hawari et al., 2020). These models simulate the degradation of a sewer pipe to estimate the expected future condition. Based on this future condition, a maintenance schedule is constructed. A failure can be described as a hydraulic failure, structural failure, or operational failure (Anbari et al., 2017). A hydraulic failure means the system is unable to transport the content, leaving the pipe to overflow A hydraulic failure is often an indicator of another failure in a nearby pipe. Operational failures refer to build up of grease or fat, or tree roots in the pipe that block transportation of wastewater. Finally, the structural failures are failures of the pipe due to degradation. These are the most predictable through modeling and are the aim to prevent with maintenance.

Literature review

Inspection policies of sewer pipes should be created based on the expected current state. A blanket policy has been shown to underperform compared to a dynamic scheduling policy (de Jonge & Scarf, 2020). For modeling a dynamic scheduling policy, the environment must be modeled to represent the costs of the pipe per moment in time. Since failures are hidden, penalty costs are used, that refer to the costs of a failure for each period this failure is remains undetected. With this, a POMDP can be used to represent the expected costs per period. This modeling method also considers the observational probabilities, which allows for modeling of imperfect inspections.

Model

Through the literature review, the system environment is modeled through a POMDP. This will also allow easier adaption for future research. A weakness of the POMDP is that solving can take a significant time. The model considers all possible observations and beliefs across these states. Consequently, solving time is generally long. One method of limiting the solving time is by implementing a pruning algorithm. This algorithm removes redundant cost vectors from the

solution, improving the solving speed. Here, pruning is done by modeling the schedule with the expected state and posterior state probabilities after an inspection. The assumption is made that no other observations come in between instances of maintenance. Therefore, the first action in this schedule will not change throughout the lifetime of the pipe. The subsequent schedule of the pipe is suboptimal since this schedule is built around the expected state. Thus, the schedule should be updated whenever an observation is made. This observation takes the form of an inspection, where the previously hidden state becomes clear with a certain observational probability. This subsequent schedule repeats the process, resulting in a policy for each possible outcome after an inspection. The policy at the current time will only suggest the next optimal action, whereafter the new information should be used to reschedule the maintenance.

A MILP model is constructed that prunes irrelevant value vectors and determines the time of the first action. A MILP is generally slow, but the method can provide an exact schedule, meaning the schedule is guaranteed to be optimal. Heuristic methods will not provide an optimal schedule and leave the solution with an uncertain gap to the optimal solution. Therefore, measuring performance with these methods is more difficult. However, the scalability of these methods is better when compared to MILP.

Results

A parametric analysis was conducted to determine reasonable testing parameters for evaluation of results. After determining appropriate values, the performance of the model was tested against simpler maintenance policies. Results show that the model outperforms simpler scheduling methods and reoptimizing the schedule after inspections shows a significant difference in performance. Furthermore, the schedule changes significantly based on the number of resources allowed. When maintenance is the bottleneck, the schedules times change significantly.

Conclusion

Resource constraints should always be considered during the construction and optimization of maintenance schedules. Even if a budget is not identifiable, a hard limit on the number of actions will affect the schedule and performance significantly. Future research should expand this model to represent a real sewer network system more accurately. The model in this thesis has been made only for a single pipe and runtime becomes unreasonable when multiple pipes are considered simultaneously. For this, heuristics may have to be used, or a more efficient pruning method should be developed.

Table of Contents

Acknowledgements 2

Abstract 3

Management Summary 4

List of figures 9

1. Introduction 10

  1.1 Problem context 10

  1.2. Problem analysis 11

  1.3. Research goal 12

  1.4. Research questions 12

  1.5. Approach and contributions 13

2. Current system analysis 14

  2.1 Sewer networks 14

    2.1.1. System interaction and configuration 14

    2.1.2 Pipes 15

  2.2 Degradation 15

  2.3 Maintenance 16

  2.4 Data quality in sewer maintenance 17

  2.5 Conclusion 17

3. Literature review 19

  3.1. Inspection strategies 19

    3.2.1. Periodic inspections 19

    3.2.2. Static aperiodic inspections 19

    3.2.3. Dynamic aperiodic inspections 20

  3.2 Optimization objectives 21

    3.2.1 Inspection value 21

  3.3 Modeling methods 22

    3.3.1 Modeling environment 22

    3.3.2 The POMDP 23

    3.3.3 Model solving methods 24

    3.3.4 Maintenance effectiveness 25

  3.4 Conclusion 25

4. Model description 26

  4.1 Degradation model 26

    4.1.1 Markov Chain description 26

    4.1.2 Example 27

4.2 Optimization model 27
4.2.1 Model description and goal 28
4.2.2 Assumptions 29
4.2.3 POMDP 29
4.2.4 Objective function 30
4.3 Model example 31
4.4 Solving methods - POMDP 34
4.4.1 Exact solving methods 34
4.4.2 Approximate solving methods 35
4.5 Optimization model – MILP 35
4.5.1 Objective 36
4.5.2 Constraints 36
5. Results 39
5.1 Initial performance evaluation 39
5.1.1 Schedule evaluation 40
5.1.2 Performance evaluation 41
5.2 Parametric analysis 42
5.2.1 Degradation analysis 43
5.2.2 Penalty cost testing 44
5.2.3 Maintenance cost testing 44
5.2.4 Initial state 45
5.2.5 Horizon length 46
5.2.6 Discount factor 47
5.3 Testing budget restrictions 48
5.3.1 Comparing schedules 48
5.3.2 Dynamic and aperiodic schedule performance with budgets 50
5.4 Imperfect inspections 51
5.4.1 First experiment 52
5.4.2 Second experiment 53
6. Conclusion and recommendations 55
6.1 Conclusion 55
6.2 Added value of POMDPs 56
6.2 Limitations and discussion 57
6.3 Future research 58
6.4 Remarks 59
Appendix 60

A. The cost comparison of dynamic and aperiodic schedules 60
A.1 A budget of 6000 60
A.2 A budget of 10,000 60
A.3 A budget of 15,000 (repeated) 60
References 61

List of figures

Figure 1 The problem cluster 12
Figure 2 The research roadmap for the research 13
Figure 3 A visual representation of a Markov Chain 26
Figure 4: A visualization of the policy tree against the aperiodic schedule 40
Figure 5: The difference in the costs across a simulation and the confidence intervals 42
Figure 6 The aperiodic schedule with different levels of degradation 43
Figure 7 The aperiodic schedule with varying penalty levels 44
Figure 8 The aperiodic schedule with varying costs of actions 45
Figure 9 The aperiodic schedule with varying initial states 46
Figure 10 The aperiodic schedule with different horizon lengths 47
Figure 11 The aperiodic schedule with different discount factors 47
Figure 12 The aperiodic schedule with different budget levels 49
Figure 13 The expected costs for different budget values 50
Figure 14 The comparison of costs between the aperiodic and static schedule 51
Figure 15 The imperfect inspection expected costs 52
Figure 16 The imperfect inspection aperiodic schedule 52
Figure 17 The imperfect inspection aperiodic schedule with a budget of 8,500 53
Figure 18 The imperfect inspection aperiodic schedule with a budget of 10,000 54

1. Introduction

This project examines the difficulties of scheduling maintenance for sewer networks. This paper will document the process of developing a model that can schedule maintenance while considering the uncertainty of scheduling and failure characteristics of sewer networks. Section 1.1 provides a short introduction to the problem and state a research problem. Then section 1.2 will identify a research gap to help solve the research problem. Section 1.3 defines the goal of the research. Then section 1.4 proposes the main research question and sub-questions help to answer this main question. Finally, section 1.5 will outline the method through which the main research question will be answered.

1.1 Problem context

A sewer network is a vital part of infrastructure. The networks can be considered one of the most important components of water infrastructure (Nguyen & Seidu, 2022). It was able to reduce the risk of infectious diseases (e.g., cholera, typhus) spreading through densely populated areas (Obradović et al., 2023). This led to further expansion, slowly increasing the size and societal dependency. Nowadays, a town between 100,000 and 500,000 inhabitants contains anywhere between 1,000 and 5,000 km of sewer pipes (Salihu et al., 2022). With this reliance on the network, failures are undesirable and can cause severe economic or social damage. These possible damages have resulted in strict governmental and environmental regulations for the underground network (Fenner, n.d.).

Prior to failures, the condition of the pipes in the network starts to degrade. This degradation can decrease the performance of the pipes, creating issues such as leakage or contamination of fresh water. Monitoring the network has become more difficult in recent years. The network has expanded, without extra budget allocation in recent years (Salihu et al., 2022). Furthermore, since the network is located underground, access for rehabilitation or replacement is a costly endeavor (Bukhsh et al., 2022). This makes a poor condition assessment a costly mistake. So, the challenge has become to maximize the value derived of an asset over its lifetime against an acceptable level of risk (Tscheikner-Gratl et al., 2019).

Two main directions can support maximizing asset value. These are: 1) condition assessment, or 2) frequency of assessment (Osman et al., 2012). Condition assessment refers to the accuracy of predicting the condition of a pipe. Accurate assessment of the pipes in the network is not easy, which is a known issue in literature (Tscheikner-Gratl et al., 2019). The challenge has been researched in papers such as (Hawari et al., 2020; Nguyen & Seidu, 2022; Okwori et al., 2020). Accurate assessment is important to predict risks of failures and preventively maintain. In the case of failure, the monetary costs can be up to 10 times the life cycle costs of a sewer pipe (Obradović et al., 2023).

Alternatively, frequency of assessment refers to the number of times an action is taken based on the expected condition of pipes. This refers to either inspecting pipes to determine the condition or rehabilitating damaged pipes. While accurate prediction of conditions is important, (Elmasry et al., 2019) found that more than 50% of inspections were unnecessary. They stated the pipe was in a condition where the risk of failure was extremely low, so the inspection could have been postponed without incurring unnecessary risk. This shows that optimizing the inspection interval is still a relevant issue. Therefore, this research will focus on optimizing frequency of assessment of sewer pipes. This gives the following research problem:

Current inspection policies schedule too many inspections.

1.2. Problem analysis

With an action problem selected, a specific part of this problem can be researched. For this, A problem cluster will be used, as proposed by (Heerkens et al., n.d.). This cluster can be found in Figure 1. At the bottom of this cluster is the action problem. From this problem, the next step is examining the model for calculating the inspection interval. A possible cause to inaccurate inspection intervals, is that the assumptions are unrealistic. Most research suggestions about these assumptions were excluded from the cluster. They can be found in table 1 with their respective references. In the remaining problem cluster, two main directions remain that can be researched. Degradation is a separate research direction, so it is considered outside the scope.

With these research directions eliminated, the problem of assuming infinite resources remains. In (Dao & Zuo, 2017), two main resource constraints are considered: budget and resources. Both directions are relevant and could be researched here. Resource constraints for sewer network have already been researched in papers such as (Draude et al., 2022). Furthermore, they are more difficult to define clearly. Often, the resource constraints are denoted with a hard limit on the number of actions. Budget constraints seem to be the most promising research gap. Some papers found were in the field of nuclear reactors and multi-state serial systems, (Dao & Zuo, 2017) and (Singh et al., 2008) respectively. For sewer maintenance, the closest was (Mancuso et al., 2016). The paper considers a pipe network, where the most critical pipes were identified through a ranking system. It includes a budget but generalizes pipe inspection costs. The generalization of cost modeling seems to be a research gap in sewer network maintenance (Tscheikner-Gratl et al., 2019). This research direction is also suggested in (Bukhsh et al., 2022). Therefore, this research will focus on cost modeling and budget constraints in sewer network maintenance scheduling.

Research direction	**Mentioned in**
Reliability and quality of inspections	
Failure detection quality of inspections	Tscheikner-Gratl et al, 2019; Mohamed et al.m 2023
Failure dependency between components	De Jonge and Scarf, 2020; Xie et al, 2020
Parameter updates between failure interval	Kleiner, 2001; De Jonge and Scarf, 2020
Inclusion of economies of scale	Kleiner, 2001; M. Wang and Yin, 2022; Tscheikner-Gratl et al, 2019
Consideration of future costs	Xie et al., 2020; Draude et al., 2022; M. Wang and Yin, 2022
Readability of model results	Xie et al., 2020; Fink, 2020; Iqbal et al., 2017

Table 1 The different research directions

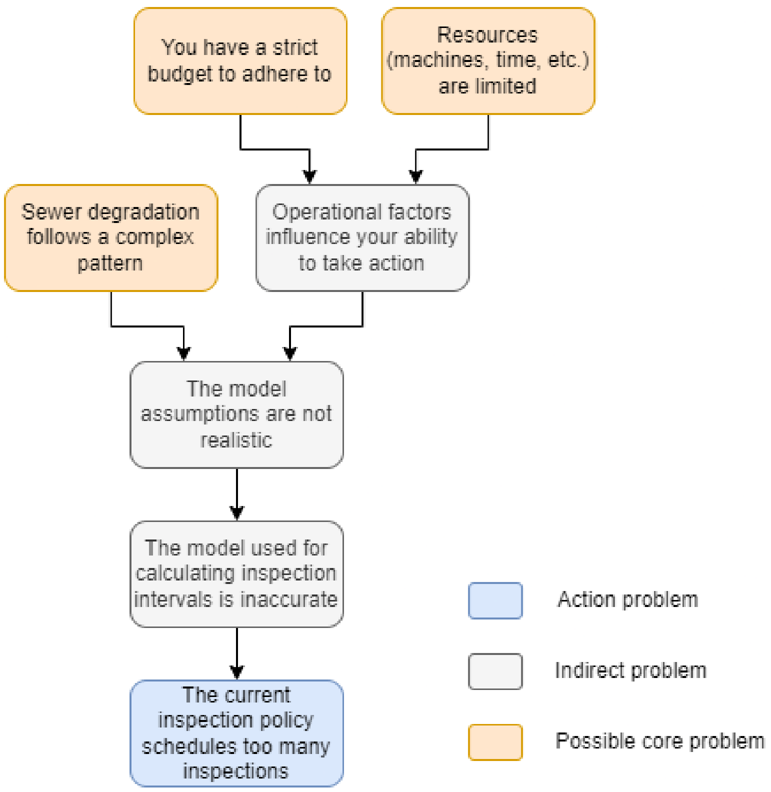


Figure 1 The problem cluster

1.3. Research goal

In this research, the main goal is to provide a model that can consider costs in sewer network maintenance. In previous research of sewer network maintenance, cost is simplified (e.g., (Bukhsh et al., 2022; Mancuso et al., 2016). The model must show how a schedule changes when costs are considered in more detail. It will be interesting to determine the schedule changes with different budget and cost configurations.

It is infeasible monitor the whole network continuously(Fan & Yu, 2022), so limits on the number of inspections and maintenance are important. You could prioritize based only on risk of failure, but not all risk should be considered equal. Some pipe failures can have a larger social impact, especially if the effects on the environment are large. Future research can benefit from understanding how ignoring costs can affect the quality of the results. The changes in the inspection interval should be interesting to monitor.

1.4. Research questions

For solving the research problem, the following main research question is defined:

To what extent do budgetary constraints for maintenance affect the determination of optimal inspection intervals of inspection decisions for sewer pipes?

This question drives the rest of the research and is answered at the end. To outline this process, four smaller research questions are defined.

1. *What information is required for modeling optimal inspection intervals and policies?*
2. *How can you evaluate the performance of inspection interval policies?*
3. *What methods can solve the model for optimal inspection policies?*

4. *How do inspection strategies change under different constraints and environments?*

Each question addresses one part of the main research question. The first question defines the required prerequisite knowledge. This knowledge contextualizes the issue as a part of a process, instead of an isolated problem. Question 2 considers the methods through which performance can be described. Performance is an ambiguous term and must be defined in the context of research. Next, question 3 asks how the performance is optimized. This defines the results of the model and how these results are interpreted. Finally, question 4 looks at the effect of changing conditions to the model. The model must be able to function under varying circumstances. It is undesirable to propose a model that only works under specific conditions.

1.5. Approach and contributions

To illustrate how the main research question is answered, a research roadmap is included in figure 2. This roadmap describes the structure of the remaining chapters of this research. First, chapter 2 establishes a theoretical framework for the rest of the research. Through a literature review, existing theories can serve to support the development of arguments in the chapters afterwards. Chapter 3 describes the sewer network. The chapter explains the context of the problem in more detail and defines the system in a formal manner. The models are described in chapter 4. These models are then tested, and the results are presented in chapter 5. Finally, the research will be concluded in chapter 6. This chapter describes the answer to the main research question and the limitations of the results. Furthermore, any promising research directions will be proposed based on the findings.

This research will provide the following:

1. A model that can determine the optimal inspection interval for sewer pipes.
2. Evaluation that shows the performance of the model compared to other scheduling methods.
3. A description detailing the model functions and applications.

The main contributions of this research will be a model that can produce an inspection schedule. The schedules from the model are tested and evaluated against other scheduling methods.

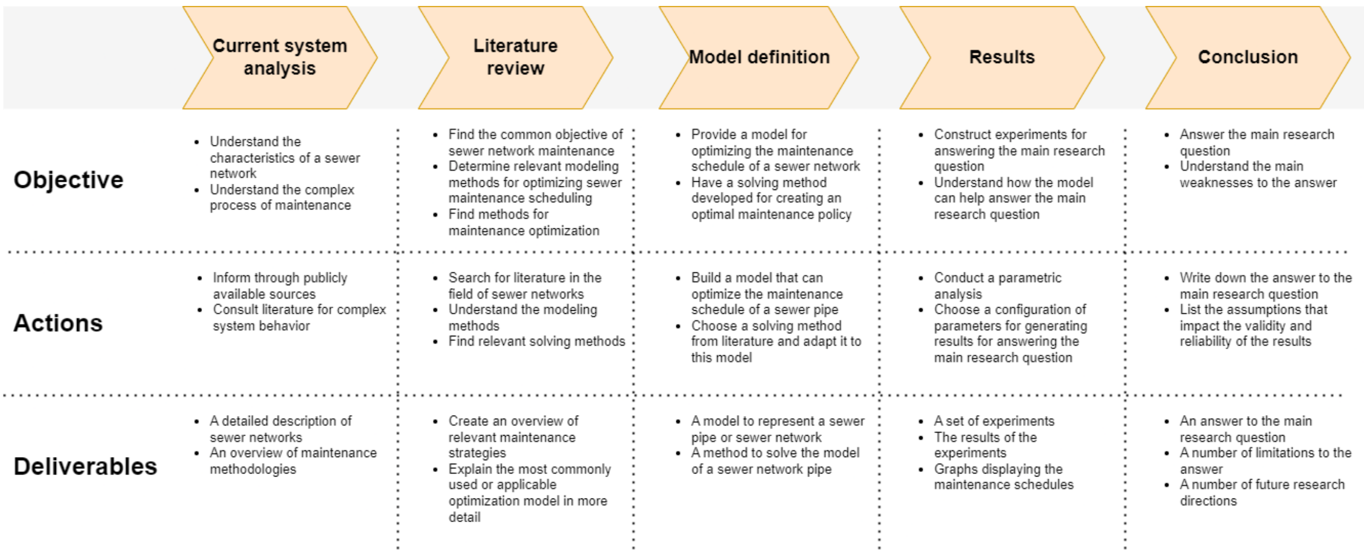


Figure 2 The research roadmap for the research

2. Current system analysis

This chapter describes the context of the problem. The goal of this chapter is to answer the first research question: What information is required for modeling optimal inspection intervals and policies? First, section 2.1 describes the network in more detail. Section 2.2 describes the degradation and failures. Then 2.3 describes the maintenance for the sewer networks. Finally, section 2.4 discusses the data quality problems in sewer networks briefly.

2.1 Sewer networks

The sewer network is an underground network of pipes, access points, and treatment plants. The goals of the network are summarized as:

- Continuous collection of wastewaters in a defined system area
- Safe transport of wastewater to treatment plants
- Necessary treatment of wastewater prior to its release into the environment

The sewer network type determines the wastewater that enters the system. A combined sewer system collects drainage from homes, industry, and stormwater. The content from this type of network must always be treated before being discharged. These networks would overflow during heavy rains, which led to separate sewer systems. The separation would allow rainwater to be discharged into the environment during heavy rains. Normal wastewater would still be sent to the treatment plant before being discharged. For this research, the focus is on the sage transportation of wastewater.

The transportation network is an important and hidden part of infrastructure. The safe transportation of wastewater prevents waterborne illnesses, making failures dangerous. Furthermore, failures prevent homes and industries from being able to safely discharge their wastewater. Industries that use a large amount of water may need to halt production when unable to use the network. Due to this societal impact of failures, governmental organizations conduct maintenance. Furthermore, failures can damage surface level infrastructure and cause environmental problems. A failed pipe can cause street and road degradation, pollution, or collapse. Long-term exposure of sewage damages the environment as well. The damage to the environment of waste is difficult to determine beforehand, due to the different types of networks. A separate system may leak only rainwater, which would naturally leak into the environment. However, industrial waste can be dangerous if it contains dangerous pollutants such as cleaning agents.

2.1.1. System interaction and configuration

The network contains multiple pipes that can interact, making the models complex. (Alaswad & Xiang, 2017) state that multi-component systems are often modeled through multiple single-unit models, ignoring dependencies between components. Furthermore, (de Jonge & Scarf, 2020) recommend modeling network systems as multi-unit systems. Alternatively, they suggest dividing the system in repairable sub-sections. Optimizing individual pipes instead of networks is common in literature, see (Anbari et al., 2017; Elmasry et al., 2019; Kleiner, 2001; Nguyen & Seidu, 2022). An extension is to first optimize on pipe level, and then optimize on network level. (Osman et al., 2012) optimizes at pipe level and adds a secondary optimization stage for the network level.

Without considering the network structure of the system, the dependencies between components are simplified. These dependencies influence the operational maintenance schedule. (de Jonge & Scarf, 2020) argue that ignoring the dependencies is not realistic but does not invalidate the results. They argue that discarding dependencies should be a conscious choice and stated as a limitation of results. There are three main forms of dependencies (de Jonge & Scarf, 2020):

1. Economic dependence: Components close to one another can be maintained or inspected together to save costs.
2. Structural dependence: Maintenance of one component requires dismantlement of another component.
3. Stochastic dependence: Failure times of one component are related to another component due to e.g. sharing of loads, common external factors, etc.

All three dependencies are relevant to optimizing sewer network maintenance. An example of modeling an economic dependency in sewer networks is found in (Bukhsh et al., 2022). Their research is focuses on an economic dependency. A combination of dependencies is not often researched.

2.1.2 Pipes

The transportation of wastewater is achieved through pipes. These pipes transport the waste using either gravity, vacuum, or pressure. The pipes are installed with one of these transportation methods in mind. Pipes feed into a tank or chamber that collects waste from multiple homes at once, that then continues transporting the collective waste. The function of the pipe is predefined and unchanged without revisions to the surrounding network. The pipes vary in aspects such as diameter, material, length, and buried depth. Multiple pipes are connected in sequence using joints to transport wastewater across larger distances. The surrounding soil of the pipe is used as a method to suspend the pipe in the correct position when it is laid. After construction, the soil is replaced and kept covered until access is required.

When the surface has infrastructure such as roads or walkways, these must be deconstructed to reach a damaged pipe. The surface is repaired when the maintenance is completed or when the pipe is replaced. The expected lifespan of surface infrastructure is usually shorter than sewer pipes. Therefore, access can be combined with surface infrastructure renovations. Furthermore, the surface infrastructure changes can lead to a change in demand. This change can be significant enough to warrant the installation of new pipes before the scheduled end of life is reached for the pipes. If the pipe does not need to be replaced preemptively, the average lifespan of a pipe can vary between 50 and 100 years.

2.2 Degradation

The pipes of the sewer network slowly degrade over time until a moment of failure. The failures are generalized into three different categories. The first is structural failure, which has three stages: 1) initial defect, 2) deterioration, and 3) collapse. The final stage is usually triggered by some event after a longer period of deterioration. Before this stage, the degradation evolves at an uneven and difficult to predict rate. The second failure category is operational failure. This category refers to debris, infiltration, intrusion, obstruction, or grease build-up. The flow of the pipe is obstructed, and the pipe is unable to fulfil its function. These operational failures are the most common types of failures in the wastewater collection network. The failures do not affect the condition of the pipe and are resolved during a maintenance procedure. The final category is hydraulic capacity failure. This occurs when the pipe does not have enough capacity for the flow of (waste-)water. A hydraulic failure is usually a sign of another failure, structural or operational, somewhere in the network.

The moment of failure is difficult to determine, as the excess water from a failure can drain through the soil instead of rising to the surface. This can lead to wastewater leaking into the soil undetected for longer periods of time. Failures are only detected during maintenance or inspections, or if the damage is significant enough to damage the surface.

Predictive degradation models have been developed to mitigate the risk of hidden failures. These predictive models estimate the risk of failure based on historical data. (Hawari et al., 2020; Salihu et al., 2022) provide good overviews of the different types of models. Statistical models are common ways to predict the failure probability. These statistical models use a probability distribution to predict the moment of failure. Simple statistical models use a degradation limit on operational or non-operational pipes (Bukhsh et al., 2022). In sewer networks, the pipe characteristics can be used in the calculation of failure behavior. In (Bukhsh et al., 2022), some characteristics such as pipe length are included to specify the degradation patterns more towards sewer network maintenance. An extension to this type of models is using a Markov chain model. The Markov chain model uses states to describe the current condition of the pipe. The pipe has a probability distribution across different states that is updated each period. This method can describe degradation in more detail than a probability distribution.

Machine learning methods have become more popular to predict failure probability as well. These models can incorporate dependence more efficiently than the statistical models. Neural networks can handle multiple variables without increasing computation or solving time too much. Furthermore, the networks can be tailored to each asset, whereas the statistical models generalize. The best example of this degradation model can be found in (Nguyen & Seidu, 2022). They split physical and environmental factors in two sets. The least important features are removed, until a remaining number of features is still able to predict sewer pipe degradation accurately.

2.3 Maintenance

The aim of maintenance is to maximize the lifetime functionality of a sewer system against the lowest costs. For reaching this goal, maintenance on the sewer network pipes has transitioned from corrective to preventive. A corrective or reactive strategy causes the system to be out of function for periods between failure and response. A reactive strategy results in low functionality against high costs of maintenance. Therefore, more preventive, or proactive methods have been introduced.

Proactive maintenance attempts to maintain a system before a failure. The sewer system is hidden, so this strategy relies on the accuracy of predictive models. An accurate predictive model can schedule maintenance against a threshold of risk. The lifetime functionality of the pipe can improve against lower costs if the strategy is well-implemented. A proactive maintenance strategy consists of three actions: inspections, rehabilitation, and replacement. A replacement occurs in both corrective and preventive maintenance. However, preventive maintenance attempts to replace before a failure has occurred. Inspecting and maintaining is unique to preventive maintenance.

Inspections are used to assess the current condition of the pipe. These inspections are considered a key component of maintenance for sewer networks. Inspections provide information about the current state of the network. Inspections can show a failure in a sewer pipe before surface damages have been able to accumulate. Furthermore, the information from inspections is used to calibrate the predictive degradation models. Overviews of the different methods of inspections can be found in (Mohamed et al., 2023; Tscheikner-Gratl et al., 2019).

Rehabilitation refers to improving the condition of the pipe. These rehabilitation methods are only able to improve one specific failure risk of the pipe. An operational failure is prevented using a technique called flushing. Here, water is shot through the pipes to flush away unwanted buildup of waste. For this, a manhole is required, but it affects pipes in an area rather than a single pipe. Too much build-up will clog the pipe, increasing the risk an operational failure. This method is not able to prevent or resolve damage through cracks or infiltration. For these structural damages, there are several actions that can improve the condition without replacement of the whole pipe. The methods can be classified in three categories: 1) reconstruction, 2) resurfacing and 3) mitigation. Reconstruction refers to replacing the damaged section of the pipe with a newer section. By covering the inside of the pipe, the lifespan of the pipe can be extended. The second category uses a thermosetting resin or mesh that covers the inside of the pipe to slow further degradation. This requires cleaning of the inside of pipe, since the material must adhere to the old structure. Thirdly, damage can be mitigated by preventing the pipe from eroding the supporting soil. Here, a gel-like substance in injected into the soil to cover the damaged area around the pipe. This maintenance is a non-structural rehabilitation, where the structure of the pipe is still poor, but the damage is being prevented. This final method is only applicable with an initial defect or small deterioration.

2.4 Data quality in sewer maintenance

Data is required for determining the parameters of the degradation model. In sewer networks, this data suffers from poor quality of the data. As described in (Tscheikner-Gratl et al., 2019), the data is prone to selective survival bias, recruitment bias, and information censoring. Survival bias refers to the data prioritizing the pipes that have survived until the moment of inspection. Consequently, the models are expected to underestimate the current state of the network. This leads to the models overestimating the pipe lifetime. This type of bias is most crucial to the development of predictive degradation models. This bias can be difficult to establish in a dataset and can therefore be overlooked more easily than recruitment bias or information censoring.

Most of the data issues are problems for designing the degradation model. It is therefore important that the results can perform under some level of uncertainty. Bias in data can be difficult to prove and does have to follow any pattern. Therefore, a schedule from one model may not be close to the optimal schedule due to the data discrepancy. Future research may be able to mitigate data quality issues or further improve accuracies. It would be an interesting research topic for research focused on the degradation model.

2.5 Conclusion

In this chapter, the general structure of a sewer network is analyzed. The pipes have high failure costs compared to inspection and maintenance cost, so preventive maintenance is preferred. The sewer pipes do not have a self-announcing failure, but failures that can remain hidden for some time. A structural failure is a gradual process. An operational failure can be prevented but is difficult to predict as well. Inspections are the only way these failures can be detected, after which the pipe should be rehabilitated or replaced. Ignoring system dependencies does not invalidate results, but the decision should be made before modeling. Furthermore, there are different damages through which the pipes degrade. This degradation is simulated through one of the three categories:

statistical models, Markov Chains, and machine learning. There are different inspection methods and maintenance types. These have a different effect on the change of condition of the pipe.

3. Literature review

This chapter provides a theoretical background for the rest of the research. First, section 3.1 looks at different maintenance strategies that are used in literature. Furthermore, this section will list common inspection policies in related literature. Next, section 3.2 examines modeling methods that are used in examining these inspections policies. This section will focus on the method that will be used for this optimization model as well. Finally, section 3.3 provides an overview the applicable solving methods.

3.1. Inspection strategies

Inspection schedules are usually constructed in one of three categories: 1) continuous monitoring, 2) periodic-, and 3) non-periodic schedules (Alaswad & Xiang, 2017). Continuous monitoring requires either sensors or constant access to the asset (Alaswad & Xiang, 2017). The sewer network is too large to install sensors, and constant access is unreasonable. Therefore, this strategy is not considered here. Periodic schedules are schedules that build a single interval between two inspections. This interval does not change. Non-periodic schedules can be further divided into two sub-categories: static or dynamic(de Jonge & Scarf, 2020). Static aperiodic intervals are constructed beforehand and do not change based on the results of the first inspection. Dynamic schedules change based on the results of the first inspection. According to (Kleiner, 2001), an optimal schedule should be change based on the results of each inspection.

By considering the inspection strategies, this research can contribute to determining the loss of performance with simpler inspection strategies. This performance gap can show the performance that can be gained by expanding the maintenance scheduling policy.

3.2.1. Periodic inspections

Periodic inspections are commonly used in complex production systems and crucial infrastructure to verify functional condition and safety (Phan & Zhu, 2015). However, few papers consider a periodic inspection schedule without some dynamic rescheduling or intervention. Periodic inspections are more cost-effective than continuous monitoring, but risk higher failure costs because of uncertainty (Alaswad & Xiang, 2017). In general, these schedules are easier to implement than non-periodic inspections, because the interval remains constant once established. This type of schedule can be found in sewer network optimization. (Osman et al., 2012) optimizes the interval length, by considering costs of maintenance and failure risk. They construct static policy without resource constraints. (Phan & Zhu, 2015) define clusters of geo-distributed assets and optimize the inspection policy for each cluster. As a result, each cluster has its own optimized static inspection interval. (Babishin & Taghipour, 2016) construct an interval per component of a multicomponent system and include opportunistic maintenance as an opportunity. Furthermore, their interval considers both hard and soft failures.

3.2.2. Static aperiodic inspections

Non-periodic inspections can lead to potential cost savings, due to less inspections in the beginning of the components' life (Alaswad & Xiang, 2017). However, more documentation is required for the increased complexity of the schedule. Furthermore, there is a higher chance of rescheduling work and effect of human error is higher. These complexities can make periodic schedules preferable.

The interpretation of this scheduling method is different across literature. In (W. Wang, 2000), two intervals are defined: one for new components, and a second for all subsequent inspections. Furthermore, the inspections are simultaneously optimized with a failure threshold. This method could perform well in sewer network maintenance, due to the long lifetime and degradation uncertainty. (Liu et al., 2018) optimizes the number of inspections together with the times at which the inspections take place for natural gas pipelines. They provide a complete non-periodic schedule for a pipeline. These pipelines degrade more consistently than sewer pipelines, due to less varying contents. (Fauriat & Zio, 2020) does the same, but instead by testing with value of information. The result in a predefined number of inspections at predefined times. Their model is generalized to condition-based maintenance, which would make their method applicable here.

Another different is found in (Lin et al., 2015), who propose three different models depending on the scenario. The models share two optimization criteria: 1) a reliability threshold and 2) an interval length based on that reliability. The paper considers a maintenance intervention always a repair rather than an ability to gain information. This makes the second interval almost identical, since repaired components are always repaired to the same state. Alternatively, (Hajipour & Taghipour, 2016) proposes a schedule fully optimized based on expected costs and item life cycle. They define costs and optimize based on costs alone, with downtime penalties to compensate for reliability. The result is a schedule reporting the months in which inspections are optimal. Finally, the paper of (Kim et al., 2022) uses the balance between epistemic and aleatory uncertainty to schedule. They define an uncertainty threshold and as soon as the epistemic uncertainty becomes larger than the threshold, an inspection should be scheduled. The result is a sequence of inspection times that increases with uncertainty.

No inspection policies were found in this category specifically for sewer network maintenance.

3.2.3. Dynamic aperiodic inspections

The most complex schedule is the dynamic aperiodic scheduling method. This requires more documentation and is more prone to errors than the periodic and static aperiodic scheduling method. Research argues that decisions should be made fully on the current condition of the system, so a predetermined schedule will always be suboptimal. (Abubakirov et al., 2020; Kleiner, 2001) apply this strategy by only considering the first maintenance instance. This instance is optimized, so the schedule should be rebuilt when this time is reached. They consider buried oil pipelines and large infrastructure assets respectively. Buried oil pipelines are again comparable but have more predictable contents. Therefore, only external inspections may be necessary. Large infrastructure is broad and can refer to buried pipelines or buildings, which makes assessing applicability difficult.

Other papers use prioritization the rather than considering an inspection interval. Instead of scheduling ahead of time, this strategy ranks components based on risk of failure. The components with the highest risk of failure will be inspected. This method is used in maintenance of sewer networks as well. (Elmasry et al., 2019) ranks pipes based on operational factors and pipe failure risk. The result is a list of components to inspect. Similarly, (Anbari et al., 2017) ranks pipes based on the risk- and consequence of failure as well. They pose the decision is for the stakeholder to make. (Nguyen & Seidu, 2022) models failure behavior, and provides a ranking of pipes which are most likely to be in a poor condition. They do not consider possible consequences of failure or operational factors, however. Rankings for sewer pipeline maintenance may not be applicable in research where the inspection interval is considered with respect to resource constraints.

Then, (Mancuso et al., 2016) builds different inspection portfolios and suggests the stakeholder should pick which they would prefer. Each portfolio is a possible inspection schedule based on the risk- and costs of disruptions of sewer pipelines. Finally, (Draude et al., 2022) optimizes based on the crew locations and travel distances. The result is a schedule with a score per pipe, and the expected costs of travel and maintenance. This approach does not optimize inspection intervals, but rather optimizes the number of inspections instead. Although both approaches are interesting, they do not directly relate to the main research question.

Dynamic aperiodic scheduling seems the preferred strategy in sewer network maintenance. The reason may be the uncertainty of the life cycle of buried pipes. A schedule can become obsolete if repairs are required. Therefore, considering the entire schedule instead of the next instance of maintenance is unnecessary. There are numerous methods to present the optimal maintenance strategy. However, considering the main research question, the most appropriate method may be a traditional schedule. This schedule presents the future maintenance actions and times of those actions.

3.2 Optimization objectives

Optimal policies are determined by the objective of a model. Optimization of policies is always performed by minimizing or maximizing an objective function (Van Horenbeek et al., 2010). The objective of this function can be split into single-objective and multi-objective optimization. In general, single-objective optimization is more thoroughly researched than multi-objective optimization, while multi-objective optimization may be more representative (Van Horenbeek et al., 2010). Multi-objective optimization introduces complexity to the model, making solving more difficult. Furthermore, the multi-optimization methods face criticism by making the results more uncertain and subjective (Ding & Kamaruddin, 2015).

The optimization objective can be divided into three categories: costs, availability, or reliability (Van Horenbeek et al., 2010). Another category is safety; however, this objective is mostly used in areas where loss of life is a consequence of failure (Alaswad & Xiang, 2017). In sewer networks, reliability and costs are important to consider. These objectives can be broken down into more concrete goals, usually specific to a model. Examples are minimization of long-run costs, minimization of total (discounted) costs with a finite horizon, maximization of availability, and maximization of reliability during subsequent activities (de Jonge & Scarf, 2020). In pipeline network maintenance, costs are commonly used as an optimization objective. In (Kleiner, 2001; Osman et al., 2012), total costs are reduced to costs across a finite horizon and used as optimization criterion. (Liu et al., 2018) provides a more complex cost optimization model, where costs of multi-level decision making are optimized. Alternatively, costs and probability of failure can be combined to simplify multi-objective optimization (Abubakirov et al., 2020) (Anbari et al., 2017). A condition index can be used as an indication of the performance of certain schedules (Marzouk & Omar, 2013). This allows for a partial multi-objective optimization. Alternatively, (Quatrini et al., 2020) suggested usage of a feasibility criteria. This feasibility index is used to indicate the subjectivity of the schedule.

3.2.1 Inspection value

Inspections yield no return in terms of value. Therefore, researchers have developed alternative methods of measuring the value of inspections to incentivize models to schedule inspections. For example, (Elmasry et al., 2019) minimized costs and total time spent inspecting, while maximizing the number of inspected pipelines. This incentivizes the model to schedule inspections while minimizing the expected costs. In (Babishin & Taghipour, 2016; Hajipour & Taghipour, 2016), soft failures generate a penalty for each period it is remains undetected. Therefore, inspections prevent further penalties from occuring. Furthermore, they include the option to inspect after a hard failure, so opportunistic inspections are cheap. (Osman et al., 2012) minimizes the sum of inspection costs and costs of imperfect information. These imperfect costs of information are the failure probability matrix multiplied with the error matrix. The result is a value of information (VoI) matrix. If VoI is higher than inspection costs, inspections are beneficial to schedule. This VoI occurs in other scheduling research as well. In (Fauriat & Zio, 2020), the difference between an action without knowledge and with knowledge is defined as the VoI. The time at which the VoI is the highest, is the optimal moment to schedule an inspection. A negative VoI means that no inspection is valuable enough. (Abubakirov et al., 2020) values inspections through risk- and cost of inspections. A utility function is constructed, and the lowest utility will provide the best moment for an inspection. A strictly increasing utility function means no inspection is the best decision.

3.3 Modeling methods

For optimizing the maintenance schedules, a representation of the system is required. This representation should define the environment in which the objective is optimized. The objective and strategy should be defined before constructing and testing the model (Van Horenbeek et al., 2010). Maintenance scheduling in sewer networks has primarily focused on minimizing risk. Researchers such as (Tscheikner-Gratl et al., 2019) acknowledge the need for more sophisticated scheduling methods. Therefore, previous research does not use universal modeling methods that can be translated into universal models. These models instead establish an objective and optimize based in this objective.

3.3.1 Modeling environment

The environment of the model refers to the representation of time, deterministic or probabilistic models, planning horizon, and constraints. As stated by (Van Horenbeek et al., 2010), general modeling techniques is about the choices you must make. These are choices that you must acknowledge and support but are subjective to your model. The most important choices, mentioned in (Van Horenbeek et al., 2010), will be discussed here.

First, discrete time seems to be used often in sewer maintenance, e.g. (Kleiner, 2001; Osman et al., 2012). Considering the expected lifetime of pipes can reach 80 years (Tscheikner-Gratl et al., 2019), discrete time is not unreasonable to assume. Secondly, both deterministic and probabilistic models were found. (Elmasry et al., 2019) optimizes a deterministic model, but (Bukhsh et al., 2022) optimizes a probabilistic one. It is difficult to say which model would perform better. Next is the planning horizon. Most models used a finite planning horizon, such as (Abubakirov et al., 2020; Elmasry et al., 2019; Kleiner, 2001; Mancuso et al., 2016; Osman et al., 2012). One possible reason could again be the long-expected lifetime of pipes. Finally, there were both constrained and unconstrained models, e.g., (Elmasry et al., 2019; Osman et al., 2012) respectively.

No research was found that argued for a certain modeling method for objective and environment. Static decision-making was in general less common in pipe networks, so the method should be able to optimize dynamically or online. For optimizing reliability, Bayesian Networks (BN) have been used. This is a probabilistic, graphical model representing the conditional dependencies between failure root causes and symptoms (Shafiee & Sørensen, 2019). The model is useful to examine fault prediction, but less applicable for optimizing costs (Abubakirov et al., 2020). A Markov Decision Process (MDP) had been used in previous research when attempting to optimize costs (Bukhsh et al., 2022). Furthermore, (M. Wang & Yin, 2022) finds that the generalized partially observable Markov decision process (POMDP) has been used in pipe infrastructure.

3.3.2 The POMDP

The MDP is a model of a fully observable sequential decision processes (Braziunas, 2003). A valid infinite horizon MDP model must consist of states, actions, reward or cost function, and transition probability (Puterman, n.d.). Infinite MDPs are described as tuples using $M = (S, A, P, R, \gamma)$. Here:

- S = the set of states.
- A = the set of actions,
- P = the set transition probabilities.
- R = the set of reward functions
- γ = the discount factor (not included in some notations)

This tuple should also include the number of decision epochs for a finite horizon MDP (Puterman, n.d.). The goal of this model is to find a policy that maps the states to actions that result in the highest expected reward (Braziunas, 2003). The policy is defined by optimizing a value function, which is generally described as:

$$V^{\pi}(s) = \sum_{a} \pi(a|s) * \sum_{s',r} p(s', r|s, a) * [r + \gamma V^{\pi}(s')] \quad for\ all\ s \in S$$

This formula uses the Bellman notation (Sutton & Barto, 2018). The formula determines the value of being in a state s and by following policy π. Both the immediate reward and the continued reward are considered by considering the probability of moving from one state to another. The optimal policy π^* is where the value is maximized for all states. This function is given with:

$$V^{\pi^*}(s_t) = \max_{a \in A} R_t(s_t, a_t) + \gamma \sum_{s' \in S} P(s'|s, a) * V^{\pi^*}(s_{t+1}, a_t)$$

The optimal policy chooses the action where the current reward and expected future reward are maximized. The optimal policies can be defined by the value function. An infinite horizon greedy policy can be described by (Braziunas, 2003):

$$\pi^* = \underset{a}{\operatorname{argmax}}\, V^{\pi^*}(s, a)$$

A weakness of this policy is that is must be updated accordingly with the history of observations (Pack Kaelbling et al., 1998). Therefore, the computation of a complete policy can be time intensive.

Alternatively, a stochastic policy would describe a distribution over all actions instead (Braziunas, 2003). Water pipe optimization has seen use of this model in (Bukhsh et al., 2022) to combine buried pipe scheduling and clustering maintenance. These networks have self-announcing failure instead of hidden failures. A limitation of the MDP is that the agent requires full knowledge of the state at every decision epoch (KAEBLING). Therefore, the MDP needs to be extended to include hidden states and observational probabilities to be applicable to sewer network maintenance.

The POMDP model includes uncertainty about systems where the state is only partially observable (Braziunas, 2003). POMDPs include uncertainty about the current state of the system and probabilities of observations. The formal MDP description is extended with $M = \{S, A, T, R, \Omega, O, \gamma\}$ (Pack Kaelbling et al., 1998). Here, the set of observations and observational probabilities are described by Ω and O respectively. To incorporate these new uncertainties, the value of a belief is taken instead of a state, since the real state of the system is hidden. To determine the value of the optimal policy of starting in a belief b, the formula becomes:

$$V^{\pi^*}(b) = \max_{a \in A} \left[\sum_{s \in S} R(s,a) * b(s) + \gamma * \sum_{o \in O} P(o|b,a) * V^{\pi^*}(b_a^o) \right]$$

Again, the formula follows the Bellman notation (Braziunas, 2003). This formula now includes a vector of beliefs of being in a state s. Furthermore, the continued rewards are multiplied by the conditional probability of observing o, given belief b and action a. Using this in the policy formula, a greedy policy can be stated as:

$$\pi^*(b) = \underset{a \in A}{\operatorname{argmax}} \left[\sum_{s \in S} R(s,a) * b(s) + \sum_{o \in O} P(o|b,a) * V^{\pi^*}(b_a^o) \right]$$

The policy extends the current reward to include the probability of occupying state s. Furthermore, the observational probabilities are included in the expected future rewards. The formula still has the same goal, but the hidden states are modeled explicitly. The formula rewards a system where the probabilities of being in a good state are high.

Sewer network maintenance has seen use of the POMDPs. (Osman et al., 2012) models a sewer network as a POMDP to determine the methods for inspection and the optimal static inspection policy. They include different inspection reliabilities per method of inspection. A limitation of the model is that the policy is assumed static, and that the solution is determined through a genetic algorithm. Therefore, the quality of their solution is uncertain. Furthermore, section 3.2.1 explains that periodic inspections are unable to perform as well, due to the high level of simplification.

3.3.3 Model solving methods

POMDPs have high computational costs since the history of observations affect the optimal policy. An exact solution is difficult to calculate even with small state and observation spaces (Braziunas, 2003), which means approximate methods are necessary. Complex algorithms for solving POMDPs exactly are suggested in (Pack Kaelbling et al., 1998) and (Braziunas, 2003). These algorithms use pruning strategies to find a set of value vectors for which the total rewards are maximized.

Approximate methods are unable to determine the performance of results, but they are constructed more quickly and in reasonable time than exact solutions. In the reviews of (Janga Reddy & Nagesh Kumar, 2020; Kumar & Yadav, 2022), they mostly suggest using (meta-)heuristics, with an emphasis on evolutionary algorithms (EA). Of the EA, the genetic algorithm (GA) has already seen use in pipe network maintenance optimization. It is used in papers such as (Marzouk & Omar, 2013; Osman et al., 2012). The GA optimizes a schedule by utilizing a survival of the fittest idea from nature. A population of unique solutions is defined, and the fittest solutions can produce a new population through exchanging small parts, called genes, of solutions. This new population repeats this process until some termination property is met. The fitness of the individual solutions is determined by the objective function.

Other research areas provide with additional methods (Shafiee & Sørensen, 2019). One category is called operation research techniques. This category mainly consists of e.g. linear programming, non-linear programming, and dynamic programming. These have seen use in sewer networks as well. (Elmasry et al., 2019) uses a mixed-integer linear program in pipe network maintenance. (de Jonge & Scarf, 2020) states that approximate dynamic programming may be appropriate for (complex) multi-component systems. Approximate dynamic programming is another term for reinforcement learning (RL), according to (Sutton & Barto, 2018). One example can already be found in (Bukhsh et al., 2022), who use deep RL (DRL). RL is a form of machine learning, where an agent interacts with the environment to establish the next action based on its current state. The method can find good policies, even in partially observable environments (Sutton & Barto, 2018).

3.3.4 Maintenance effectiveness

The maintenance effectiveness is not 100% in sewer network maintenance (Tscheikner-Gratl et al., 2019). Although perfect information can be assumed, some form of error is realistic to include in the model environment. An operator can misjudge the condition, leading to wrong asset information. Research has included maintenance effectiveness in different forms. In (Osman et al., 2012), this uncertainty is a constant error for each inspection method. Therefore, the uncertainty increases linearly the longer inspections are postponed. (Mancuso et al., 2016) assigns weights to certain degradation indicators, where lower weights were assigned to uncertain indicators. Thus, uncertainty is somewhat mitigated by increasing the reliance on reliable indicators. (Fauriat & Zio, 2020) implements imperfect inspection with an error that is normally distributed. (Abubakirov et al., 2020; Kleiner, 2001; Liu et al., 2018) and assume perfect information on the other hand.

3.4 Conclusion

This review establishes that optimal maintenance strategies for sewer network optimization require online optimization over a static or blanket policy. Static policies are unlikely to perform as well but are easier to implement. The objective of the model must be defined before construction and is usually defined through optimization of costs or reliability in sewer networks. This objective is defined as a function in an environment. The environment has a predefined number of assumptions about definition of time, stochasticity, horizon, and constraints. Solving the objective in the environment is computationally difficult, and usually done with approximate methods.

4. Model description

This chapter describes the model that will represent the environment of the pipes. Section 4.1 starts by defining the degradation model. Section 4.2 describes the optimization model and the goal for determining an optimal policy. This section also models the environment as a POMDP specifically. Afterwards, section 4.3 provides a simple example of the model. Then, section 4.4 defines the POMPD solving methods. Finally, section 4.5 will define a more straightforward model for determining the schedule and performance.

4.1 Degradation model

A degradation model is a model that simulates the damage that occurs over time to an item or asset, in this case a pipe. The literature divided the models as statistical, Markov chains, and machine learning models. In this research, the degradation is modeled as a homogeneous discrete time Markov Chain (DTMC). The statistical models are considered too simple to accurately model degradation. The machine learning models are not yet researched thoroughly enough, so using these models may be a risk to accuracy. Therefore, the best choice is to use Markov chains.

4.1.1 Markov Chain description

A homogeneous DTMC is a memoryless stochastic process. These processes are independent of the previous states. Therefore, the expected degradation of a system can be fully described by the current state and the degradation matrix. The degradation matrix $\boldsymbol{P}$ of a DTMC is a symmetrical matrix of size $K \times K$, where the K refers to the number of states in the Markov chain. A MC uses probabilities to model the degradation between states. A system degrades in each load cycle until the failed state has been reached. From this failed state, the condition of the system is unchanged without some intervention such as a repair. Since the states are discrete levels of degradation, the degradation can be insignificant, such that the state does not change across load cycles. The probabilities of moving between states are described by values $0 \le p_{ij} \le 1$ with $i, j \in K$.

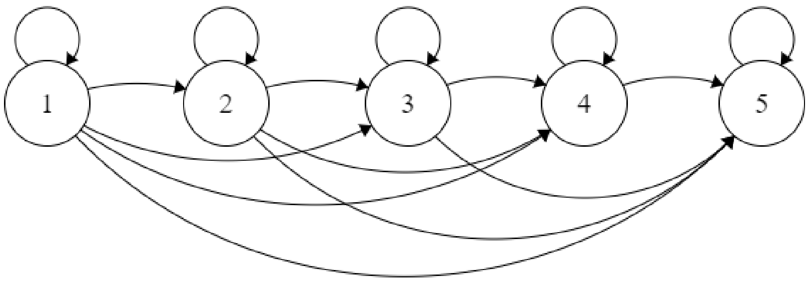


Figure 3 A visual representation of a Markov Chain

This research divides the degradation into $K = 5$ discrete states. In these states, state 1 describes a system that is as good as new. State 5 is a failed state and is absorbing, meaning a pipe in this state cannot move to another state unaided. This model can be found in figure 3. In the figure, each arrow describes a transition probability p_{ij}. The figure also shows that state 5 has no transition probabilities moving to other states, but only a recursive probability. Using this model, the probability distribution across states after any number of periods can be calculated. For calculating the state probabilities t periods after current time n yields:

$$p_t = p_n * \prod_{j=n}^{t-1} P_j$$

In the formula, $\boldsymbol{p_t}$ and $\boldsymbol{p_n}$ are vectors with K elements, describing the probability of the system being in a state. At time t, the system must be in one of the K states. It is impossible for the system to occupy two states at the same time. In observable systems, decisions are based on the current state and the transition probability.

The degradation matrix for this model is:

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ 0 & p_{22} & p_{23} & p_{24} & p_{25} \\ 0 & 0 & p_{33} & p_{34} & p_{35} \\ 0 & 0 & 0 & p_{44} & p_{45} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4.1.2 Example

To illustrate how the degradation model works, an example will be provided. A degradation matrix is provided below, with random values. In every degradation matrix, the sum of values on the horizontal axes must always be 1. Similarly, values of moving up in states are all 0.

$$\boldsymbol{P} = \begin{bmatrix} 0.9 & 0.05 & 0.05 & 0 & 0 \\ 0 & 0.8 & 0.1 & 0.05 & 0.05 \\ 0 & 0 & 0.85 & 0.1 & 0.05 \\ 0 & 0 & 0 & 0.7 & 0.3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Consider a pipe that has just been installed. Since no load cycle has transpired yet, the probability of the pipe being in state 1 is 100%. Therefore, the state probability vector is: $\boldsymbol{p}_n = [1,0,0,0,0]$**.** The sum of probabilities in this vector must add up to 1. Using the matrix above, we can calculate the probabilities of the pipe being in a state after some time has passed. After one period, the probabilities are: $\boldsymbol{p}_{n+1} = \boldsymbol{p}_n * \prod_{j=0}^{1} \boldsymbol{P}_j = [0.9\ ;\ 0.05\ ;\ 0.05\ ;\ 0\ ;\ 0]$. After 10 periods, the distribution becomes: $\boldsymbol{p}_{n+10} = \boldsymbol{p}_n * \prod_{j=0}^{9} \boldsymbol{P}_j = [0.39\ ; 0.13\ ; 0.21\ ; 0.08\ ; 0.19]$. This vector is also known as a belief vector, since it describes the belief of being in any state at a given time. The belief vector states that a new pipe has a 39% probability of being in state 1 after 10 load cycles, and a 13% probability of being in state 2, etc. With an infinite number of samples, this vector describes the average distribution of states among the samples.

Without observing the system, the belief vector is the only indication of the state. Maintenance decisions are based on the probability of systems being in one of the five states. With an inspection, the state will become known, and the schedule updates to incorporate the new information. Until the inspection has been

4.2 Optimization model

The optimization model will use the degradation model to estimate the state of the pipe and choose what time you should maintain. The goal of the model is to find a policy for which the expected costs across a finite horizon are minimized. The model is designed around the theory of value of information and POMDPs. One problem with POMDPs is that the policy can become difficult to define, due to the high state and observation space. Additionally, the belief states are continuous, which impact the computation time of the solutions.

For convenience, the notation across the rest of the chapter will remain consistent. The notation and meaning of each parameter is listed in table 2.

Notation	Type	Meaning
$\boldsymbol{c^p}$	Vector	Penalty cost of being in a state $s \in S$
$\boldsymbol{c^I}$	Parameter	Cost of inspection
$\boldsymbol{c^R}$	Parameter	Cost of repair
$\boldsymbol{c^{EM}}$	Parameter	Cost of emergency repair
$\boldsymbol{P}$	Matrix	Transition probability matrix
$\boldsymbol{b^t}$	Vector	The state probabilities of a pipe at time t
$\boldsymbol{T}$	Parameter	The end of the scheduling horizon
$\boldsymbol{I_{max}}$	Parameter	The maximum number of inspections
$\boldsymbol{R_{max}}$	Parameter	The maximum number of repair instances
$\boldsymbol{B_{max}}$	Parameter	The maximum budget for the horizon
$\boldsymbol{A}$	Set	The set of possible actions: {'Do nothing','maintain', 'Inspect'}
$\boldsymbol{S}$	Set	The set of states, with indicator s
$\boldsymbol{x_t}$	Decision variable	Binary variable where 0 is do nothing or maintain and 1 is inspect
$\boldsymbol{y_t}$	Decision variable	Binary variable where 0 is do nothing or inspect and 1 is maintain
$\boldsymbol{\delta}$	Parameter	The discount factor

Table 2 The notation for all parameters and variables

4.2.1 Model description and goal

The goal of the model is to minimize the expected costs of a buried pipe. For this goal, the model should return a policy that minimizes the sum of the expected costs that are incurred by a pipe across its lifetime. These costs are split into the expected failure costs and the maintenance costs. Failure costs are incurred when a pipe fails before its scheduled end of life. Since failures are hidden until detection, failure costs are described as a penalty costs instead. These penalty costs describe the reduce performance of failed pipes and the risk of additional damages to nearby infrastructure. This infrastructure includes pipes and joints, but also describes the damage to walkways, roads, and buildings. The penalty costs are incurred every period that a failure remains undetected. The maintenance costs describe the sum of all incurred maintenance costs. These maintenance costs refer to inspections and repairs. Preventive replacement costs are not included in this model.

To prevent failures, a pipe can be inspected to determine the current state or repaired to improve the current state. Repairing the pipe will always improve the condition to state 2. If the pipe is already in state 2 or in state 1, there are no improvements that can be made. Even though no repairs can be made, the repair costs are still incurred. An inspection does not change the state of the pipe but will show the current state. When a failure is detected through inspection, the pipe must be repaired. With scheduled maintenance, the resources are already on location and no further costs are incurred for a failed pipe. If a failure is detected through an inspection, the pipe must be repaired through emergency repairs. These emergency repairs cover all additional costs, such as rerouting existing resources, repairing nearby infrastructure, etc. Therefore, the emergency repair costs are an additional amount of costs that can be incurred during the inspection. Furthermore, since the maintenance of the pipe is unexpected, these costs are assumed to be greater than regular maintenance costs.

Since a POMDP policy depends on the history of observation, a full schedule cannot be constructed without knowledge of maintenance outcomes. The history of observation changes with a new observation. Therefore, the policy must be specified around the current time and belief of a pipe. Whenever a new observation is acquired, the policy must be reconstructed based on the received information.

4.2.2 Assumptions

Some assumptions are required for the model. These are:

1) The planning horizon is static and finite.
2) The demand and degradation remain constant across the entire horizon.
3) The maintenance strategy does not deviate from the optimal schedule.
4) Failures are only detected through inspections or maintenance.
5) Any dependence between pipes is ignored.
6) Costs are constant across the horizon.

Firstly, the model will be constructed around a finite planning horizon. Therefore, a finite planning horizon is required. The model assumes that the network has been constructed with a life expectancy and that preventive replacements are undesirable. In these horizons, the assumption must be made that the degradation or demand of pipes does not change as well. Changes to the expected degradation change the optimal schedule, thus the degradation is assumed static. Furthermore, the model assumes that its advice is followed. A future policy cannot account for unexpected deviations from the optimal schedule. The final solution space would become much larger than reasonable. The fourth assumption also limits the solution space. With this assumption, disruptions to the decision tree only occur on predefined moments in time. Assumption five refers to the dependencies between pipes. For this model, these dependencies are ignored. The economic, structural, and stochastic dependencies are not considered in this model. The final assumption assumes that the costs do not change across the scheduling horizon.

4.2.3 POMDP

An infinite POMDP is defined through the tuple of $M = \{S, A, O, T, \Omega, R, b_1, \gamma\}$. The states are defined by the degradation model. The states are discrete steps in degradation from 1 to 5. However, since the main research question is searching for resource constrained maintenance schedules, the remaining resources at time t must be included in the state definition as well. Furthermore, the current time t is included as well. The current time influences the decision; therefore, it is an important part of the state definition. This gives a state vector of $S = [\boldsymbol{b}, t, k]$. $\boldsymbol{b}$ describes the belief vector, t the current time, and k the budget remaining. The current time and resources remaining are a deterministic aspect of the model. Both are fully observable at the start of load cycle. Belief is the non-observable part of the model.

The action space is determined by the available actions at time t. With budget available, the available actions become: $A = \{Do\ Nothing, Maintaing, Inpsect\}$. As stated, the goal of the model is to optimize a schedule for a finite horizon problem without considering preventive replacements. These actions have a limit, that will be defined either through a combined budget, or a hard limit on the number of actions. The action inspect will only consider one inspection method, so this is only one action.

Observation space O describes the observations. The observation space contains all the possible observations that the model can make at any time. Due to assumption 4, the system only receives different observations during an inspection or maintenance. When the model does not take an action, the observation will always be one load cycle of the degradation process and is therefore completely predictable. This space is defined by observation function Ω, which is a matrix that determines the accuracy of the current state of the pipe. This matrix is the same size as the transition probability matrix from the degradation model, except this matrix depicts the probable errors of determining the state of the pipe. In the case of multiple inspection methods, this observation matrix differs between inspection methods. In the case of perfect inspections, the diagonal of the matrix is 1, since there is no observation uncertainty included. This matrix is defined for each action.

The reward function R describes the rewards that the model receives based on the current state and the current action. Across the stochastic horizon, the sum of these rewards should be minimized to determine the optimal schedule. There are four rewards that the model can incur based on the actions.

- $R(s_t, Do\ Nothing) = \sum_{s \in S} c^P(s)$
- $R(s_t, Maintain) = c^R + \sum_{s \in S} c^P(s) \quad \forall s \in S$
- $R(s_t, Inspect) = c^I + \sum_{s \in S} c^P(s) \quad s \in [1,2,3,4]$
- $R(5, Inspect) = c^I + c^{EM} + c^P(5)$

Note that the penalties are incurred at the start of any period. Due to this, each action incurs penalties in addition to any maintenance costs.

The set of transition functions T is defined for the space $S \times A \times S$. The probability of the transition functions is denoted as $T(s, a, s') = P(s'|s, a)$. After each action, the transition functions apply one load cycle of the degradation model. Maintaining the pipe before the load cycle will change poor states to state 2 with a probability of 100%. If the pipe is in state 1, the pipe is guaranteed to remain in state 1 as well. An inspection will only affect the state of a failed pipe, improving it to state 2. Otherwise, the state will remain constant.

Finally, b_1 describes the initial belief state. This is the state in which the model starts. γ is the discount factor. This factor influences the amount of value that is attached to future rewards.

4.2.4 Objective function

The costs consist of the failure costs and the maintenance costs. From literature, the value and policy functions of infinite POMDPs had been described. Since this model uses a finite horizon, the definition is extended to include the current decision epoch. The goal of the model is to find the policy that can minimize these average discounted costs. Using the formulae from the literature review, the description of the value function becomes:

$$V^{\pi^*}(t, b) = \min_{a \in A} \sum_{s \in S} R(s, a) * b(s) + \gamma \sum_{o \in O} P(b_a^o | b, a) * V^{\pi}(t+1, b_a^o)$$

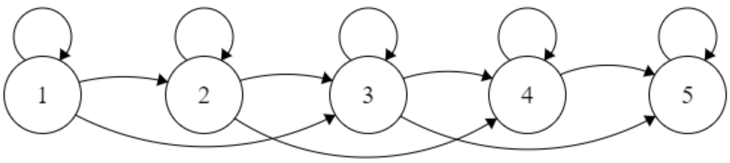
This value function describes the value of the optimal action in time t. Since the goal of the model is to minimize costs, the goal is set to minimize instead of maximize over all possible actions. Since the state of the system is hidden, the current rewards must be multiplied by the belief $b(s)$. Additionally, the value of the next epoch is multiplied by the observation probabilities respective of each belief. For the optimal policy π^*, the function becomes:

$$\pi^*(t,b) = \underset{a\in A}{\text{argmin}}\left[\sum_{s\in S} R(s,a) * b(s) + \gamma \sum_{o\in O} P(o|b,a) * V^{\pi^*}(t+1, b_a^o)\right]$$

This policy formula will be used for constructing the policy π^* at a time t. The policy formula takes a greedy approach, looking for the optimal action to take rather than a probabilistic policy.

4.3 Model example

Having defined the model, a toy example will be given to explain the model. The purpose of this example is to illustrate how this model can find the optimal action at the current time and belief. The data for this example is randomly chosen.



$$\boldsymbol{P} = \begin{bmatrix} 0.98 & 0.01 & 0.01 & 0 & 0 \\ 0 & 0.98 & 0.01 & 0.01 & 0 \\ 0 & 0 & 0.98 & 0.01 & 0.01 \\ 0 & 0 & 0 & 0.98 & 0.02 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$O_{INSPECT} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$b_1 = [0.1; 0.3; 0.4; 0.1; 0.3]$$

$$c^I = €2{,}000$$

$$c^R = €4{,}000$$

$$c^{EM} = €15{,}000$$

$$c^P = [0,0,0,100,4000]$$

$$T = 2$$

$$\gamma = 1$$

Here, P is the transition matrix for the Markov chain in figure 1. For this example, perfect inspections are assumed. This is visualized using the observation matrix $O_{INSPECT}$. In the matrix, the probability of the state being the same as observed is 1, therefore the model does not include any error. Note that the inspection changes the observation probability for state 5 to state 2. Due to the emergency repair, the state is observed as if the real state is state 2. The values for the other parameters can be found below the observation matrix. The horizon of this scenario is only 2, so the only decision epoch is at the current time. In this example, no resource constraints are considered yet. It is assumed that there are infinite resources.

With the data above, the optimal policy for the current belief state and current time can be calculated with the model. First, the value of the action of doing nothing will be considered. Without inspecting in the current period, the expected penalty costs are incurred. These are:

$$V(1, Do\ Nothing) = \sum_{s \in S} b_1 * c^P + \gamma \sum_{o \in O} P(o|b,a)\, V^*(2, b_2^{DN})$$

$$V(1, Do\ Nothing) = 0.1 * 100 + 0.3 * 4000 = €1{,}210 + \gamma \sum_{o \in O} P(o|b,a)\, V^*(2, b_2^{DN})$$

This is the expected reward for the action 'Do Nothing' at the current time. Before calculating $V(2, b'_{DN})$, the current value of the other actions will be calculated first. For inspections, this is:

$$V(1, Inspect) = \sum_{s \in S} b_1 * c^P + b_1(5) * c^{EM} + c^I + \gamma \sum_{o \in O} P(o|b,a)\, V^*(2, b_2^{INS})$$

$$V(1, Inspect) = €7{,}710 + \gamma \sum_{o \in O} P(o|b,a)\, \gamma V^*\left(2, b_2^{INS}\right)$$

For maintenance, the value is:

$$V(1, Maintain) = \sum_{s \in S} b_1 * c^P + c^R + \gamma \sum_{o \in O} P(o|b,a)\, V^*(2, b_2^{MAIN})$$

$$V(1, Maintain) = €5{,}210 + \gamma \sum_{o \in O} P(o|b,a)\, V^*(2, b_2^{MAIN})$$

Note that the penalty costs across all actions are the same. A failed pipe at the start of decision epoch t will incur the penalty costs. Therefore, every action has the same expected failure costs.

With this calculated, the action 'Do Nothing' has the lowest expected costs. However, the future belief states must be updated according to the respective action. Furthermore, the costs of the final state must be calculated before a decision for the current epoch is made.

Doing nothing will maintain the old beliefs and multiply these with one load cycle of the degradation model. No observation about the next state can be made besides this load cycle observation, so the observation matrix is not included in this calculation. This gives:

$$b_2^{DN} = \boldsymbol{b_1} * \boldsymbol{P}$$

$$b_2^{DN} = [0.098; 0.295; 0.396; 0.105; 0.306]$$

With this the value of $V^*(2, b_2^{DN})$ can be calculated. Any actions will have no effect anymore since the horizon is only $T = 2$. The actions affect the belief state after the maintenance or inspection has been conducted, so only the terminal penalty costs must be considered. These costs are:

$$V^*(2, b_2^{DN}) = \sum_{s \in S} b_2^{DN}(s) * c^P$$

$$V^*(2, b_2^{DN}) = 0.105 * 100 + 0.306 * 4000 = €1{,}234.50$$

This gives a total of:

$$V^*(1, Do\ Nothing) = €1{,}210 + €1{,}234.50 = €2{,}444.50$$

For the other $V^*(2, b_2^{INS})$ and $V^*(2, b_2^{MAIN})$, the costs are divided depending on the state in which the pipe was encountered. Each of the $V^*(2, b_2^a | b_1 = s)$ can be calculated at the end of the horizon. For each of the V^*, the corresponding b_2 must be calculated as well. Since perfect inspections were assumed, the posterior belief after an inspection will always be 1. This posterior belief is then updated with a load cycle, but with the knowledge of the state. This gives:

$$V^*\left(2, b_2^{INS} | b_1^{INS} = 1\right) = \sum_{s \in S} c^P * [0.98; 0.01; 0.01; 0; 0] = €0$$

$$V^*\left(2, b_2^{INS} | b_1^{INS} = 2\right) = \sum_{s \in S} c^P * [0; 0.98; 0.01; 0.01; 0] = €1$$

$$V^*\left(2, b_2^{INS} | b_1^{INS} = 3\right) = \sum_{s \in S} c^P * [0; 0; 0.98; 0.01; 0.01] = €41$$

$$V^*\left(2, b_2^{INS} | b_1^{INS} = 4\right) = \sum_{s \in S} c^P * [0; 0; 0; 0.98; 0.02] = €80$$

$$V^*\left(2, b_2^{INS} | b_1^{INS} = 5\right) = \sum_{s \in S} c^P * [0; 0.98; 0.01; 0.01; 0] = €1$$

For the calculation of costs V^* of inspections, the observational probabilities and costs of the previous decision epoch must be incorporated, giving:

$$V^*(1, Inspection) = €7.710 + \sum_{o \in O} P(o|b, a) * V^*(2, b_2^{INS} | b_1 = o)$$

With perfect information, the $P(o|b, a) = b_1(s)$. Therefore:

$$V^*(1, Inspection) = €7.710 + \sum_{s \in S} b_1(s) * V^*(2, b_2^{INS} | b_1 = s)$$

$$V^*(1, Inspection) = €7.710 + €25 = €7.735$$

For maintenance, the same calculations are required. The state is known after maintenance, and the load cycle must be incurred, the same as with inspections. Thus:

$$V^*\left(2, b_2^{MAIN} | b_1^{MAIN} = 1\right) = \sum_{s \in S} c^P * [0.98; 0.01; 0.01; 0; 0] = €0$$

$$V^*\left(2, b_2^{MAIN} | b_1^{MAIN} = 2\right) = \sum_{s \in S} c^P * [0; 0.98; 0.01; 0.01; 0] = €1$$

$$V^*\left(2, b_2^{MAIN} | b_1^{MAIN} = 3\right) = \sum_{s \in S} c^P * [0; 0.98; 0.01; 0.01; 0] = €1$$

$$V^*\left(2, b_2^{MAIN} | b_1^{MAIN} = 4\right) = \sum_{s \in S} c^P * [0; 0.98; 0.01; 0.01; 0] = €1$$

$$V^*\left(2, b_2^{MAIN} | b_1^{MAIN} = 5\right) = \sum_{s \in S} c^P * [0; 0.98; 0.01; 0.01; 0] = €1$$

For this example, maintenance will consider perfect information as well. With this, the $V^*(1, Maintain)$ becomes:

$$V^*(1, Maintain) = €5.210 + \sum_{s \in S} b_1(s) * V^*(2, b_2^{MAIN} | b_1 = s)$$

$$V^*(1, Maintain) = €5.210 + €1.1 = €5{,}211.10$$

Thus, the optimal value function $V^*(1, b_1)$ can be written as

$$V^*(1, b_1) = \{€2{,}444.50; €7{,}735; €5{,}211.10\}$$

And the optimal policy is to do nothing at this moment. Or in terms of a policy definition:

$$\pi^*(1, b_1) = argmax\{Do\ Nothing, Inspect, Maintain\} = \{Maintain\}$$

In this example, the best course of action was to do nothing. However, the example shows that the expected future costs become lower by maintaining or inspecting. Maintaining resulted in a future expected costs of only €1.10, while doing nothing incurred a penalty of €1,234.50. With an increased horizon, the benefits of maintaining would have eventually become larger than doing nothing. Furthermore, with strict reliability constraints, inspections and maintenance can guarantee that a model meets these constraints.

4.4 Solving methods - POMDP

Solving a POMDP can be difficult, depending on the number of states, actions, and observations. Furthermore, finite horizon POMDPs are somewhat sparse in literature. Most literature from the literature review considered a concrete goal instead of a planning horizon. Both exact and approximate methods are available when solving POMDPs. Exact methods use either value- or policy iteration to determine the optimal value function for a problem. These methods have a high computation time, and the solving time is dependent on the size of the planning horizon. Without pruning value vectors from the solution, the computation time is doubly exponential based on the horizon length. Since this planning problem considers a pipe with a large horizon, the exact methods may not be feasible. However, both approximate and exact methods will be considered as possible solving methods.

4.4.1 Exact solving methods

For solving the POMDP exactly, a value iteration algorithm is used. This algorithm works like the example of calculating the POMDP. The value function of a t-step horizon is defined based on the current belief and the expected future belief depending on the action one takes. The formula for this method is described by:

$$V_{t+1}(b) = \max_{a \in A} \left[\sum_{s \in S} b(s) * R(s, a) + \gamma \sum_{o \in O} P(o | a, b) * V_t(b_o^a) \right]$$

This function determines the value function of a $t + 1$-step horizon. For a finite horizon POMDP, all value functions from the terminal stage until the first decision epoch must be written out in terms of expected future value.

A complication is the number of value functions that are introduced using this equation. Recalling the example, a 5-state inspection and maintenance policy must construct the future value function for each possible future observation. This example only considered a single decision epoch, but it still required at least 11 value functions. Increasing the decision epoch would increase the number of value functions exponentially. Furthermore, since the belief state is continuous, future calculations must be redone with different belief states since the history of observations has changed. Thus, the total number of value functions $\mathcal{V}$ that exist in a policy tree of size T, is: $|A| * |\mathcal{V}_{T-1}|^{|O|}$. Here, $\mathcal{V}_{T-1}$ denotes the number of value functions in the policy tree of size $T-1$, A is the set of actions and O is the set of observations.

With a restrictive budget, this equation can still be solved using dynamic programming. A restrictive budget limits the number of actions that can be taken across a finite horizon. Recall that assumption 4 states that the failures remain hidden. With this, the set of observations when doing nothing is equal to 1. The only observation for future states is that the system incurs one load cycle of the degradation model.

4.4.2 Approximate solving methods

To overcome the difficulty in exact planning, approximate methods are used as well. These methods estimate the value of being in a belief state, instead of requiring an exact representation. With this, schedules can still be constructed without the high computational complexity of the exact methods. These approximate methods are unable to guarantee an optimal schedule and may result in different schedules under the same conditions. This trade-off between computation time and performance cannot be avoided.

The approximate value function is constructed using reinforcement learning. The equation below changes, where the continuation reward based on the observations now changes to consider an approximation of the future rewards. This approximation across values is denoted by Q. This gives:

$$Q_{t+1}(b) = \max_{a \in A} \left[\sum_{s \in S} b(s) * R(s,a) + \gamma Q_t(b^a) \right]$$

Reinforcement learning calculates this approximation by iterating this formula across different samples. Every sample, the approximation of the real value function is updated with some learning parameter. One issue is that the belief state is continuous. With tabular RL algorithms, the size of the value function table would become infinite. Thus, the belief state must be approximated as well, or replaced by some approximation function.

Due to the quantity of solving methods, the RL algorithm will be comparatively simple. A Q learning algorithm can approximate the value well enough to at least provide information into the expected performance of reinforcement learning in POMDP spaces.

4.5 Optimization model – MILP

As an alternative of POMDPs, costs could be modeled explicitly and solved through modeling the costs as a mixed-integer linear program (MILP). Instead of constructing each value vector individually, they can be constructed and optimized simultaneously. The objective would be to minimize the expected costs across the scheduling horizon, using the expected belief states. The result is a schedule that is optimal for the average belief state across some planning horizon. To return a contingency plan like the dynamic program, the MILP must be run multiple times, considering all possible outcomes for each of the maintenance interventions. Alternatively, the policy from this method can be used as an aperiodic scheduling policy rather than a dynamic policy. This policy should perform well, but not as good as a dynamic policy. However, this sacrifices performance of the schedule to speed up computation time.

4.5.1 Objective

For the MILP, the same notation of variables and parameters is used as for the POMDP definition. The objective function of the MILP is the sum of all rewards across the scheduling horizon. For incurring inspection and maintenance costs, binary decision variables can be used. Decision variable x_t and y_t will refer to the action of inspecting or maintaining an asset at time $t \in T$. If both variables are 0, the model chooses to do nothing.

The belief state b is modeled through decision variables as well. Although the belief state cannot directly be influenced by the model, the values are determined by the choices in the model and the initial belief. Therefore, it cannot take the form of a parameter and it must be in modeled as a decision variable. b is defined as a decision variable for each state $s \in S$, across the horizon T. Other parameters in the model are deterministic. With this, the objective of the MILP is modeled as:

$$\min \sum_{t=1}^{T} \delta^t \left(\sum_{s \in S} \left(b_s^t * c_s^p \right) + x_t * \left(c^I + c^E * b_5^t \right) + y_t * c^R \right)$$

In this, the emergency repair costs cause non-linearity in the model. Since the expected emergency repair costs are dependent on the probability of encountering the pipe in state 5, the decision variable x_t and b_5 are multiplied. This causes the objective function to become non-linear. To solve this issue, an indicator variable u_t is introduced that takes the correct value when an inspection is conducted. Without inspections, the indicator variable will become 0. With this, the new objective function is:

$$\min \sum_{t=1}^{T} \delta^t \left(\sum_{s \in S} \left(b_s^t * c_s^p \right) + u_t + y_t * c^R \right)$$

4.5.2 Constraints

Using MILP to solve the model requires constraints that define the search space of the model. The constraints are defined explicitly to allow for recreation of the model. The complete set of constrains are split for readability.

1) $x_t + y_t \leq 1 \;\; \forall t \in T$
2) $\sum_{t \in T} x_t \leq I_{max}$
3) $\sum_{t \in T} y_t \leq R_{max}$
4) $b_1 = current\ state$

The first four constraints describe the actions and initial state. Constraint 1 states that the model can conduct maintenance, conduct an inspection, or do neither. Constraint 4 defines that the initial belief state is equal to the current state. If this initial state is a belief, the initial belief is set to the initial belief vector instead. Constraint 2 and 3 define the limit to the number of actions. When a budget is considered, this constraint changes to: $\sum_{t \in T} c^I * x_t + c^R * y_t \leq B_{MAX}$. This constraint would limit the sum of scheduled maintenance costs to be less than a predefined budget. The emergency costs are unscheduled maintenance costs, so these are not included in this constraint.

5) $u_t \leq x_t * M \quad \forall t \in \{1,2 \dots, T\}$
6) $u_t \geq c^I + c^{EM} * b_5^t - (1 - x_t) * M \quad \forall t \in \{1,2 \dots, T\}$
7) $x_t, y_t \in \{0,1\} \quad \forall t \in \{1,2 \dots, T\}$
8) $u_t \in \mathbb{R}^+ \quad \forall t \in \{1,2 \dots, T\}$
9) $b_s^t \in \mathbb{R} \cup [0,1] \quad \forall t \in \{1,2 \dots, T\}, \ \forall s \in S$

These constraints define the variables. Constraint 5 and 6 set the indicator value to the correct value by using big M constraints. This M is a value that is at least as high as $c^I + c^{EM}$. If the model does not inspect, u_t must be 0, since M is multiplied by 0. If the model does inspect, then the constraint states: $u_t \geq c^I + c^{EM} * b_t(5)$. Since the objective is to minimize costs, the variable is set to the minimum value it can take, which is the correct amount of costs.

10) $b_1^t * p_{1,1} - x_t \leq b_1^{t+1} \leq b_1^t * p_{1,1} + x_t \quad \forall t < T$
11) $b_2^t * p_{2,2} + b_1^t * p_{1,2} - x_t - y_t \leq b_2^{t+1} \leq b_2^t * p_{2,2} + b_1^t * p_{1,2} + x_t + y_t \quad \forall t < T$
12) $\sum_{s \in \{1,2,3\}} b_s^t * p_{s,3} - x_t - y_t \leq b_3^{t+1} \leq \sum_{s \in \{1,2,3\}} b_s^t * p_{s,3} + x_t + y_t \quad \forall t < T$
13) $\sum_{s \in \{1,2,3,4\}} b_s^t * p_{s,4} - x_t - y_t \leq b_4^{t+1} \leq \sum_{s \in \{1,2,3,4\}} b_s^t * p_{s,4} + x_t + y_t \quad \forall t < T$
14) $\sum_{s \in \{1,2,3,4,5\}} b_s^t * p_{s,5} - x_t - y_t \leq b_5^{t+1} \leq \sum_{s \in \{1,2,3,4,5\}} b_s^t * p_{s,5} + x_t + y_t \quad \forall t < T$

These five constraints describe the update of the belief state if there is no action taken. The upper and lower bound of the constraint is equal if both decision variables are 0. Therefore, the belief state will always update according to exactly one load cycle of the degradation model. The observation matrix is not included in these five constraints. The model receives no other signal between actions, meaning there is no probability of a different observation.

15) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,1} - (1 - x_t) \leq b_{t+1}(1) \ \forall t < T$
16) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,1} + (1 - x_t) \geq b_{t+1}(1) \ \forall t < T$
17) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,2} - (1 - x_t) \leq b_{t+1}(2) \ \forall t < T$
18) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,2} + (1 - x_t) \geq b_{t+1}(2) \ \forall t < T$
19) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,3} - (1 - x_t) \leq b_{t+1}(3) \ \forall t < T$
20) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,3} + (1 - x_t) \geq b_{t+1}(3) \ \forall t < T$
21) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,4} - (1 - x_t) \leq b_{t+1}(4) \ \forall t < T$
22) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,4} + (1 - x_t) \geq b_{t+1}(4) \ \forall t < T$
23) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,5} - (1 - x_t) \leq b_{t+1}(5) \ \forall t < T$
24) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Inspect) * b_t(s') * p_{s,5} + (1 - x_t) \geq b_{t+1}(5) \ \forall t < T$

These next ten constraints describe the belief state update if the model decides to inspect. Each upper and lower bound constraint is split in two separate constraints. These constraints also contain the observation probabilities, denoted by $O(s', s, Inspect)$. This set of observation probabilities is denoted by a matrix like in the example.

25) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,1} - (1 - y_t) \leq b_{t+1}(1) \ \forall t < T$
26) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,1} + (1 - y_t) \geq b_{t+1}(1) \ \forall t < T$
27) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,2} - (1 - y_t) \leq b_{t+1}(2) \ \forall t < T$
28) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,2} + (1 - y_t) \geq b_{t+1}(2) \ \forall t < T$
29) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,3} - (1 - y_t) \leq b_{t+1}(3) \ \forall t < T$
30) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,3} + (1 - y_t) \geq b_{t+1}(3) \ \forall t < T$
31) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,4} - (1 - y_t) \leq b_{t+1}(4) \ \forall t < T$

32) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,4} + (1 - y_t) \geq b_{t+1}(4) \; \forall t < T$
33) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,5} - (1 - y_t) \leq b_{t+1}(5) \; \forall t < T$
34) $\sum_{s \in S} \sum_{s' \in S} O(s', s, Maintain) * b_t(s') * p_{s,5} + (1 - y_t) \geq b_{t+1}(5) \; \forall t < T$

These final 10 constraints are comparable to the inspection constraints. However, these constraints contain the decision variable y instead of x and use the observation matrix for maintenance rather than inspection. This observation matrix would move states below state 2 back up to state 2.

5. Results

This chapter discusses the results of the model. First, section 5.1 conducts a parametric analysis. This analysis shows the performance of the model under different circumstances. Section 5.2 takes one combination of parameters from 5.1 to examine the performance of the maintenance policy from the different solving methods. A main method for the rest of the research is chosen and used for a parametric analysis in section 5.2. Next, section 5.3 will compare performance of a maintenance policy using unrestricted resources to a constrained policy. Finally, 5.4 will consider the impact of using the POMDP to model imperfect inspections.

The MILP model was programmed in Spyder using Python 3.11. The library used for creating and solving the MILP is called ‘Gurobi’, using an academic license. The figures were made using matplotlib.

5.1 Initial performance evaluation

Multiple solving methods have been established, and they are compared before conducting exhaustive tests. These comparisons are done with random parameters and data to examine the different performance levels between different solving methods. The goal is to find how the methods construct different schedules and how their performance differs.

The reinforcement learning model must be trained first. This is done by generating sample paths for the state of a pipe, based on the degradation model. The model only has access to the expected state, current time, and current budget. The decision of the model is made based on the current belief state, and the expected future belief state. After the model has inspected or maintained, the belief state is changed to the real state of the pipe. If the model changes the state through a scheduled- or emergency repair, the sample path must be redrawn. The old sample path does not account for this new state and the sample path must be redrawn.

The values for the model parameters can be found in table 3.

Parameter	**Base value**
c^p	[0,0,0,500,3000]
c^I	€1,500
c^R	€6,000
c^{EM}	€10,000
P	$\begin{bmatrix} 0.95 & 0.04 & 0.01 & 0 & 0 \\ 0 & 0.92 & 0.07 & 0.01 & 0 \\ 0 & 0 & 0.93 & 0.05 & 0.02 \\ 0 & 0 & 0 & 0.91 & 0.09 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
$O(inspect)$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$
T	60
B_{max}	15,000
δ	0.98

Table 3: The initial parameter calibration of the tests

5.1.1 Schedule evaluation

The schedule of the dynamic program cannot be reasonably be represented. Instead, a representation of the policy tree after the first action is represented in figure 4. Here, the first action is at the same time, for all the different results of the inspection. The next action for state 5 is the same as state 2 since this state has received emergency repairs. Furthermore, the first action after observing state 4 is to maintain the pipe. The risk of failure in this state is apparently high enough to warrant immediate repairs. The solving method must consider the expected future states, which it does not have access to now. Due to the assumption that no other observation is received between maintenance, the time of the first action will be the same initially. The subsequent actions change quite significantly based on the results. A poorer state than expected will move maintenance closer to the inspection time. Similarly, finding a good state moves the next action further down the scheduling horizon.

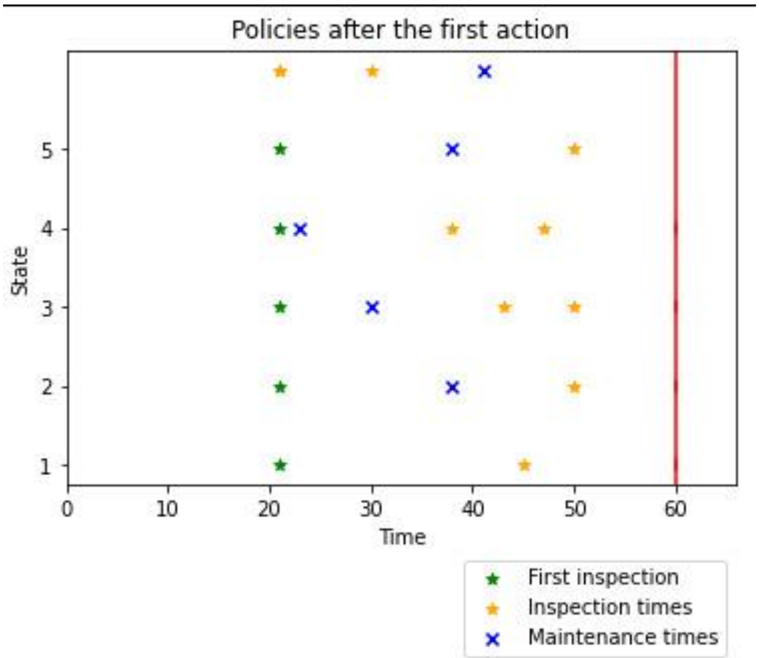


Figure 4: A visualization of the policy tree against the aperiodic schedule

The RL algorithm also constructs dynamic schedules that are difficult to visualize. However, the RL model only scheduled inspections and no maintenance. As with the dynamic program, the first instance was always the same for all pipes, although the RL algorithm did not schedule this at the same time as the dynamic program. Since the expected state is consistent for across the testing samples, this makes sense. Sadly, the RL algorithm did not schedule any maintenance in these tests. A possible cause can be the high costs attached to maintenance in this experiment. This may make it seem as if maintenance is not beneficial at first. This problem could be solved through more exploration or by initializing the state values to be some closer approximation of the real value.

The MILP was able to construct a schedule, for which the type and time of the first action correspond with the dynamic program. This schedule is found in figure 4, above the first node in the policy tree. This supports the hypothesis that both the MILP and dynamic program based the first action on the future expected state of the system. Furthermore, this finding supports that both schedules are indeed optimal, although the MILP generates an aperiodic schedule instead. Furthermore, none of the times of the subsequent schedule correspond with the times of the dynamic schedule. This finding shows the amount of difference between the two scheduling methods.

5.1.2 Performance evaluation

First, the dynamic program took approximately ten times longer to solve than the MILP. Furthermore, both models returned the same initial action at the same time. This makes sense, given that both construct this initial action around the same expected future state of the system. The expected costs of the schedules were not the same, however. The dynamic program returned lower expected costs across the planning horizon than the MILP. The difference in performance is due to the dynamic program being able to consider a new schedule after each inspection. Therefore, this difference in performance is expected.

The difference in runtime was much larger than expected, however. Furthermore, the MILP can exploit the faster runtime of the model, by rerunning the MILP for each possible observation after the maintenance. As a result, the MILP can construct a policy tree like the dynamic program, without having a significant increase in runtime. The MILP runtime becomes faster after the first action, since the remaining scheduling time and budget are smaller, limiting the search space. This method allows pruning the search space, which limits the future belief states to be only the relevant belief states instead of considering the complete belief space. For example, a model that finds the pipe in state 2 will never consider the option of fining the pipe in state 1.

The RL model was unable to provide a schedule with the same levels of performance as the MILP or dynamic program. The inspections were scheduled early, and the model did not seem to be able to schedule on the approximation of the state space. To further support the poor performance, the dynamic program constructed a myopic maintenance policy, where only 1 instance of maintenance or an inspection, could be scheduled at some optimal time. Comparing this to the reinforcement learning model, the performance of the RL model shows to be quite a bit poorer than the myopic policy. The RL model was able to confidently outperform no maintenance, but the other solving methods were all outperforming the RL algorithm. The RL algorithm did not schedule maintenance often, while the myopic policy only conducted maintenance. This is likely why the myopic policy outperformed the RL algorithm. Maintenance is reliably able to repair the system for reasonable costs, so it should be a priority when scheduling maintenance.

Finally, the myopic policy performed very similar to the more complex scheduling methods. Only the dynamic schedule was able to perform significantly better that the myopic policy. The cause of the similar performance is the decrease in gain of scheduling maintenance. The first intervention will decrease the risk significantly. Each additional action will reduce the risk further, but against diminishing returns. At some point, the increase in maintenance costs no longer reduces risk enough to warrant the inclusion of additional maintenance. This is best seen in the difference between initial risk and the myopic policy. The difference between the expected costs is the most significant, since the initial decrease in risk is the most significant.

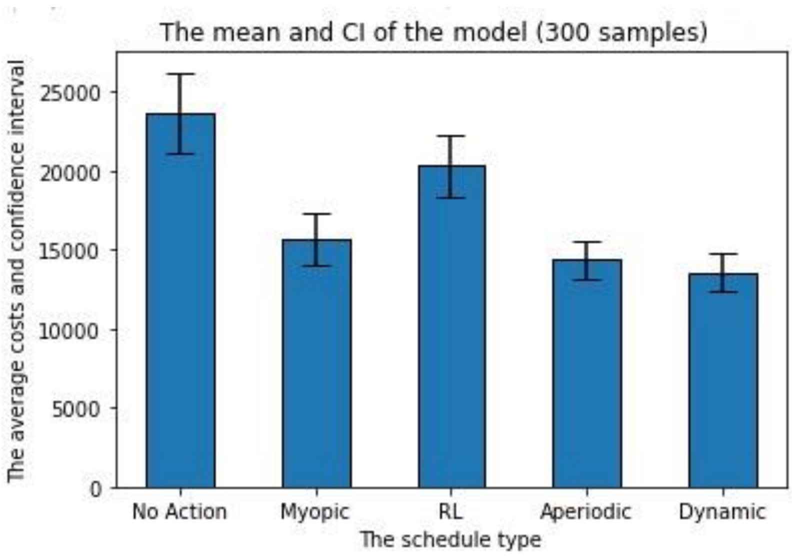


Figure 5: The difference in the costs across a simulation and the confidence intervals

5.2 Parametric analysis

The previous chapter showed the similar performance of solving methods. For the remaining research, one method is chosen to limit the number of experiments. For the rest of the research, the MILP will be used. The dynamic program was too slow, and the RL algorithm unable to provide consistent results.

Before considering model performance, the effect of model parameters must be understood. For this analysis, one parameter changes at a time to determine the effect on the policy. For this section, only the aperiodic policy $\pi^*(1, b_1)$ is considered. This initial policy is the expected schedule of the pipe, without considering the changes in schedule that result from observations of the system. The next section considers the impact this scheduling method has on the results of the model.

For each of the analyses, the other parameters are kept a constant value. These values can be found in table 4.

Parameter	**Value**
c^P	[0,0,0,500,3000]
c^I	€1,500
c^R	€6,000
c^{EM}	€10,000
P	$\begin{bmatrix} 0.95 & 0.04 & 0.01 & 0 & 0 \\ 0 & 0.92 & 0.07 & 0.01 & 0 \\ 0 & 0 & 0.93 & 0.05 & 0.02 \\ 0 & 0 & 0 & 0.91 & 0.09 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
$O(inspect)$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$
T	60
B_{max}	15,000

δ	0.98

Table 4 The base values for the parametric analysis

5.2.1 Degradation analysis

The values of the different degradation matrices can be found in table 5.

Run	P
1: High deg	$\begin{bmatrix} 0.89 & 0.08 & 0.03 & 0 & 0 \\ 0 & 0.89 & 0.10 & 0.01 & 0 \\ 0 & 0 & 0.91 & 0.07 & 0.02 \\ 0 & 0 & 0 & 0.89 & 0.11 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
2: Avg deg	$\begin{bmatrix} 0.95 & 0.04 & 0.01 & 0 & 0 \\ 0 & 0.92 & 0.07 & 0.01 & 0 \\ 0 & 0 & 0.93 & 0.05 & 0.02 \\ 0 & 0 & 0 & 0.91 & 0.09 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
3: Low deg	$\begin{bmatrix} 0.97 & 0.02 & 0.01 & 0 & 0 \\ 0 & 0.95 & 0.04 & 0.01 & 0 \\ 0 & 0 & 0.94 & 0.05 & 0.01 \\ 0 & 0 & 0 & 0.94 & 0.06 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Table 5 The degradation parameter values

By testing the degradation, the model shows that it is capable of scheduling maintenance based on different progression of degradation. The lowest degradation scheduled the fewest number of actions, while high degradation scheduled the highest number of actions. Due to the budget constraint, only two maintenance actions could be scheduled. After two repairs, only €3,000 would be left in the budget. This would be enough for two inspections, but the degradation is not high enough for this additional inspection to be valuable. Thus, the highest degradation still schedules one inspection.

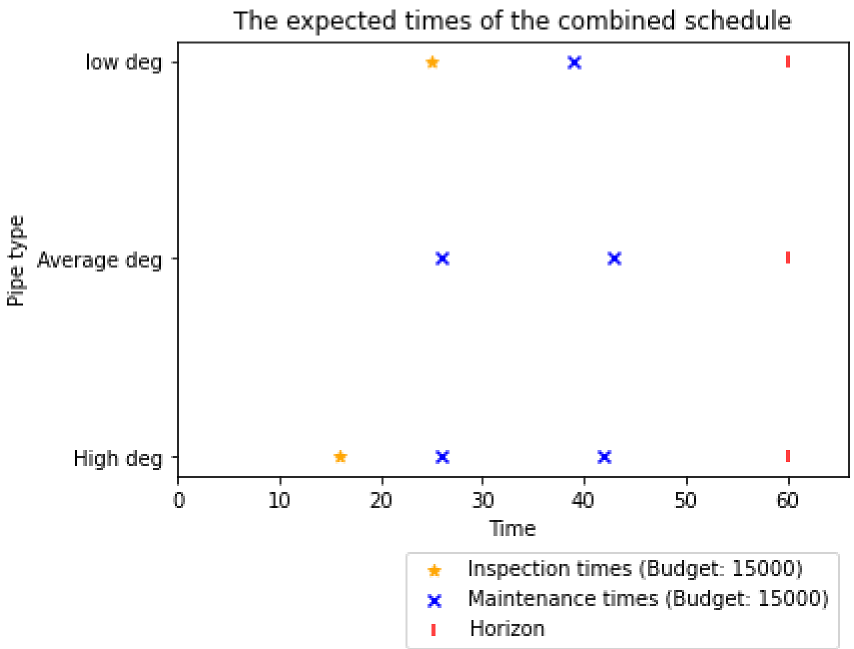


Figure 6 The aperiodic schedule with different levels of degradation

5.2.2 Penalty cost testing

Run	1	2	3	4	5
c^P	[0,0,0,0,1000]	[0,0,0,500,2000]	[0,0,0,500,3000]	[0,0,0,1000,4000]	[0,0,500,1500,5000]

Table 6: The different values for the penalty costs

The amount of scheduling increased with a higher penalty cost. The budget was set, such that a very limited amount of maintenance could be scheduled. As a results, the model ended up developing a schedule for which the expected time spent in the worst state is minimized. It is surprising that the schedules with the highest penalty cost of run 4 and 5 were the exact same. The budget is high enough to schedule two instances of maintenance and two inspections. However, it seems that the high penalty costs in earlier states, together with the emergency repair costs are a deterrent to scheduling inspections. At some level of penalty costs, the model will schedule an additional inspection.

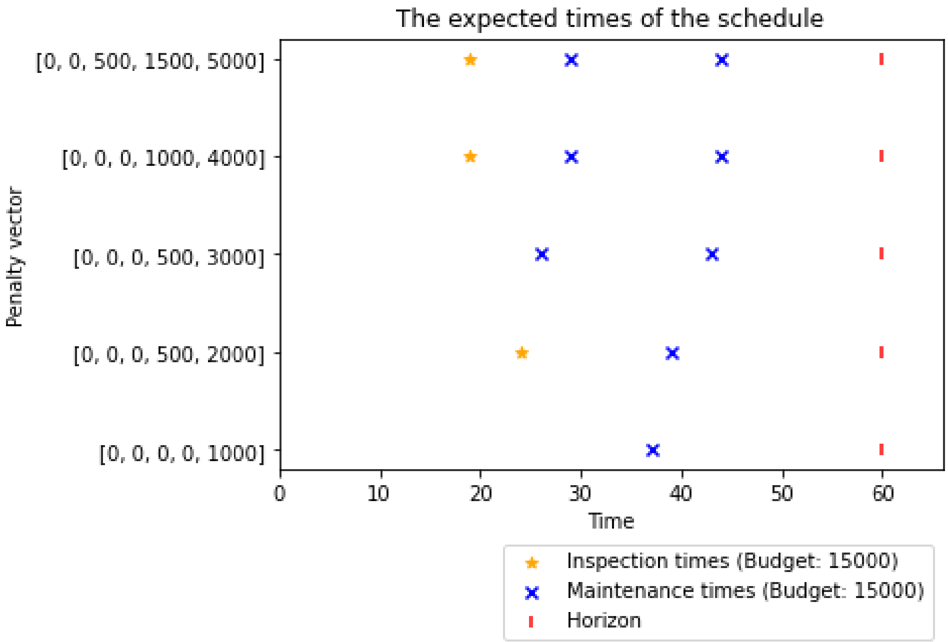


Figure 7 The aperiodic schedule with varying penalty levels

5.2.3 Maintenance cost testing

Run	1	2	3	4	5
c^I	1500	1500	1500	2500	5000
c^R	3000	6000	7500	10,000	10,000
c^{EM}	10,000	10,000	10,000	15,000	20,000

Table 7 The different maintenance cost values

The higher maintenance costs filled the budget quickly. In run 5, the model could only schedule 1 instance of maintenance and 1 inspection. In run 4 and 5, the effect of the emergency repair costs is interesting to see. Run 4 and 5 both used the same costs for maintenance, but different inspection costs. With the increase in inspection and expected emergency repair costs, scheduling 1 inspection is not valuable anymore. This choice increases risk, but the difference in expected final costs is minimal.

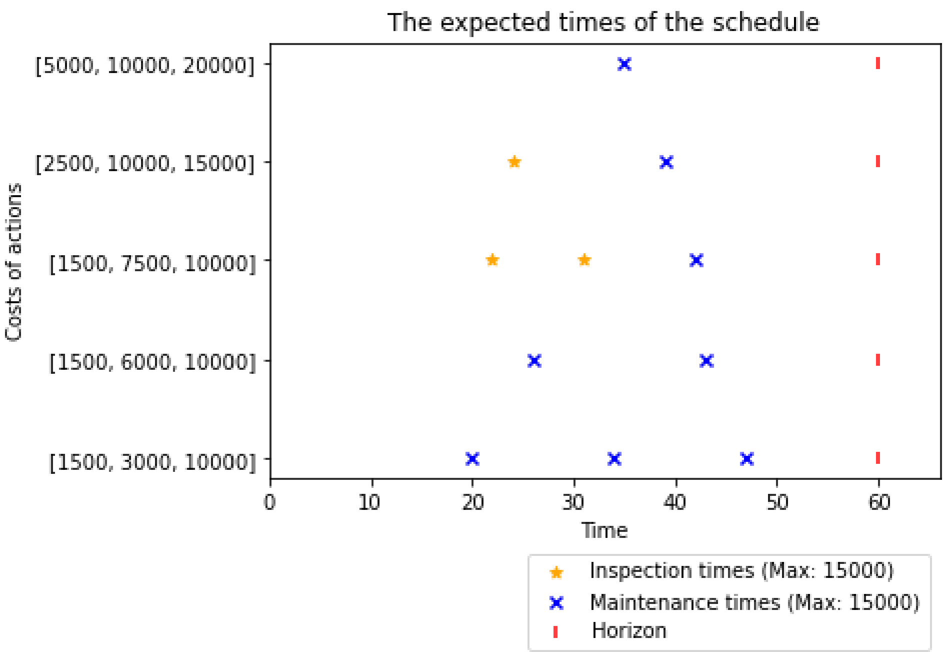


Figure 8 The aperiodic schedule with varying costs of actions

5.2.4 Initial state

Run	**1**	**2**	**3**	**4**	**5**
b_1	[1,0,0,0,0]	[0.7;0.2;0.1;0;0]	[0.5;0.5;0;0;0]	[0.4;0.4;0.1;0.1;0]	[0.2;0.5;0.1;0.1;0.1]

Table 8 The different initial states

The progression of the initial state shows an interesting pattern. The maintenance instances are scheduled with a somewhat consistent period between two instances. Most likely, maintenance is a predictable intervention. Therefore, expected state between two instances of maintenance are comparatively consistent. Further interesting to see is that the two maintenance instances move to earlier in the timeline for worse initial states. If the two instances of maintenance are best scheduled some constant times apart, the schedule around these two instances must be optimized. This can be seen in run 4 and 5, where the inspections are used to mitigate the remaining probabilities of failed states.

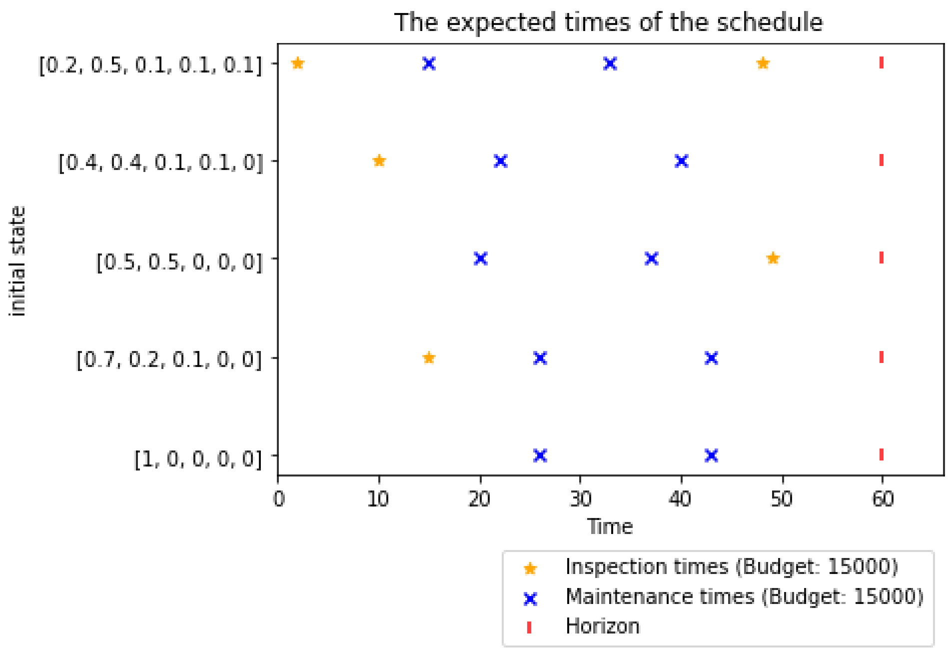


Figure 9 The aperiodic schedule with varying initial states

5.2.5 Horizon length

Run	**1**	**2**	**3**	**4**	**5**	**6**
T	30	40	50	60	70	80

Table 9 The different values for the horizon lengths

With a longer horizon, more load cycles occur for each pipe. Therefore, the model is incentivized to schedule more maintenance. This pattern can be found in the resulting schedules. The lowest horizon only scheduled one inspection to reduce the failure state probability. Interestingly, run five showed the same scheduling pattern that was seen at higher levels of degradation. Run six was incentivized to add one additional inspection to the expected schedule. However, the first action did not change to a later moment in time. Therefore, both policies should be considered as identical. Their initial recommendation does not change. The increase in horizon also affects the interval between two instances of maintenance, although not by much.

Furthermore, the horizon impacted the runtime quite significantly when compared to the other parameters tested in this chapter. Run 6 took 28 minutes, very close to the preemptively set time limit of 30 minutes. For the future experiments, a shorter horizon will be useful in shortening the runtime.

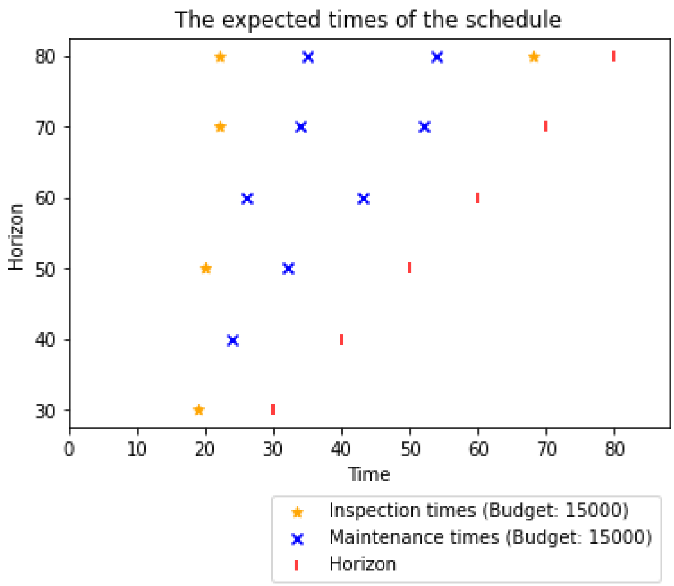


Figure 10 The aperiodic schedule with different horizon lengths

5.2.6 Discount factor

Run	**1**	**2**	**3**	**4**	**5**
γ	0.50	0.80	0.95	0.98	1.00

Table 10 The discount factor values

The discount factor has no other effect besides prioritizing immediate or future rewards. A discount factor of 0 places full priority on minimizing immediate reward. Alternatively, a discount factor of 1 equalizes the priority of immediate and expected future costs. This pattern is found in the model as well. The effects of the discount factor are as expected. A low discount factor removes the incentive to schedule. The difference between the discount factor of run 3 and 4 show how little this effect is.

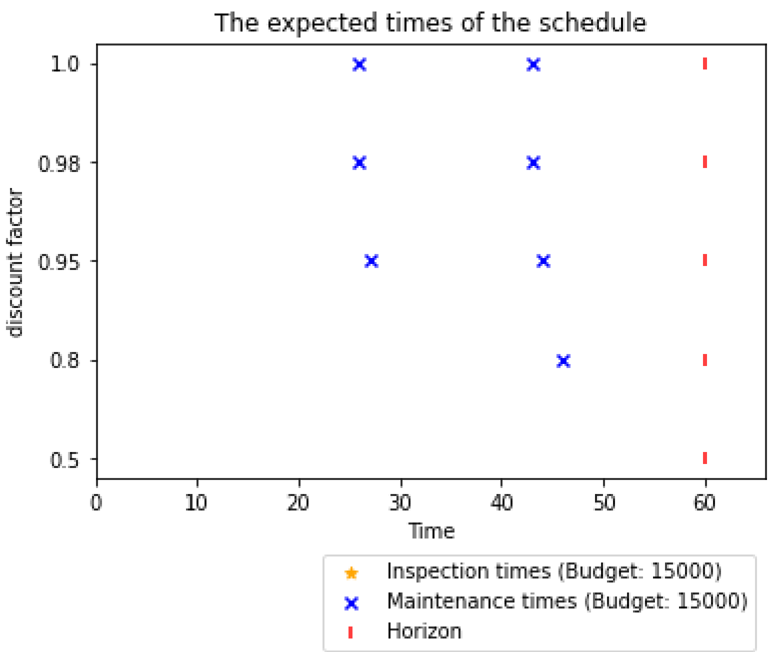


Figure 11 The aperiodic schedule with different discount factors

5.3 Testing budget restrictions

With a solving method established and a good understanding of model behavior, the main research question can be examined in detail. Different budgets are tested and compared against unrestricted budgets to determine the changes in the maintenance schedule. Only the aperiodic schedules will be compared, not the dynamic schedules. The dynamic program runtime is unreasonable, considering the runtime, so the MILP will be used instead. Due to this, the resulting schedules are aperiodic rather than dynamic. Furthermore, testing the unrestricted schedules is illogical. If one model is allowed to consistently schedule according to the optimum and the other cannot, the comparison does not make sense.

The values for these tests can be found in table 11. The horizon has been shortened slightly to speed up testing, but the other parameters are the same.

Parameter	**Value**
c^p	[0,0,0,1000,6000]
c^I	€1,500
c^R	€7,000
c^{EM}	€10,000
P	$\begin{bmatrix} 0.95 & 0.04 & 0.01 & 0 & 0 \\ 0 & 0.92 & 0.07 & 0.01 & 0 \\ 0 & 0 & 0.93 & 0.05 & 0.02 \\ 0 & 0 & 0 & 0.91 & 0.09 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
O*(inspect)*	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$
T	50
δ	0.98

Table 11 The base values for testing different budget values

5.3.1 Comparing schedules

The first test was run using the old parameters, but with a smaller penalty cost than in the table. This schedule reached its optimal schedule quickly and did not need to schedule further using a higher budget. These graphs can be found in appendix A. The other schedules all consider the doubled penalty costs, as listed in the table above.

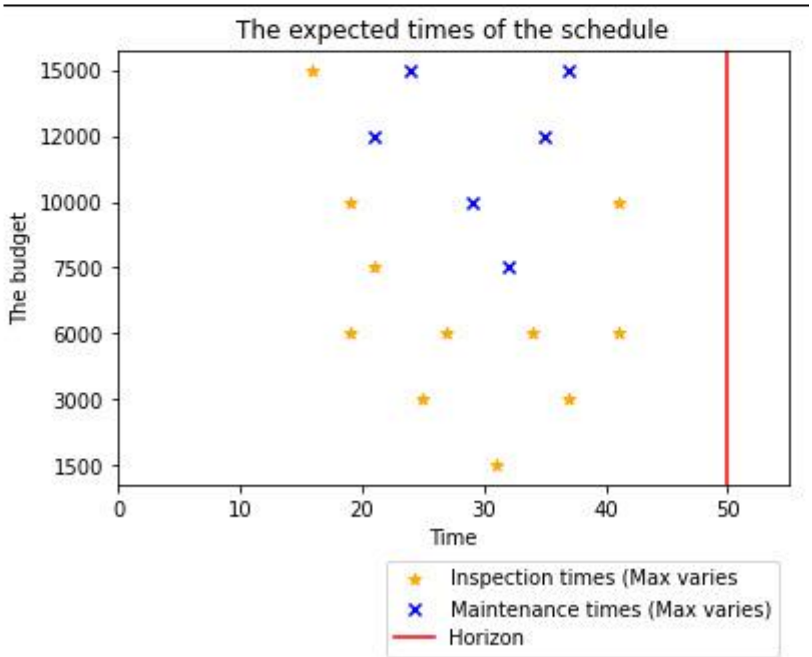


Figure 12 The aperiodic schedule with different budget levels

Clearly, the schedule changes based on the level of the budget. However, it is also interesting to see how the distribution of actions changes. It is more beneficial to schedule four inspections rather than one instance of maintenance. The model must find the probability of a failed state too high if only one instance of maintenance is conducted, thus it schedules 4 inspections instead. Furthermore, the time between maintenance increases with a more lenient budget. This is especially visible between the low budgets. The first inspection is pushed to be later in the schedule, likely due to the reduction of risk at that time. The more inspections that are being scheduled, the further back these inspections are conducted in the scheduling horizon.

At some point, maintenance becomes preferable due to the lower costs of repair compared to inspections. At the highest budget, the optimal schedule has been determined, or the budget needs to increase significantly. However, this seems unlikely when looking at the expected costs per schedule, in figure 12. The initial decrease is significant, but these improvements get smaller as the total reduction in risk becomes smaller. The difference between the final schedules is minimal, so it is likely that no schedule can improve upon these expected costs. Furthermore, the maintenance costs seem to increase almost linearly, except at the point where the first instance of maintenance is introduced.

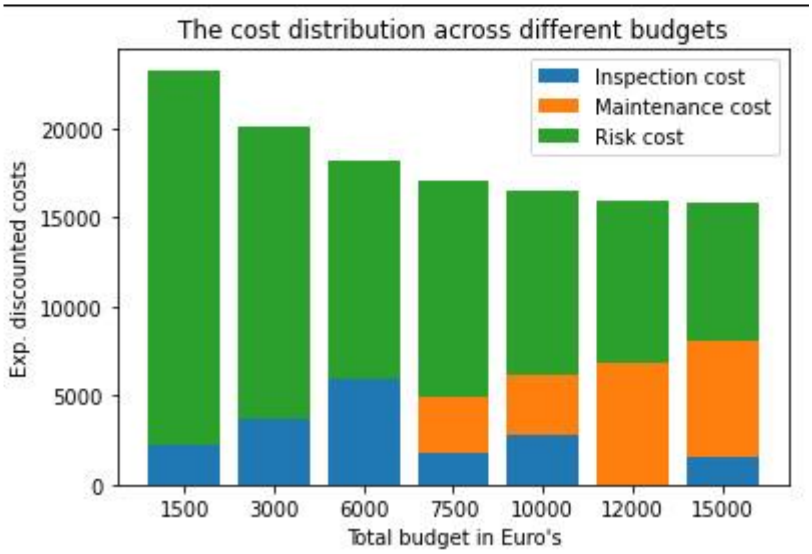


Figure 13 The expected costs for different budget values

5.3.2 Dynamic and aperiodic schedule performance with budgets

Now, the difference in performance between a static and periodic schedule under budget constraints is more closely examined. The expectation is that a dynamic schedule can react better to inspections than an aperiodic schedule. A dynamic schedule should be more efficient with the budget, postponing maintenance when the state of the system is good.

The initial schedule and the first posterior action policy tree can be found in figure 4, in section 5.1.1. The parameters were set in a way that the first action would be an inspection. Furthermore, only one instance of maintenance can be conducted with the current budget. This was to incentivize a reliance on inspections.

The difference in performance is not very significant. Both expected costs are very comparable, and the confidence intervals overlap. However, the results seem to suggest that the dynamic schedule will outperform the aperiodic schedule under specific circumstances. Tests were run with different levels of budgets, and one test with a larger sample size. This seemed to confirm that the dynamic schedule is improving the expected costs more quickly than the aperiodic schedule. The graph of these costs can be found in appendix A. The difference between both is still not enough to be significant, but this gap increased with larger sample sizes and budgets. In this scenario, it is unlikely that the aperiodic schedule will become significantly worse, without using irrational parameters. However, other domains may have characteristics where the costs of failure are higher, and inspections are required frequently. These domains would benefit significantly from implementing a dynamic schedule. However, implementation would also require more documentation and confidence in predicting degradation. Similarly, if frequent inspections may be required, a static interval could be much easier to consider.

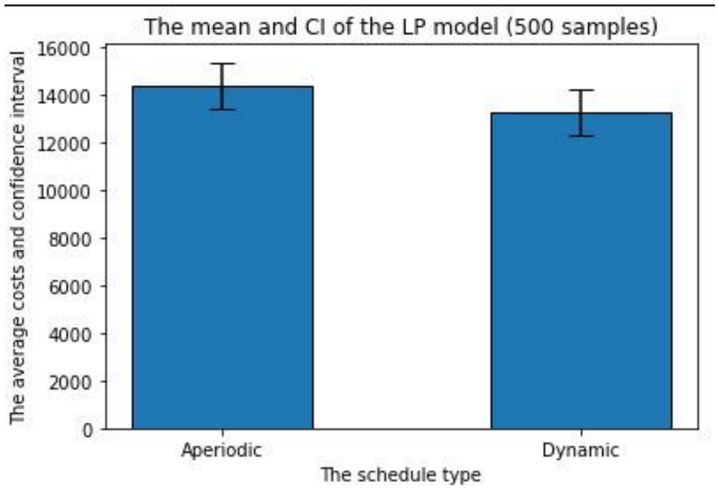


Figure 14 The comparison of costs between the aperiodic and static schedule

5.4 Imperfect inspections

Finally, the impact of considering imperfect inspections will be examined. One of the advantages of the POMDP is the option to consider imperfect inspections. Therefore, by modeling the costs alternatively, this advantage is lost. To determine the level of performance that is lost across a schedule, this impact is examined in more detail. First, multiple aperiodic schedules will be constructed, with varying levels of uncertainty of inspections. This comparison should show the expected difference in performance, that is examined afterwards.

Observation is only included in inspections. Furthermore, the parameters will be set to be a budget of $10{,}000$, double penalty costs and a horizon of 50. These values are derived from the previous test, where the model scheduled one inspection before and after an instance of maintenance.

Finally, three distinct observational matrices are considered. The first is the instance of perfect information, without any uncertainty of inspections. The next considers some uncertainty but assumes that damage can only be underestimated. If an inspection is wrong, the system must be in a poorer state. The final observation matrix considers both options, although not equally. A failed state cannot be missed or misjudged, still. In table 13, the observation matrices are located. They are $K \times K$ in size. The values along the horizontal axis of the matrix describe the new belief vector. The vertical axis describes the actual state of the system. The high uncertainty vector may cause the inspections to become so invaluable that they will not be scheduled regardless of the budget.

Run	**O(Inspection)**
1: No uncertainty	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$
2: Some uncertainty	$\begin{bmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 & 0 \\ 0 & 0 & 0.8 & 0.2 & 0 \\ 0 & 0 & 0.1 & 0.9 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

3: High uncertainty	$\begin{bmatrix} 0.7 & 0.2 & 0.1 & 0 & 0 \\ 0.1 & 0.7 & 0.1 & 0.1 & 0 \\ 0.1 & 0.2 & 0.6 & 0.1 & 0 \\ 0 & 0.1 & 0.1 & 0.8 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Table 12 The observational probability matrices

5.4.1 First experiment

For these tests, it was assumed that a failed state is always identified correctly, and that maintenance would still be perfect. The initial effects were minimal, but still interesting. The model removed one of the inspections from the schedule and rescheduled the maintenance to mitigate the increased risk. The cost comparison and schedule can be found in figure 15 and 16 respectively. The reduced value of inspections mitigated too little risk to warrant the additional inspection. Since both uncertainty level schedules do not include any inspections, the results are the same. Additionally, the schedules would be an optimal myopic schedule where you can only schedule one instance of maintenance (comparable to 5.1, but with a shorter horizon). Since both do not include any inspections, the level of uncertainty does not impact the optimal schedule. The increase in total costs is very little, so the inspection did not add much value, even with perfect inspections.

Since the first test did not show any significant results, two more experiments are run with new parameters. First, the penalty costs are doubled to incentivize scheduling maintenance. Secondly, the budgets are set on 8,500 and 10,000. This would only allow 1 repair, so further penalty costs must be mitigated through inspections.

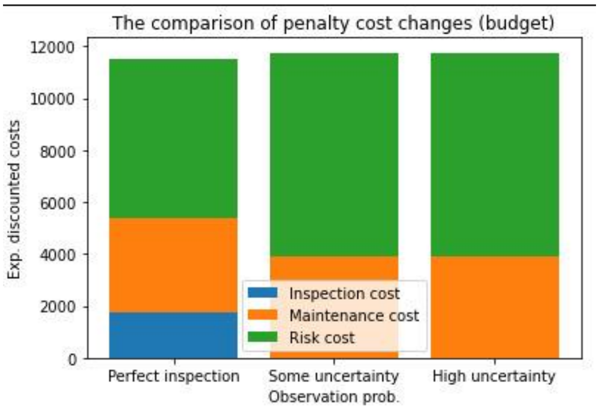


Figure 15 The imperfect inspection expected costs

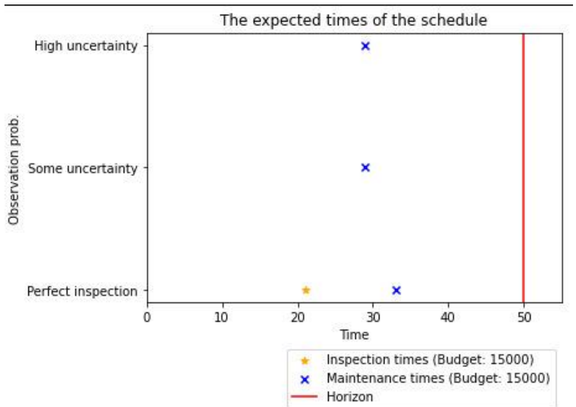


Figure 16 The imperfect inspection aperiodic schedule

5.4.2 Second experiment

The results of the second round of experiments can be found in figures 17 and 18. Again, the model reduced the number of inspections when imperfect inspections are considered. This time, the impact on the schedule is more visible. The incurred risk increases by a visible margin since inspections do not yield the benefits of mitigating poor states as well. The highest uncertainty changes the order of maintenance and inspections. Furthermore, the increased budget confirmed the hypothesis in 5.4. The optimal schedule had been found one iteration earlier since the value of information is too low for an additional inspection to have any value. Therefore, the schedule contains less maintenance than its more accurate counterparts.

Another finding is that the inspections are placed close to the horizon and the repair in the case of small uncertainties. Since the inspection still removes any probability of occupying state 5, the value of inspection is almost as much with perfect inspections as it is without. However, the reentry rate into state 5 is higher since the belief state after an inspection is poorer. The model responds by scheduling inspections closer to the end of life or repairs. Since both can reset the belief state to a better state, it makes sense for the model to place the inspections closer to these more impactful events. This both mitigates risk, while also mitigating the additional uncertainty that is incurred through the imperfect inspections.

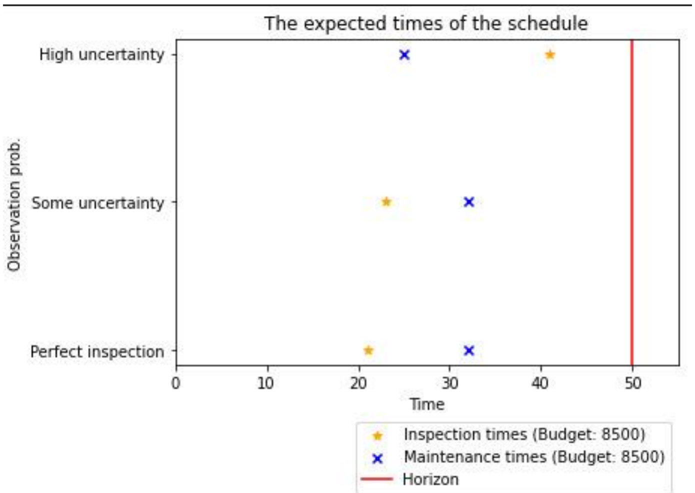


Figure 17 The imperfect inspection aperiodic schedule with a budget of 8,500

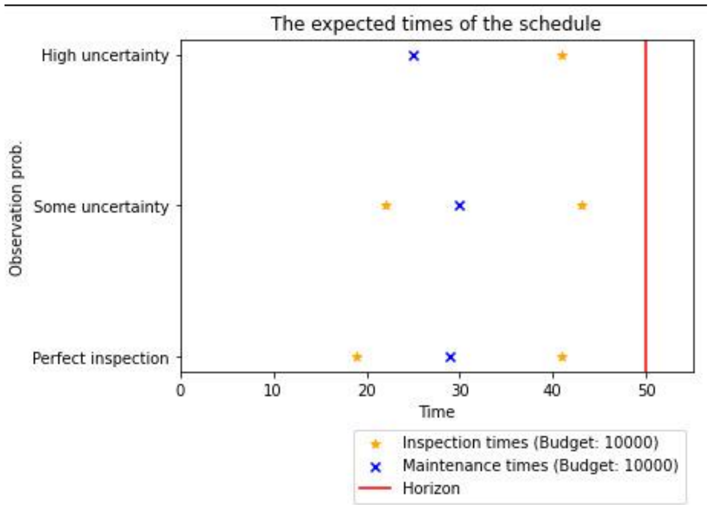


Figure 18 The imperfect inspection aperiodic schedule with a budget of 10,000

6. Conclusion and recommendations

This chapter discusses the conclusions that are drawn from the findings. The main conclusion is drawn in section 6.1. Then, the added value of modeling the problem as a POMDP is discussed in 6.2. The limitations are discussed in section 6.3. Finally, some future research directions are presented in section 6.4. Some concluding remarks can be found in section 6.5.

6.1 Conclusion

The problem of maintaining a sewer network is interesting, due to the hidden failures of the network and the inability to observe the system. Furthermore, the size of the system limits the ability of inspecting or maintaining at will. This has resulted in a risk averse scheduling policy, where reliability of the network has priority over efficiency. However, the aversion to risk has led to inefficiency of assigning resources.

A model is constructed to design a maintenance schedule where the limited resources are taken into consideration during the scheduling process. Furthermore, the model schedules based on the expected hidden state, and can consider the inaccuracies of inspecting an asset. With this model, the different schedules were examined and optimized.

For this research, the problem of maintaining a sewer network was tackled. The network does not fail obviously, and the failures are costly if or when they occur. In the introduction, the most important research directions were discussed, and the following research question was proposed:

To what extent do budgetary constraints for maintenance affect the determination of optimal inspection intervals of inspection decisions for sewer pipes?

This question has been examined in detail and a model was constructed to look at its effects. The literature review found use of POMDPs to model these optimization problems, and a large quantity of methods to solve these models. The choice was made to model the problem as a POMDP, using belief states to replace the normally observable state. These beliefs indicated the probability distributions across states. They allowed the model to make decisions based on the expected state and removed the need to assume the state. To further test the results of the model, an MILP was designed to explicitly model the costs and generate an exact schedule based on the expected costs of maintenance across the horizon.

The results showed that the model understood how to schedule inspections and maintenance optimally. However, performance was poor with approximate methods. The schedules were inconsistent, and some performances were worse than models without maintenance. The RL algorithm was even unable to outperform a myopic scheduling method. This myopic schedule was constructed optimally under the circumstances, so it makes sense that this policy performs well, but the difference was still significant.

In the initial results, the exact solving method took too long to solve. However, the MILP was able to establish the optimal moment of the first maintenance action. Therefore, this model was used to test the parameters of the solving methods to determine if the model would be able to provide good results within different scenarios. The times of the schedules differed, but there was a pattern that emerged, based on the parameter values. Testing the schedule using simulations resulted in a good performance.

The maintenance and inspection schedule changes significantly when a budget is imposed on the schedule. Furthermore, the times at which inspecting is considered optimal changes based on the available data and budget. Thus, when optimizing a scheduling problem, an upper limit on the number of inspections should always be considered. Furthermore, a schedule that can consider multiple maintenance actions and reschedule based on results can in theory outperform a model that cannot. In practice, the additional scheduling and administration may cause confusion or mismanagement. However, the performance gap increases with higher failure costs and more lenient budgets. Therefore, the gap in performance can become significant in some domains.

However, the parameters influence the levels at which these budget restrictions impact the performance of the final schedule. A slowly degrading component may not fill the entire budget, so both a restricted and unrestricted budget will perform similarly if not the same. High degradation may require the model to conserve available actions, thus scheduling at times when it may not be optimal without a budget. The differences in performance vary with budget, the timespan/horizon, the penalty costs, and the actions available. The findings show that there are instances where the performance differences become significant.

Imperfect inspections cause inspections to become much less beneficial and the number of inspections decreases as uncertainty increases. Furthermore, the time between maintenance is decreased to account for the uncertainty.

In the end, the model was able to attach a value to inspections. Inspecting allows decision makers to prevent hidden failures in this scenario. The higher the penalty of a failed condition, the more risk-averse the model will behave. Even though the concept is simple, the results seem to be more than reasonable. The model schedules maintenance intuitively and was able to interpret the future state probabilities. The benefit of dynamic scheduling is small, unless the costs of failure is high compared to maintenance. Thus, aperiodic schedules can suffice if there is no reliability restriction.

6.2 Added value of POMDPs

This research modeled the problem as a POMDP and modeled the costs explicitly. With this, the added value of specifically modeling the problem as a POMDP could be examined. Due to the high expected computation time, both exact and approximate methods were used.

It is not reasonable to use a naïve method and explicitly model the value vectors of the POMDP and solve them, even at a small to medium size scale. The observable domain becomes too large to model all value vectors individually. Literature considered several algorithms that would prune relevant value vectors, to limit the search space. However, these algorithms were not considered important due to the comparatively small scale of the problem. After all the results, these algorithms should have been implemented to speed up the process. However, it is unclear if these algorithms would have been effective enough for the small test instances in this research. Furthermore, these algorithms will not be enough for large-scale problems. A small sewer network can still contain multiple kilometers of pipes, so the minimal scale of this system is high compared to other networks. Thus, it does not seem reasonable for any POMDP to be exactly solvable in this problem domain at this moment. It should be noted that the POMDPs is still a developing concept and that the algorithms may improve in the future.

The MILP turned out to be an alternative pruning method for the POMDP. The aperiodic schedule was optimal for only the first action, due to observations being limited to inspections and maintenance. This first action could be conducted, and the schedule rebuilt based on the results.

This allows the schedule to become optimal, even without having to construct the full policy tree from the current time. While computation may be faster, this is also a disadvantage. The insight into the expected use of resources is limited without a near-complete policy tree. Therefore, considering all possible alternatives may be valuable.

The reinforcement learning algorithm was able to solve the problem in a reasonable time, but the solution performed worse, even when compared to a myopic policy. Of course, many simplifications were made to make the RL algorithm work and it was not the focus of this research. Firstly, the state space was approximated using a grid-based approximation method. This allowed the belief space to be dividable into discrete steps instead of a continuous value. Secondly, simple RL algorithms are known to perform poorly under budgetary constraints without some additional risk mitigation algorithm. This was discovered after the results were poor. Therefore, a conclusion about the levels of performance that RL can achieve in partially observable domains cannot be drawn here. More complex algorithms may be able to better approximate the value function and perform better.

The added value of the POMDP is to have a dynamic policy, that can incorporate current information in future decision-making. However, the gain per additional instance of maintenance is diminishing, and the computation time is significant. Therefore, one must consider the real benefit in terms of cost or reduced risk with using POMDPs to schedule maintenance. The additional administrative requirements of dynamic schedules may invalidate these benefits. Aperiodic schedules require less complexity and can scale up much faster than POMDPs. Additionally, these can be approximated more easily than POMDPs and are more common in current scientific literature. POMDPs may be useful in high value asset situations, where the costs of failure are a tier higher than in this research.

6.2 Limitations and discussion

As to the limits of this research, there are several. The most important and most emphasized so far has been degradation. Without an accurate degradation model, a schedule will not perform well in this optimization model. The choices are fully made on the expected state distribution of the item, so inaccuracies decrease performance by over- or under scheduling maintenance. It is for this reason that sewer asset managers seem to err on the side of caution, as was found in the problem description. Furthermore, the inherent data biases make it even more difficult to properly determine the level of precaution. Research into degradation modeling seemed to be an ongoing research topic in literature. Once a good degradation model is found, the findings of this research should be reexamined. The maintenance schedules found throughout this study were good with different degradation models and parameters. The schedules from these experiments may be suboptimal under a different degradation model.

Further limitations regarding degradation are the damage type, and associated repair costs. Here, the damage was simplified into one degradation model, that used a single type of repair cost for analysis. More accurate would have been to split the damage types, which would further support the degradation modeling. Additionally, since the repair cost estimation is based on the damage type of the pipe, the repair cost should not be a single parameter as it was here. A future model could change the penalty vector to include more values for the different damage types, improving model accuracy. When done, the penalty vector itself should also be expanded to include the different damage types and their own penalties. In this research, only a singular vector was considered. Operational failures occur most often in this domain, so this should be the primary focus.

A deep reinforcement learning model may be able to both provide more accurate degradation modeling and more accurate schedules. Runtime should be much better with deep reinforcement learning models. Furthermore, these models can incorporate different damage levels and types without loss of computation time. These models should be used with care, however. They are reliant on the quality of data, which is not guaranteed in sewer and water maintenance. This was one reason why machine learning was not considered as a degradation model here.

Additional limitations are the assumptions of having no failures or any other disruption in the model. Markov chains do not necessarily model failure across an entire horizon very well unless constant normalization is applied. With the MILP, the normalization would create non-linearity, further increasing runtime. Therefore, failure was considered as risk in the penalty vector instead. Other disruptions to the network would include calamities or random environmental effects, that would lower states or increase the degradation. These are not uncommon, when the entire horizon of the asset is considered. Across sixty years, it is reasonable that some incident may happen that will change the optimal schedule. These are difficult to model, so they were not included here.

Finally, the final limitation is that scale was kept small. The network is large, and opportunistic maintenance can reduce costs possibly more than removing a single inspection or maintenance could. Combining maintenance is already done in practice, but the foundation on which decision-making is built is not ideal. Including this practice in a model would make the runtime even more egregious. For this reason, only one pipe was considered as well. Optimizing the schedule for multiple components at the same time was tested, but each instance ran into the time limit of 30 minutes, even with a limited budget.

6.3 Future research

For future research, more sophisticated approximate methods should be used. Previous literature discussed the evolutionary algorithms, which can certainly be applicable here. The genetic algorithm was a common method, and it was proposed as one of the better scheduling tools. Other methods that have gained more attention are the machine learning methods. Reinforcement learning was used in this research, but the results were less than ideal. POMDPs have not been tested with RL in a scheduling problem, so the poor results were somewhat expected. When introducing a strict budget, a simple RL algorithm was unable to consider a lack of resources in the future, thus scheduling inspections and maintenance freely or not at all. Of course, the more complex deep RL subverts this limitation by including a neural network. However, the freely available libraries do not focus on partial observability or finite horizons, meaning one may have to be developed from scratch.

Other future research within sewer maintenance should focus on damage types. The main cause of blockages are sudden operational failures. Here, only the physical damages were considered. A better examination of damages may be one that differentiates between internal and external damages and maintenance. The types of maintenance impact repair costs and degradation, so including this will expand on two current limitations at the same time. If using this model, that would require a change to at least the way belief states are read, and the penalty vector. The current model is simple, and expansion should be reasonable. Solving this more complex objective may become too time intensive. Therefore, heuristic methods such as the genetic algorithm may have to be used.

As an alternative, this model has seemed reasonably sturdy within this case study. The schedules were solid when compared to their simpler counterparts and the expected costs were reasonable. The potential of this model may lay outside the field of buried infrastructure, or another specific part of buried assets. The ability to influence the parameters so freely and still retain a reasonable schedule makes this a model that could work in a variety of areas. If inspections and maintenance are too expensive for a regular interval, while failure is undesirable, this model may be able to prove useful. The budgeted model has shown that it can handle larger choices of actions, while still running within a reasonable time limit. The constant monitoring of belief states will give a good indication of the level of risk you are running with the schedule, and you can react accordingly.

6.4 Remarks

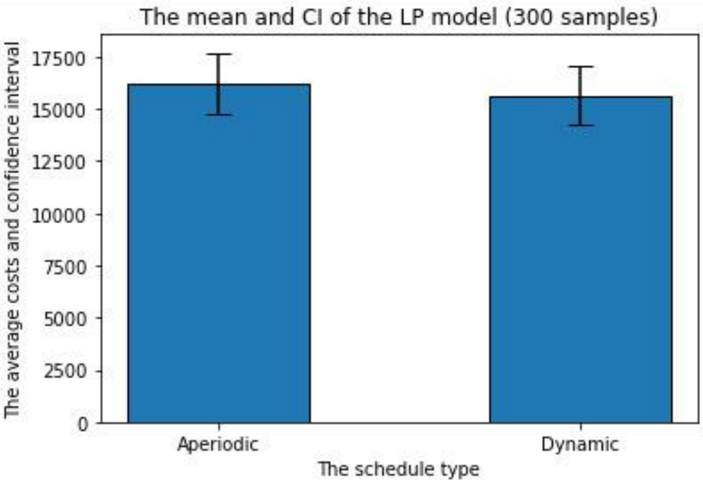
Further testing was conducted with more complex RL algorithms using free libraries in Python and genetic algorithms. With these libraries, a deep Q-learning (DQN) algorithm was tested by building a custom environment and running the premade learning and solving algorithms. These RL algorithms were unable to construct good schedules as well. It is thought the RL models were unable to consider restricted resources or finite horizon. Their schedules were scheduling maintenance very early in the horizon. This resulted in little maintenance scheduled after the initial maintenance. Furthermore, these libraries seem focused on infinite horizon problems rather than finite horizons. Therefore, it is very possible that coding errors were made in constructing the environment or calibrating the model parameters.

The genetic algorithm (GA) was able to construct the optimal schedule in some runs, depending on the input parameters. The objective was the same as the MILP, minimizing the expected costs across the horizon to establish an aperiodic maintenance schedule. The runtime was low, and it was able to optimize larger scale problems containing multiple pipes at the same time. However, the schedules became inconsistent with an increased number of pipes. The schedules in the tests were very different, but all managed to achieve similar expected costs in simulations. For constructing aperiodic schedules across a finite planning horizon, the evolutionary algorithms seem very applicable.

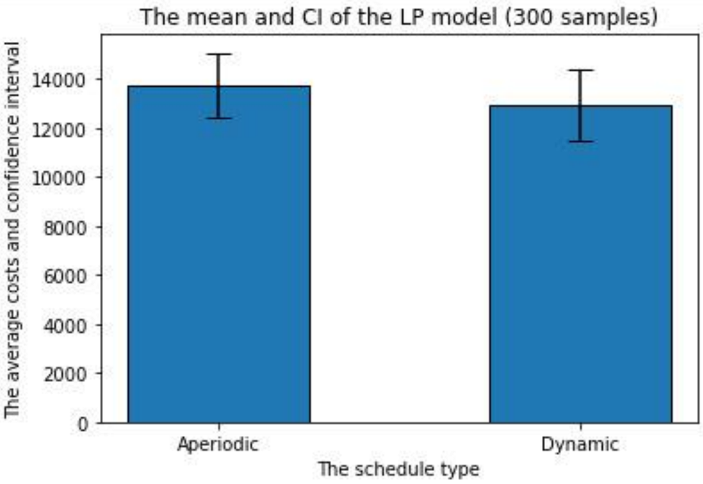
Appendix

A. The cost comparison of dynamic and aperiodic schedules

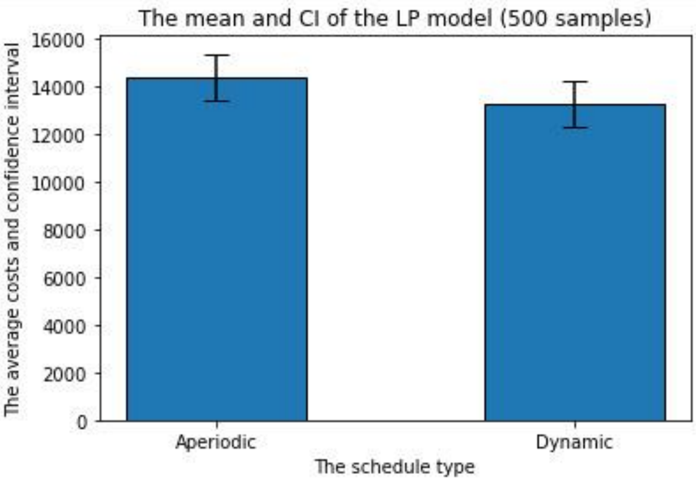
A.1 A budget of 6000



A.2 A budget of 10,000



A.3 A budget of 15,000 (repeated)



References

Abubakirov, R., Yang, M., & Khakzad, N. (2020). A risk-based approach to determination of optimal inspection intervals for buried oil pipelines. *Process Safety and Environmental Protection*, *134*, 95–107. https://doi.org/10.1016/j.psep.2019.11.031

Alaswad, S., & Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering and System Safety*, *157*, 54–63. https://doi.org/10.1016/j.ress.2016.08.009

Anbari, M. J., Tabesh, M., & Roozbahani, A. (2017). Risk assessment model to prioritize sewer pipes inspection in wastewater collection networks. *Journal of Environmental Management*, *190*, 91–101. https://doi.org/10.1016/j.jenvman.2016.12.052

Babishin, V., & Taghipour, S. (2016). Optimal maintenance policy for multicomponent systems with periodic and opportunistic inspections and preventive replacements. *Applied Mathematical Modelling*, *40*(23–24), 10480–10505. https://doi.org/10.1016/j.apm.2016.07.019

Braziunas, D. (2003). *POMDP solution methods*.

Bukhsh, D. A., Zhang, Z., & Jansen, Y. (2022). *Grouping of Maintenance Actions with Deep Reinforcement Learning and Graph Convolutional Networks*. *2*, 574–585. https://doi.org/10.5220/0000155600003116

Dao, C. D., & Zuo, M. J. (2017). Optimal selective maintenance for multi-state systems in variable loading conditions. *Reliability Engineering and System Safety*, *166*, 171–180. https://doi.org/10.1016/j.ress.2016.11.006

de Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. In *European Journal of Operational Research* (Vol. 285, Issue 3, pp. 805–824). Elsevier B.V. https://doi.org/10.1016/j.ejor.2019.09.047

Ding, S. H., & Kamaruddin, S. (2015). Maintenance policy optimization—literature review and directions. In *International Journal of Advanced Manufacturing Technology* (Vol. 76, Issues 5–8, pp. 1263–1283). Springer London. https://doi.org/10.1007/s00170-014-6341-2

Draude, S., Keedwell, E., Kapelan, Z., & Hiscock, R. (2022). Multi-objective optimisation of sewer maintenance scheduling. *Journal of Hydroinformatics*, *24*(3), 574–589. https://doi.org/10.2166/hydro.2022.149

Elmasry, M., Zayed, T., & Hawari, A. (2019). Multi-Objective Optimization Model for Inspection Scheduling of Sewer Pipelines. *Journal of Construction Engineering and Management*, *145*(2). https://doi.org/10.1061/(asce)co.1943-7862.0001599

Fan, X., & Yu, X. (2022). *Machine Learning-Assisted Optimal Schedule of Underground Water Pipe Inspection*. https://doi.org/10.21203/rs.3.rs-2335041/v1

Fauriat, W., & Zio, E. (2020). Optimization of an aperiodic sequential inspection and condition-based maintenance policy driven by value of information. *Reliability Engineering and System Safety*, *204*. https://doi.org/10.1016/j.ress.2020.107133

Fenner, R. A. (n.d.). *Approaches to sewer maintenance: a review*. www.elsevier.com/locate/urbwat

Hajipour, Y., & Taghipour, S. (2016). Non-periodic inspection optimization of multi-component and k-out-of-m systems. *Reliability Engineering and System Safety*, *156*, 228–243. https://doi.org/10.1016/j.ress.2016.08.008

Hawari, A., Alkadour, F., Elmasry, M., & Zayed, T. (2020). A state of the art review on condition assessment models developed for sewer pipelines. *Engineering Applications of Artificial Intelligence*, *93*. https://doi.org/10.1016/j.engappai.2020.103721

Heerkens, H., Winden, A. van, & Tjooitink, J.-W. (n.d.). *Solving managerial problems systematically*.

Janga Reddy, M., & Nagesh Kumar, D. (2020). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: A state-of-the-art review. In *H2Open Journal* (Vol. 3, Issue 1, pp. 135–188). IWA Publishing. https://doi.org/10.2166/h2oj.2020.128

Kim, S., Choi, J. H., & Kim, N. H. (2022). Inspection schedule for prognostics with uncertainty management. *Reliability Engineering and System Safety*, *222*. https://doi.org/10.1016/j.ress.2022.108391

Kleiner, Y. (2001). Scheduling Inspection and Renewal of Large Infrastructure Assets. *Journal of Infrastructure Systems*, *7*(4), 136–143. https://doi.org/10.1061/(asce)1076-0342(2001)7:4(136)

Kumar, V., & Yadav, S. M. (2022). A state-of-the-Art review of heuristic and metaheuristic optimization techniques for the management of water resources. In *Water Supply* (Vol. 22, Issue 4, pp. 3702–3728). IWA Publishing. https://doi.org/10.2166/ws.2022.010

Lin, Z. L., Huang, Y. S., & Fang, C. C. (2015). Non-periodic preventive maintenance with reliability thresholds for complex repairable systems. *Reliability Engineering and System Safety*, *136*, 145–156. https://doi.org/10.1016/j.ress.2014.12.010

Liu, X., Zheng, J., Fu, J., Ji, J., & Chen, G. (2018). Multi-level optimization of maintenance plan for natural gas pipeline systems subject to external corrosion. *Journal of Natural Gas Science and Engineering*, *50*, 64–73. https://doi.org/10.1016/j.jngse.2017.11.021

Mancuso, A., Compare, M., Salo, A., Zio, E., & Laakso, T. (2016). Risk-based optimization of pipe inspections in large underground networks with imprecise information. *Reliability Engineering and System Safety*, *152*, 228–238. https://doi.org/10.1016/j.ress.2016.03.011

Marzouk, M., & Omar, M. (2013). Multiobjective optimisation algorithm for sewer network rehabilitation. *Structure and Infrastructure Engineering*, *9*(11), 1094–1102. https://doi.org/10.1080/15732479.2012.666254

Mohamed, M. A. H., Ramadan, M. A., & El-Dash, K. M. (2023). Cost optimization of sewage pipelines inspection. In *Ain Shams Engineering Journal* (Vol. 14, Issue 5). Ain Shams University. https://doi.org/10.1016/j.asej.2022.101960

Nguyen, L. Van, & Seidu, R. (2022). Application of Regression-Based Machine Learning Algorithms in Sewer Condition Assessment for Ålesund City, Norway. *Water (Switzerland)*, *14*(24). https://doi.org/10.3390/w14243993

Obradović, D., Šperac, M., & Marenjak, S. (2023). Challenges in Sewer System Maintenance. *Encyclopedia*, *3*(1), 122–142. https://doi.org/10.3390/encyclopedia3010010

Okwori, E., Viklander, M., & Hedström, A. (2020). Performance assessment of Swedish sewer pipe networks using pipe blockage and other associated performance indicators. *H2Open Journal*, *3*(1), 46–57. https://doi.org/10.2166/h2oj.2020.027

Osman, H., Atef, A., & Moselhi, O. (2012). Optimizing Inspection Policies for Buried Municipal Pipe Infrastructure. *Journal of Performance of Constructed Facilities*, *26*(3), 345–352. https://doi.org/10.1061/(asce)cf.1943-5509.0000216

Pack Kaelbling, L., Littman, M. L., & Cassandra ', ', A. R. (1998). Artificial Intelligence Planning and acting in partially observable stochastic domains. In *Artificial Intelligence* (Vol. 101).

Phan, D. T., & Zhu, Y. (2015). Multi-stage optimization for periodic inspection planning of geo-distributed infrastructure systems. *European Journal of Operational Research*, *245*(3), 797–804. https://doi.org/10.1016/j.ejor.2015.03.038

Puterman, M. L. (n.d.). *Markov Decision Processes*.

Quatrini, E., Costantino, F., Di Gravio, G., & Patriarca, R. (2020). Condition-based maintenance-An extensive literature review. In *Machines* (Vol. 8, Issue 2). MDPI AG. https://doi.org/10.3390/MACHINES8020031

Salihu, C., Hussein, M., Mohandes, S. R., & Zayed, T. (2022). Towards a comprehensive review of the deterioration factors and modeling for sewer pipelines: A hybrid of bibliometric, scientometric, and meta-analysis approach. In *Journal of Cleaner Production* (Vol. 351). Elsevier Ltd. https://doi.org/10.1016/j.jclepro.2022.131460

Shafiee, M., & Sørensen, J. D. (2019). Maintenance optimization and inspection planning of wind energy assets: Models, methods and strategies. In *Reliability Engineering and System Safety* (Vol. 192). Elsevier Ltd. https://doi.org/10.1016/j.ress.2017.10.025

Singh, I., Popova, E., & Kee, E. (2008). Optimal preventive maintenance policy under a budget constraint. *International Conference on Nuclear Engineering, Proceedings, ICONE*, *1*, 13–18. https://doi.org/10.1115/ICONE16-48107

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An Introduction (2nd edition 2018). In *Learning* (Vol. 3, Issue 9).

Tscheikner-Gratl, F., Caradot, N., Cherqui, F., Leitão, J. P., Ahmadi, M., Langeveld, J. G., Le Gat, Y., Scholten, L., Roghani, B., Rodríguez, J. P., Lepot, M., Stegeman, B., Heinrichsen, A., Kropp, I., Kerres, K., Almeida, M. do C., Bach, P. M., Moy de Vitry, M., Sá Marques, A., … Clemens, F. (2019). Sewer asset management–state of the art and research needs. In *Urban Water Journal* (Vol. 16, Issue 9, pp. 662–675). Taylor and Francis Ltd. https://doi.org/10.1080/1573062X.2020.1713382

Van Horenbeek, A., Pintelon, L., & Muchiri, P. (2010). Maintenance optimization models and criteria. *International Journal of System Assurance Engineering and Management*, *1*(3), 189–200. https://doi.org/10.1007/s13198-011-0045-x

Wang, M., & Yin, X. (2022). Construction and maintenance of urban underground infrastructure with digital technologies. In *Automation in Construction* (Vol. 141). Elsevier B.V. https://doi.org/10.1016/j.autcon.2022.104464

Wang, W. (2000). A model to determine the optimal critical level and the monitoring intervals in condition-based maintenance. *International Journal of Production Research*, *38*(6), 1425–1436. https://doi.org/10.1080/002075400188933