Optimising Bridge Maintenance Planning Considering Hindrance to Road Users

An improved method combining optimisation and simulation applied to the city of Amsterdam

Mela Sagasser

Optimising Bridge Maintenance Planning Considering Hindrance to Road Users

An improved method combining optimisation and simulation applied to the city of Amsterdam

by

M. Sagasser

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on October 1st, 2024

Student number:	4498313	
Project duration:	Dec 1, 2023 – Sep , 2024	
Thesis committee:	Prof.dr.ir. Serge Hoogendoorn,	TU Delft, chair
	Dr.ir. Gonçalo Correia,	TU Delft, supervisor
	Dr. Maaike Snelder ,	TU Delft, TNO
	Dr. Mr. N. Mouter, Msc	TU Delft
	Ir. Lara Borst / Laura Hakvoort	Municipality of Amsterdam
	Dr. Marjolein van der Meer	TNO

Preface

This thesis marks the end of my Master's in Transport, Infrastructure and Logistics, and with it, the end of my student years in Delft. During my Bachelor's in Technology, Policy and Management, I developed my interest in networks and optimisation within the energy sector. During my Master's I got to apply this interest to the field of urban transportation.

This research sparked my curiosity on multiple levels, developing an optimisation for Amsterdam's urban road network while working closely with real decision makers and their often complex considerations. Therefore, I really enjoyed the combination of working together with the Municipality of Amsterdam to learn about the processes of planning maintenance in practice, as well as working with TNO to delve deeper into optimisation and newly developed tools that assist in decision-making.

However, working on a thesis by yourself can sometimes feel like being lost in a mirror maze: having no idea which way to go, trying directions that turn out to be dead ends, while being confronted with yourself again and again. Luckily, I had some great supervisors who have guided me to the end of this process. From the TU Delft I would like to thank my graduation committee. Serge Hoogendoorn, Maaike Snelder and Niek Mouter, thank you for your helpful and always positively delivered feedback and specifically Gonçalo Correia, thank you for giving directions when I needed it most, always leaving me feeling positive and motivated after every meeting. From the Municipality I would like to thank Lara and Laura for their warm welcome at the Municipality and for their enjoyable brainstorm sessions. I really felt part of the team thanks to everyone of team innovation and hope that the team lunches will stick around and make everyone at PBK jealous. A special thanks to Marjolein for the weekly meetings at TNO, whose help went far beyond guidance on optimisation. Thank you for keeping me on track and above all for letting me believe in my own capabilities.

I will not end this preface before thanking all the people close to me who have made this process much more bearable. To all my friends and family, thank you for all of your unconditional support that got me through these past months.

Mela Sagasser
Delft, October 2024

Summary

The city of Amsterdam is currently facing a significant infrastructure crisis due to the poor condition of their extensive network of canals, consisting of many old bridges and quay walls. Like many countries, the Netherlands has structurally underfunded infrastructure maintenance, leading to a substantial backlog of necessary renovation and renewal work. In response to several incidents involving structural collapses, the Amsterdam municipality established the ”Programma Bruggen en Kademuren” (PBK), an initiative designed to address this maintenance backlog and ensure the safety and accessibility of the city. PBK’s responsibility includes approximately 200 kilometres of quay walls and 850 bridges.

To catch up on overdue maintenance, the renovation rate from before the programme had to be increased by a factor of 20. In a city already burdened by many construction activities, this leads to significantly more hindrance caused to road users. Currently, a condition-based approach is used, wherein structures are selected for maintenance based on their actual condition and estimated remaining lifespan. This approach causes maintenance projects to be spread over the whole city causing hindrance on many different routes. The impact on traffic is particularly significant during bridge maintenance. The municipality has adopted a “short and heavy” maintenance strategy, where roads will be completely closed to minimise the total obstruction time and increase construction worker safety. Since bridges serve as crucial links in the road network and typically have longer detour routes, their complete closure can significantly increase the travel times for road users. In cases of severe traffic disruptions, the municipality considers more costly maintenance executions that reduce the traffic impact, such as reducing the duration by employing more costly work hours per week to reduce the total obstruction time. The municipality therefore needs to weigh the traffic impact of the maintenance plans with their costs.

However, with high impact projects spreading throughout the whole city and a complex network with high traffic density, it is difficult to foresee what the effect of simultaneous closures would be, as the different closures can have impact on each other. Currently, maintenance planning still heavily relies on expert judgement to reduce hindrance to road users, as the current used traffic model of Amsterdam, ‘Verkeersmodel Amsterdam’ (VMA), is computationally too slow to evaluate numerous potential solutions. This could lead to suboptimal decision-making and more hindrance caused to the road users than necessary. To reduce hindrance caused by the planning of bridge maintenance while taking into account the maintenance cost, an improved optimisation method is needed.

Currently, the municipality is working on optimisation. PBK is actively working towards a data-driven approach, using current state data to predict the future state of structures. In a collaboration with TNO, they used these predictions to optimise the long-term maintenance scheduling. TNO’s research focused on lifetime extension scheduling for quay walls, considering risk, maintenance cost, and hindrance to the city. Using the Urban Strategy Tool (UST), which offers much faster simulations than VMA, TNO assessed the impact of road closures by looking at the additional travel time for only vehicles. To estimate the effect of combined closures on traffic, the additional travel times of individual closures were summed, as if the closures were isolated from each other. By only simulating all possible closures individually, the high computational times due to many possible combinations could be avoided. However, this simplified estimation method neglects network effects and can result in wrong conclusions about the optimal combinations of maintenance projects. When creating the short term planning of maintenance in the upcoming years, there are many different possibilities in how the selected projects for those years can be combined in the planning. It is important that these network effects are taken into account when optimising the short term planning of upcoming bridge maintenance, since different combinations of projects can have a very different effect on traffic.

Within literature there is however a notable gap in optimising short-term or more operational maintenance planning to reduce hindrance to all road users, particularly when it comes to accounting for the interdependencies between multiple simultaneous projects. There have been a number of studies in

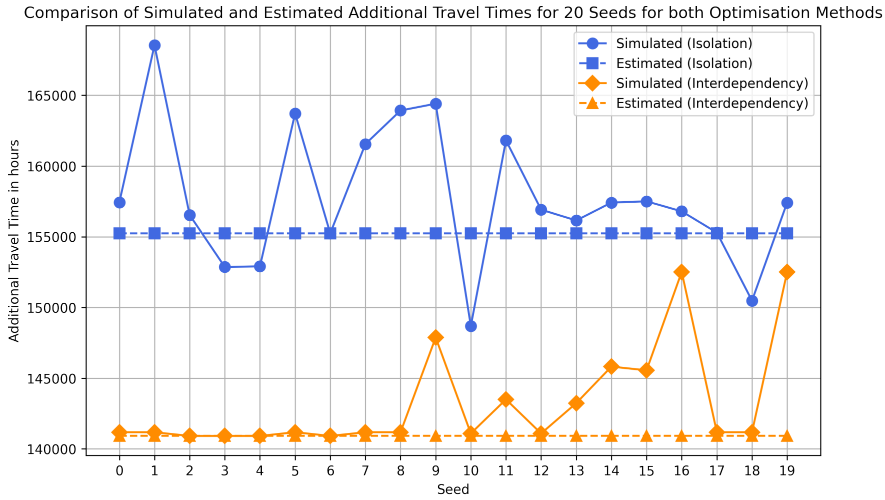


Figure 1: Comparison of the simulated and estimated additional travel times for 20 seeds for both optimisation methods. (Seeds determine the sequence of random numbers generated used in the optimisation process)

literature on optimising long-term maintenance of road structures. Often the goal was to select the best year and type of maintenance to maximise the lifetime of the structure without structural failure. If hindrance to road user due to maintenance was included in the optimisation, this was often done in a rudimentary manner and for vehicles only. The use of traffic simulation, which can improve the assessment of combined closures, is hardly considered due to the high computational times.

To fill this gap in literature and improve Amsterdam's maintenance scheduling, this study proposes an improved optimisation method using simulations for planning upcoming bridge maintenance, aiming to reduce additional travel time for all road users while balancing maintenance execution costs. Due to the complexity of bridge maintenance optimisation problems in terms of number of possible combinations and complicated evaluation, meta-heuristics are often used. For the proposed optimisaiton method in this research a genetic algorithm is used. This algorithms finds the optimal starting point and the execution duration for a given set of bridge maintenance projects for a certain period. The objective of the genetic algorithm is to minimise additional travel time to cars, freight and bicycles, while also minimising the costs for execution. A weight is given to the hindrance component in the objective function, which can be adjusted to reflect the priorities of policymakers.

For the assessment of additional travel time caused by maintenance projects, the traffic simulation of UST is used. Despite the fast simulations, the many possible combination of closures cannot all be simulated due to the computation time increasing exponentially with the number of considered bridge closures. To still be able to take network effects into account when assessing simultaneous closures, a new estimation method based on the UST simulations is proposed to improve the existing estimation method for combined closures of TNO's quay wall maintenance optimisation.

This new estimation method uses the interdependencies between bridge pairs to come to a more realistic estimate of the effect of simultaneous closures. The interdependency value of a bridge pair is determined by the change in travel time on a bridge after a closure of another. A negative percentage change would mean that closing the other bridge too would lead to less additional travel time than closing them separately. A positive percentage change would mean that closing the other bridge too would lead to more additional travel time than closing them separately. The larger the absolute percentage change, or interdependency value, the more important it is to take into account the relation between these bridges when calculating the additional travel time of the combined closure. The proposed estimation method evaluates a set of simultaneous closed bridges by pairing the bridges in order of the highest absolute interdependency value, and then using the simulation of the combined closure for each pair. This requires only all individual closures and combinations of two to be computed in advance, which can be done within a few days for up to 50 bridges. This keeps the genetic algorithm fast to evaluate solutions using different policies.

The optimisation method has been tested on case for the city of Amsterdam. Using the genetic algorithm, the maintenance plan for bridge closures in the upcoming 24 months was generated for four bridges with three different options for execution duration, having higher costs for faster execution. Three of the bridges were located near each other to see the effect of interdependencies. To compare the new estimation method using interdependencies with the previous estimation method using summation assuming isolation, both estimation methods were used to generate solutions. To access the accuracy of the additional travel time estimations of both methods, these were compared to fully simulated additional travel times of the produced schedules. These results are shown in Figure 1.

The results showed that the proposed interdependency estimation method did not only provide more accurate estimations than the isolation method, it also provided consistently better solutions in terms of additional travel time. The genetic algorithm using the interdependency method proved to be a fast option to evaluate many different possible bridge maintenance schedules while taking into account the interdependencies between simultaneous closure. For the municipality, this is therefore a useful extension of their current bridge maintenance planning in order to reduce the hindrance caused to road users by closures.

Interesting improvements of the proposed optimisation model are expanding the maintenance options considered and making them bridge specific, improving the assessment of additional travel time by including more modes or using other traffic simulations, and improving the evaluation of simultaneous closure by also taking into account interdependencies between separate evaluated pairs in combined closures. In addition to model improvements, several other future research options are suggested to reduce hindrance caused by bridge maintenance. This includes researching the effect of other factors influencing the hindrance experienced and developing methods to improve equity by distributing hindrance over different road user groups.

Contents

List of Figures

List of Tables

Abbreviations

GA	Genetic Algorithm
MRR	Maintenance, Repair and Replacement
OD	Origin-Destination
PBK	Programma Bruggen en Kademuren (Programme Bridges and Quay Walls)
UST	Urban Strategy Tool
VMA	Verkeersmodel Amsterdam (Traffic Simulation Model of Amsterdam)
V&OR	Verkeer en Openbare Ruimte (Traffic and Public Space)

1

Introduction and problem contextualisation

This first chapter introduces this research on bridge maintenance optimisation considering hindrance to road users. First, the context in which the research problem arose is stated. The research problem is then clearly formulated, based on the identified literature gap and needed improvements in practice. Next, a brief description of this research approach to addressing this problem is given, stating the research objective and questions to reach that goal. Lastly, an overview of the research will be presented.

1.1. Research context

Amsterdam is known for her historical city centre with many picturesque canals which determine the way the streets of Amsterdam look. This makes the municipality of Amsterdam responsible for almost 1800 bridges and 600 kilometre of quays (Gemeente Amsterdam, 2023). Many of those bridges and quays are more than a century old and not designed for the heavy loads and intensive use of today. Moreover, only reactive maintenance of bridges and quays has been done in the past decades due to not enough budget going to maintenance of infrastructure. In the last few years, some quays have even collapsed leading to dangerous situations (NOS, 2020).

To catch up on overdue maintenance after the incidents and renovate on a large scale, the municipality established a new program called 'Programma Bruggen en Kademuren' (Programme Bridges and Quays) in 2019 (Gemeente Amsterdam, 2023). They selected around 850 bridges and 200 km of quays based on their predicted state for maintenance in the upcoming years to ensure safety and functioning of the city. Since the establishment of the program, more renovation has been done per year than in the years before. However, to catch up on the overdue maintenance, even more construction work has to be done in the coming years. In the Amsterdam coalition agreement for 2022 to 2026, it is stated that because of the limited financial resources, hard decisions have to be made to maintain the city (Moorman et al., 2022). Specifically regarding the repair of the bridges and quays, it is said that investments are being postponed and cost-saving measures such as constructing sheet piling and closing bridges will more often be used. In recent years, maintenance work on bridges and quays has already caused significant disruptions. Given the current circumstances, the city will experience these effects even more intensely (Programma Bruggen en Kademuren, 2022).

The recent developments in Amsterdam are just one example of what is happening on national scale. Not only in Amsterdam, but in the whole country funds going to maintenance of the civil structures are structurally inadequate, which has resulted in a backlog of maintenance (Bleijenberg, 2021). The recent report of Rijkswaterstaat about the state of the Dutch infrastructure shows that a lot of maintenance will be required in the upcoming years due to deferred maintenance, heavier use of the infrastructure and the fact that most civil structures are approaching the end of their expected lifespan (Rijkswaterstaat, 2023). The problem certainly occurs with bridges, since 84% of fixed bridges and 56% of movable bridges have less than 33% of their expected lifespan left. Also in this report is indicated that much more hindrance will be caused by all construction work in the upcoming years to keep the country accessible and safe.

1.2. Research problem definition

The situation described above shows that much more maintenance has to be done in the upcoming years, resulting in more hindrance. Especially in a city like Amsterdam with great mobility demand and a complex urban road network with bridges, maintenance can lead to high costs for the road users such as cyclists, pedestrians, cars and public transport. In addition, the municipality has limited budget for the maintenance plans. Complex decisions are needed to determine the scheduling of maintenance while keeping the city safe and accessible. One of the current problems the municipality is facing, is how to reduce the hindrance to road users caused by maintenance on bridges.

In the beginning of the programme Bridges and Quays (Programma bruggen en kademuren, PBK), it was thought that almost all bridges and quays had to be renewed as soon as possible. If every bridge and quay wall needed to be renewed, maintenance could be planned wisely with the corridor approach. This approach meant that all bridges in a continuous route would be renewed at the same time, to reduce the hindrance caused to road users. After research on the current state of the bridges and quay walls, it was found out that many of the civil structures were still good for another 30 years. Due to limited budget, sustainability goals, and the amount of work that really needs to be done soon, all maintenance that is not needed in the upcoming 30 years is postponed. This changed the area-based approach to a condition-based approach, where the current state of bridges and quays determines if they will be scheduled for maintenance in the upcoming years.

Due to the cndition-based approach, maintenance projects are not clustered together but spread over the city, causing hindrance to road users on many different locations. Given the complex urban road network of Amsterdam, this makes it more difficult to foresee how the effects of different maintenance projects interact and how the projects can be planned efficiently. Especially for bridges it is important to coordinate maintenance in a way that hindrance is minimised. Bridges are important road links in the urban road network of Amsterdam, connecting different parts of the city with each other. In general, bridge closures cause more additional travel time than the closures of roads due to longer detours. Maintenance on quay walls on the other hand causes much less hindrance to traffic, since often parallel roads exists. Additionally, the development of new maintenance methods makes it unnecessary to close the road for traffic.

Closing bridges in a random order could impose high costs on road users, particularly if bridges that are each others alternative are closed together or if roads with multiple bridges have to be closed multiply times. However, current maintenance planning methods lack the capacity to minimise hindrance to road users effectively. Despite ongoing research aimed at optimising maintenance, efforts to reduce traffic hindrance have been insufficient. Most literature only focuses on finding the optimal time for maintenance based on the state of the civil structures. When traffic is taken into account, only shortest paths are considered. With high demands in a complex road network, this measure will not be sufficient. In practice, the Municipality has been doing research on optimising maintenance of bridges and quays. In a collaboration with TNO, a quay maintenance planning optimisation tool has been developed using simulation. Both the technical aspects of the civil structures and the traffic effects were taken into account. The use of TNO's own simulation tool is very promising, since the simulation is much faster than current traffic simulations. With simulation, the effect of closures is better estimated than by calculating the travel times using shortest path. However, network effects were neglected when evaluating multiple closures at the same time, which could lead to non-optimal results. In order to reduce the road user hindrance caused by maintenance on bridges, an optimisation method is needed with a better assessment of the effect of the maintenance on traffic.

1.3. Research approach and scope

Within this research, a new maintenance planning optimisation method using traffic simulations will be proposed. The goal of the method is to better incorporate the evaluation of effects of combined closures in the planning of maintenance, in order to reduce the additional travel time caused by bridge maintenance. The existing traffic simulation model of TNO for macroscopic static assignement will be used in addition with the created optimisation algorithm for this research.

The current problem of the bridges in Amsterdam will serve as a case study to design this new optimisation method for urban maintenance projects. The scope includes the bridges under supervision of PBK, meaning that bridges not accessible for vehicles are excluded. The reason for excluding quays is that new developed maintenance methods for quays will not cause significant hindrance to road users and often don't create long detours. Bridges on the other hand are often important connectors between different parts of the road network and therefore have a big impact on hindrance during maintenance. The road users included in this research are cars, freight and cyclists. Public transport is excluded from the research since it has different behaviour and more complicated constraints than the other modes. Additionally, the simulation used in the research only assigns car, freight and bicycles to the network. For this research, maintenance is considered as renewal or renovation works. Only full closures of bridges will be taken into account. This research will not be focused on predicting maintenance. It is assumed that the selection of bridges for maintenance based on their state has already been made. Different maintenance speeds are assumed, with higher costs for faster maintenance. The optimisation method will schedule bridge maintenance in a way that it minimises the additional travel time for road users, while balancing maintenance costs for the municipality.

The objective of this research is therefore to *optimise bridge maintenance planning by minimising additional travel time and improving the assessment of the effect of combined closures*. Below the main research question and the sub questions contributing to achieving the objective are formulated.

How can additional travel time caused by bridge maintenance be minimised combining traffic simulation and optimisation?

Sub research questions

1. What are the current methods for optimising bridge maintenance and similar infrastructure projects in literature?
2. What is currently known about optimising bridge maintenance in practice and what are the needed improvements?
3. How can bridge maintenance planning be optimised while minimising both additional travel times and maintenance cost for the municipality?
4. How can the additional travel times caused by simultaneous maintenance of several bridges be taken into account using simulations?

1.4. Societal and scientific relevance

The societal relevance of this research is the contribution to reducing the hindrance experienced by road users due to maintenance in urban areas. With a growing need to address the maintenance backlog not just in Amsterdam, but across the Netherlands, the number of maintenance projects is expected to increase significantly in the coming years. A more accurate assessment of hindrance in maintenance planning can help reduce additional travel times caused by all these maintenance projects for all different kinds of road users.

From a scientific perspective, this research addresses a gap in the existing literature where optimisation is rarely combined with traffic simulations due to computational challenges. The consideration of hindrance in road network maintenance optimisation remains limited in current studies. However, the development of faster simulation models opens new possibilities for their integration into optimisation. Despite these advancements, these simulation models have not been incorporated in optimisation with a good assessment of simultaneous road closures. This study contributes to the literature by proposing an improved method for combining these simulations with optimisation of bridge maintenance. In addition, the developed optimisation helps to balance minimising hindrance and maintenance cost in bridge maintenance planning. The findings provide a foundation for further research on optimising maintenance in urban road networks, with the goal of minimising hindrance to road users.

1.5. Thesis outline

First a literature review is conducted to answer the first sub question in chapter 2. Next, the second sub question is answered in chapter 3, providing a practical perspective on bridge maintenance planning in Amsterdam. Based on the current maintenance planning process and the already developed maintenance optimisation tool for quay wall, the needed improvements for maintenance planning in Amsterdam are stated. In chapter 4, the methodology of this research is given. The provided optimisation and estimation methods answers sub question 3 and 4. The application of the methodology for a test case can be found in chapter 5, of which the results will be discussed in chapter 6. The research will end with conclusions and recommendations for further research.

2 Literature review

The aim of this chapter is to review the state-of-the-art literature on bridge maintenance and other similar road network maintenance optimisation to explore the current used methods and identify the gaps in literature. To achieve this, a comprehensive literature review has been conducted, focusing on the optimisation of maintenance planning of bridges and other similar road constructions. First, the relevant studies and used methodologies will be discussed, followed by an exploration of the use of a genetic algorithm as a method for maintenance optimisation. Based on this, the identified literature gaps will be presented. These identified literature gaps highlight the areas this thesis intends to contribute to. The presented findings in this chapter will answer the first research question: *"What are the current methods for optimising bridge maintenance and similar infrastructure projects in literature?"*

2.1. Current maintenance optimisation methods in road networks

To explore the state of the art in literature on maintenance optimisation in road networks, a broad literature review has been conducted. The review extended beyond just bridge maintenance, incorporating research on similar maintenance projects in road networks to gain a comprehensive understanding of existing methods in maintenance optimisation. To initiate the broad literature research, the following search query was used in Elservier's abstract and citation database Scopus: *optimisation AND (maintenance OR "construction work") AND (bridge OR road OR infrastructure OR network) AND (traffic OR transportation) AND (planning OR scheduling)*. Scopus then returned 484 results based on studies with article titles, abstracts or keywords similar to the words in the search query. Based on the title, articles have been selected for further reading. Through backward and forward snowballing, more interesting articles have been included in the literature review.

The analysis of the literature studied is divided into papers specifically optimising bridge maintenance and papers solving optimisation of maintenance of pavement, which has many similarities with the bridge optimisation problems. The focus of the literature research was on the different methodologies and to see if and how road users were incorporated in the optimisation problems.

2.1.1. Bridge maintenance optimisation

There are numerous papers which have studied the optimisation of bridge maintenance based on the state of the bridges, given budgetary constraints. For example the study of Alsharqawi et al. (2021), which proposed a budget optimisation for the short- and long-term maintenance, repair and replacement (MRR) plans for bridge decks. For the optimisation, a performance model developed to define the current condition of bridges and predict their future deterioration rate. Using a genetic algorithm, optimal MRR plans were generated while maintaining a defined level of service and budget constraints. Many of these bridge maintenance studies did not include road users in their optimisation. This is also noticed by Alsharqawi et al., who mention the absence of user and external costs (e.g. user delay) as a limitation to their research.

Bukhsh et al. (2020) developed a framework for finding the optimal maintenance planning for a network of road bridges over a multi-year planning period, mainly focused on highways. The multiple objectives of the optimisation were maximising the performance level of bridges and minimising the maintenance cost by optimally planning the maintenance treatments of the bridges. User costs were not included in the objective function, but were used among other things to determine the priority of maintenance projects to select bridges for the multi-year planning. A Multi-Attribute Utility Theory module was used for ranking the bridges based on the preferences of the decision-makers. This module used performance indicators to quantify the socio-economic impacts of maintenance activities, one of which is user costs. These were incorporated by determining the user delay costs resulting from maintenance activities. The extra travel time due to speed restrictions was determined by the length of the working zone, the standard velocity and the reduced velocity due to maintenance. By multiplying the extra travel time with the average traffic per hour, the value of an hour of the user time and the duration of the maintenance activity, the total user delay costs were estimated. Given the bridge priority list and the budget constraint for the multi-year planning, the most important bridges that can be maintained within the available budget are selected for the maintenance planning. To find the optimal maintenance plan for the upcoming years, a genetic algorithm was applied. The planning provided the selected treatment and year for all selected bridges.

Zhang and Wang (2017) conducted a literature research on maintenance optimisation where limited resources have to be distributed strategically among all bridges to optimise the performance of the whole system. According to their literature research, the most mature and broadly understood decision method for selection is life-cycle cost (LCC) analysis of individual bridges. However, bridges are often connected in the same road network and should therefore be considered collectively as integral parts of the network. They found that only several studies have attempted to obtain the optimal bridge maintenance strategy by maximising the operational performance of a transportation network.
Zhang and Wang divided current network performance indicators into two categories: network topology-based and network functionality-based. In the first category, connectivity reliability is used to define the probability that there exists at least one path between Origin and destination (O-D) pairs. This method is often used when networks are disrupted by extreme natural hazards to evaluate post-disaster disconnected network performance. The second category is more appropriate in a strong connected network where the level of service reduces, even though the O-D pairs are still connected. Here flow capacity reliability and travel time reliability are used to assess the efficiency of transportation network functionality.
In their study, Zhang and Wang optimised bridge maintenance decisions under budget constraints, which integrates network traffic demand, bridge condition ratings, bridge capacity ratings and network topology. The goal of the optimisation algorithm was to prioritise bridge maintenance project selections at a point-in-time. They used a metaheuristic method to provide solutions to the mixed integer optimisation problem, namely the binary particle swarm optimisation algorithm. The global objective of the model is minimising total travel time, which is measured in total travel time of all the shortest paths between all possible O-D pairs in the network for all vehicles. While other studies of bridge maintenance modelled the bridge links of the network as either fully functional or completely closed, this study modelled local constraints imposed by reduced load capacity of deficient bridges.

Abdelkader et al. (2022) proposed a bridge maintenance plan optimisation model at both project and network levels. Their multi-objective optimisation determines which bridge components to repair, what intervention action to apply and when to perform the intervention action. Four possible interventions were defined: no intervention, minor repair, major rehabilitation and replacement. To create an optimal maintenance schedule over a multi-year planning horizon, an exponential chaotic differential evolution algorithm is used, of which the basic procedures are similar to genetic algorithm. This requires candidate solutions to be structured in the form of strings, as can be seen in Figure 2.1. The multi-objective optimisation used four objective functions to determine the optimal set of maintenance plans: maximisation of performance condition of the bridge, minimisation of agency and user costs, minimisation of duration of traffic disruption and minimisation of environmental impact. In this study, user cost is evaluated with respect to travel delay costs, vehicle operating costs and accident costs. Travel delay costs are calculated by looking at the additional time needed to cross the bridge with lower speed due to congestion delays and speed reductions in the work zone. Vehicle operating costs refer the additional costs for vehicle drivers as a result of the additional travel time on the bridge, such as fuel costs and tire wear. The accident costs are a result of the more dangerous situations on the bridge during

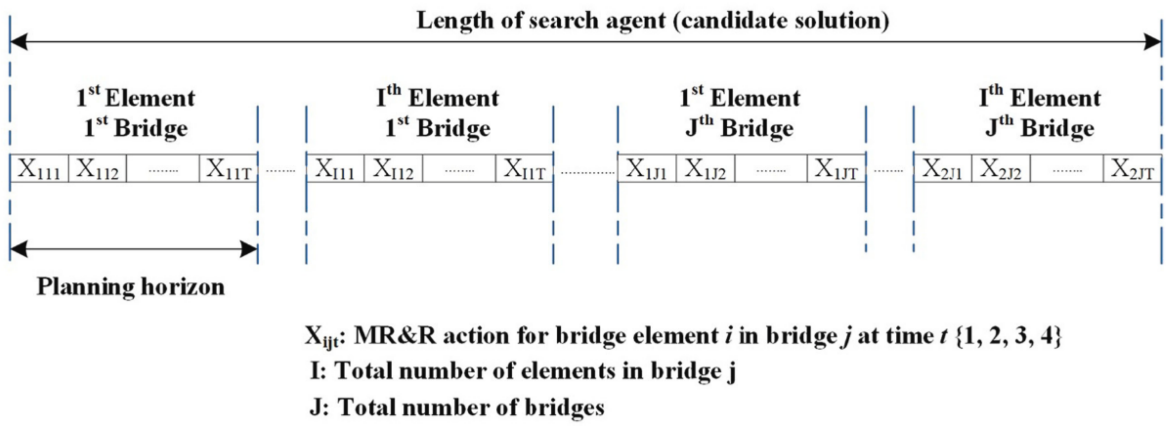


Figure 2.1: Solution structure for multi-year bridge maintenance planning optimisation using differential evolution algorithm (Abdelkader et al., 2022)

maintenance. This study thus only includes the user costs of drivers on the bridges themselves.

2.1.2. Pavement maintenance optimisation

In order to keep the literature review broad, in addition to bridges, studies on maintenance optimisation of other similar structures were also examined. In the field of maintenance, repair and replacement (MRR) optimisation, many papers have studied the optimisation of pavement maintenance. Just like with bridges, MRR optimisation aims to determine the optimal treatments for each pavement section at each time to improve the condition of the network within minimal budget (Fani et al., 2022).

Fwa et al. (2000) developed a genetic-algorithm-based procedure for solving multi-objective network level pavement maintenance programming problems. According to their research, the robust search characteristics and multiple-solution handling capability make genetic algorithms well suited for multi-objective optimisation. The three objectives included in their optimisation were the maximisation of the work production, minimisation of the total maintenance costs, and the maximisation of overall network pavement condition.

Recent studies have increasingly focused on the application of multi-objective optimisation in pavement MRR (W. Chen & Zheng, 2021). In their extensive literature review of research on multi-objective optimisation in pavement MRR planning, W. Chen and Zheng (2021) conclude that most research usually only focus on additional fuel consumption due to degrades in flatness when incorporating user costs. However, the work zone in MRR activities also causes delay costs, including queuing delays and route diversion, that are frequently ignored. Especially in urban road networks closing lanes for pavement maintenance and rehabilitation will often lead to delays and detours for vehicles since urban systems often already lack capacity to support the current heavy traffic flow. The evaluation of user delays during urban maintenance work is also more complex since alternative routes are different for each closed road. When delay costs were included in the research, the aim was often to calculate the additional emission (Choi, 2019; Galatioto et al., 2015). Galatioto et al. (2015) for example, used micro-simulation modelling of traffic to estimate the emissions caused by delays during maintenance for several traffic management options. These emissions were calculated to extend the system boundary of the life cycle assessment of road pavement. Meneses and Ferreira (2013) developed a multi-objective decision-aid tool with the two goals of minimising agency costs and minimising user costs. The agency costs include the cost of maintenance and rehabilitation activities. As user costs, only vehicle operation costs as a function of the pavement condition are included. This helped determining the optimal moment to plan maintenance to minimise pavement user costs. However, this does not say anything about effects of the maintenance work itself on user costs. One of their recommendations for further research is to also incorporate maximising the road network performance.

Gao and Zhang (2013) have pointed out that pavement MRR planning models usually distribute selected sections of pavement for MRR spatially across the network. By combining adjacent sections

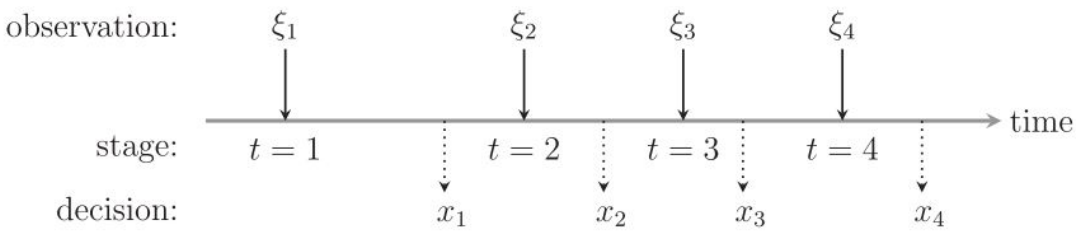


Figure 2.2: Outline of multi-stage stochastic programming problems (Fani et al., 2022)

with similar MRR needs in one single project, advantage can be taken of economies of scale. To do so, Gao and Zhang partitioned the network into smaller groups with similar MRR needs before using optimisation to allocate the MRR resources maximising system performance.

Although most models assume all of the model's parameters to be deterministic, some of the criteria in MRR are uncertain. Ignoring the uncertainty of these parameters, can lead to suboptimal solutions and unreliable pavement conditions (Fani et al., 2022). Budget fluctuates due to resource limitations and changes in government policies and pavement deterioration is complicated and stochastic. Fani et al. (2022) included this uncertainty of annual budget and pavement deterioration rate in their study on pavement MRR scheduling in large-scale networks. They formulated a multi-stage stochastic mixed-integer programming problem and solved it with the progressive hedging algorithm.
Figure 2.2 shows the outline of multi-stage stochastic programming problems. In their stochastic programming approach, uncertainty is defined by a set of discrete scenarios with corresponding possibilities of occurring. Decisions have to be made by taking the different possible future states in account. After decisions have been made, the uncertain parameters determine the outcome of the decision. An optimal solution provides a good outcome in all scenario's.

2.2. Genetic algorithm in maintenance optimisation

Optimisation methods can mainly be divided into two different categories: exact approaches and heuristic approaches (Rothlauf, 2011). Exact optimisation approaches, such as the Simplex and the Branch and bound method, guarantee finding an optimal solution to a problem. However, these exact methods suffer from computational complexity, so they can be time-consuming and computationally expensive, especially for large complex problems (TU Delft, 2024). For many real-life optimisation problems, using meta-heuristic algorithms is the main alternative to solve complex optimisation problems within reasonable time. These approximate optimisation techniques do not guarantee on finding the optimal solution, but they provide acceptable solutions. Network design problems are one of those real life problems that cannot be optimally solved for real-scale applications due to its complexity. Therefore, meta-heuristics can be applied, such as genetic algorithms (Cantarella & Vitetta, 2006).

The Genetic Algorithm is a heuristic algorithm often used to solve complex optimisation problems with large, nonlinear, and multidimensional search spaces by to exploring a wide range of possible solutions. This allows the algorithm to escape local optima and efficiently search through vast, complex solution spaces to find near-optimal solutions. It belongs to the larger class of evolutionary algorithms, inspired by evolution in nature where populations change over time by natural selection, reproduction and genetic variation. From a optimisation perspective, evolutionary algorithms can be seen as population-based randomised optimisation algorithms (Meyer-Nieberg et al., 2020). The right side of Figure 2.3 shows the evolutionary procedure of the GA. First a randomly generated set of feasible solutions is used as the initial population. The best functioning individuals based on the objective function are selected for reproduction. By applying the crossover process, new individuals are created with the combination of the parent's characteristics. Just like in nature, mutations happen during this process, improving the chances of finding better solutions. This cycle of creating new generations is repeated until an end condition is reached. During this process, the best found objective value gets better and better until a certain convergence is reached. On the left side of Figure 2.3, an example is shown of the convergence of a GA's objective value through many evaluations.

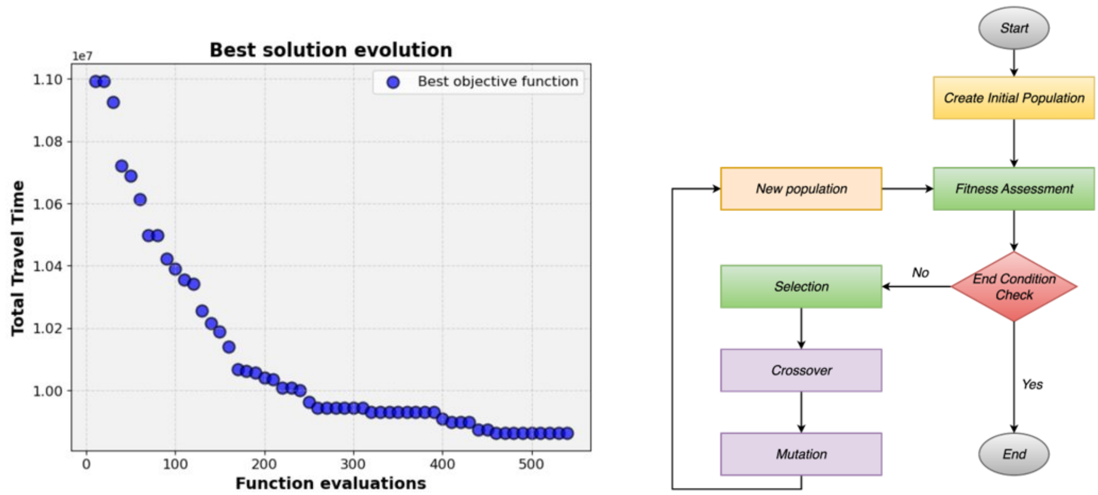


Figure 2.3: The basic procedure of the Genetic Algorithm. On the left side the convergence of the objective value, on the right side the evaluation procedure (TU Delft, 2024)

GA's have been widely used as a searched-based optimisation technique to develop accurate yet simple maintenance planning solutions constituting of multiple performance goals and budget constraints (Bukhsh et al., 2020). One of the first studies using GA for the optimisation of bridge maintenance was done by (Liu et al., 1997). In their research they emphasised the need for GA. One reason was that in bridge maintenance planning, selecting the optimal maintenance strategy is challenging due the number of possible combinations exponentially increasing with the number of bridges, the planning period and the number of maintenance alternatives. After this, many studies on optimising maintenance using GA followed, some of them shown in the literature review on maintenance optimisation in section 2.1 (Alsharqawi et al., 2021; Bukhsh et al., 2020; Fwa et al., 2000).

2.3. Identified literature gaps

This section will present the findings from the literature review to identify the literature gaps and answer the first sub-question *"What are the current methods for optimising bridge maintenance and similar infrastructure projects in literature?"*.

All MRR optimisation problems in the literature review try allocating limited resources over different maintenance projects to keep the network or structures functioning. To solve these problems, metaheuristics such as the genetic algorithm are often used to provide solutions that perform well on the optimisation objectives. Many studies focused on maximising the performance of the structures by selecting the best year and treatment for maintenance. However, only several studies have taken specifically the effect of MRR activities on road users into account in their maintenance optimisation. Often this only included the induced user cost on the bridges or pavement itself, such as the delays due to speed restrictions along the working zone during maintenance or the increased vehicle costs due to pavement deterioration before maintenance. Not many studies included the user costs of route diversion and delays due to maintenance. If they did, the additional travel time of diversion route was included by calculating the shortest paths between all O-D pairs.

Based on the literature review, it is evident that current literature on maintenance optimisation does not effectively take into account hindrance to road users caused by bridge maintenance for common modes in cities. Only vehicles have been considered in previous optimisaitons, but bicycles for example are a large part of traffic in a city like Amsterdam. Accounting for hindrance to road users is particularly crucial in complex urban road networks with high traffic demand, where bridge maintenance can result in significant delays and detours that are challenging to predict and evaluate. With many bridges to be renovated or renewed in such networks, it is important to assess the effect of simultaneous closures. To the best of the authors' knowledge, existing studies have only relied on calculating the shortest paths for vehicles to estimate the additional travel time across the network caused by maintenance. Despite the promising search properties of GA's for reducing computational time, no studies have been found that combine traffic simulations with bridge maintenance optimisation to better assess and minimise the additional travel time experienced by all road users. This research aims to fill this gap by developing a bridge maintenance optimisation using simulations to minimise additional travel time experienced by all road users in the system.

3

A practical perspective: bridge maintenance planning in Amsterdam

While much can be learned from existing literature, it is equally important to understand how maintenance planning has been implemented in practice. This research focuses on the maintenance planning of bridges and quay walls in the City of Amsterdam, providing a practical perspective on the subject.

This section provides an examination of maintenance planning by the program responsible for bridges and quay wall maintenance at the Municipality of Amsterdam, Programme Bridges and Quay walls (*Programma Bruggen en Kademuren*, PBK). Through the analysis of policy documents and multiple conversations with people within PBK, a comprehensive understanding of the current processes and developments in planning maintenance activities has been achieved. The opportunity to attend meetings and participate in discussions within the program offered valuable insights into the current challenges and desired direction for optimising maintenance.

First, a brief explanation of the functioning of PKB will be given. Next, it will discuss relevant developments, with particular emphasis on the research conducted by TNO in collaboration with the municipality to optimise quay wall maintenance. Finally, the chapter will conclude by identifying the current challenges and outlining the necessary improvements required for the effective scheduling of maintenance work by the municipality. By doing so, this chapter will answer the second sub-question: *"What is currently known about optimising bridge maintenance in practice and what are the needed improvements?"*

3.1. Bridge and quay wall maintenance in Amsterdam

The municipality of Amsterdam is responsible for 1800 bridges and 600 kilometre of quays (Gemeente Amsterdam, 2023). Many of those bridges and quays are more than a century old and therefore not designed for the heavy loads and intensive use of today. Moreover, only reactive maintenance of bridges and quays has been done in the past decades due to not enough budget going to maintenance of infrastructure. After multiple incidents of collapsing quays, the municipality decided more work should be done to ensure safety. To investigate and renovate the bridges and quays in the city on a large scale, PBK, a specific programme for Bridges and Quays has been established in 2019.

3.1.1. The task of the programme Bridges and Quay walls

With the start of the new programme, PBK takes over part of the responsibility of the municipal organisation *Verkeer en Openbare Ruimte* (V&OR, translated: traffic an public space)(Programma Bruggen en Kademuren, 2020). V&OR is normally responsible for the management of all civil structures within the city, including bridges and quays. The major challenge to catch up on overdue maintenance and ensure safety of the bridges and quays was too big of a challenge on its own beside the other responsibilities of V&OR. Thus, PBK was established as a separate programme to take over the renewal of the bridges and quays to ensure safety as soon as possible. Additionally to the renovation and renewal

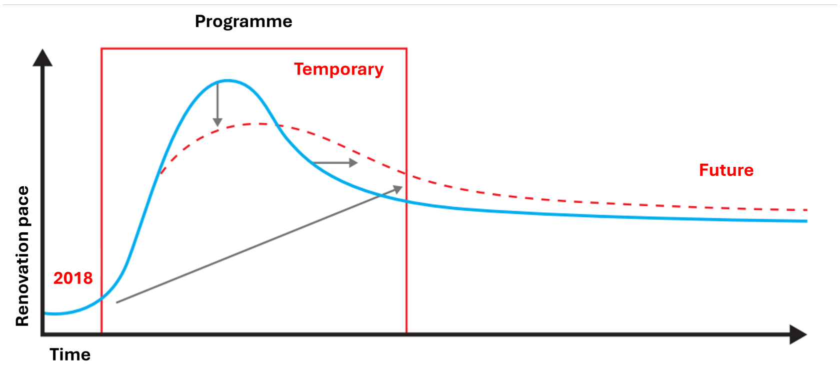


Figure 3.1: Increase in renovation pace over time: Starting at a low renovation pace of 1 to 2 bridges and 500 meter of quay walls, the renovation speed had to be increased to a sustainable renovation pace of 8 bridges and 2 kilometre of quay walls. To catch up on overdue maintenance, this would lead to a temporary peak in maintenance. New methods in bridge and quay wall maintenance spread out the anticipated peak, shown by the dotted line. *Images from the programme policy plan, adapted by author (Programma Bruggen en Kademuren, 2020)*

works, they would also focus on investigation and gaining knowledge about the state and degradation of these structures within the city. Within the scope of PBK are around 850 bridges and 200 km of quays. These are selected based on their importance for the city's accessibility and on their risk of safety problems. This selection includes only bridges where motorised traffic passes. Bridges only for cyclists or pedestrians are not the scope of PBK and still the responsibility of V&OR. The programme is supposed to be temporary, meaning that as soon as the major task of catching up on overdue maintenance has been completed, V&OR takes over responsibility again for all bridges and quay walls.

3.1.2. The initial area-based approach

The first priority of the programme was to keep the bridges and quays safe and prevent any more collapses and sinkholes at any cost. Many safety measures were taken, such as sheet piling and closing bridges and quays for heavy traffic. The most important bridges and quays were scheduled first for maintenance, selected based on their known bad state or importance to the cities accessibility. However, at that time, not much was known about the current state of all bridges and quays or about the degradation of the structures. Based on standard calculation models from Rijkswaterstaat and the general assumption that civil structures have an average lifespan of 100 years, it was estimated that nearly all bridges and quays were nearing the end of their lifespan and required full renewal. This indicated a need for a significant increase in the stable rate of renovation to maintain these assets in good condition. However, due to insufficient funding for infrastructure maintenance, the renovation pace was limited to only 1 to 2 bridges and a maximum of 500 meters of quays per year before the PBK programme. In contrast, a healthy and sustainable renovation rate for the large number of assets would be 8 bridges and 2 kilometres of quay wall per year. To catch up on the overdue maintenance, the renovation rate had to be even temporarily increased to as many as 20 bridges per year. The programme would be responsible for catching up on this overdue maintenance, ensuring that, once cleared, a sustainable renovation pace could be maintained under the responsibility of V&OR. This temporary peak of maintenance projects can be seen in Figure 3.1.

The municipality of Amsterdam is committed to keeping the city safe, accessible and liveable. In addition to the PBK renovation projects, there are many other projects in the city already causing hindrance to road users. These include maintenance on the sewage system, installation of internet cables, expansion of the electricity network, and changes to street layouts under the '*Autoluw Amsterdam*' (car-free Amsterdam) program. Coordinating all these projects within the city's maintenance schedule while maintaining accessibility is a challenging task. The increased maintenance demands for bridges and quays would further strain the city's accessibility. Due to the importance and scale of the PBK project, it was given priority over all other maintenance and construction work within the city, requiring other projects to adjust their schedules accordingly.

Because it was thought that almost all bridges and quays had to be renewed, the priority of PBK

Figure 3.2: The 9 bridges of the project '*Oranje Loper*'. From left to right: bridge number 359, 135, 108, 167, 117, 63, 106, 22, and 8

allowed for an area-based approach. The idea of this approach was to only close a street or route once and make sure all of the maintenance needed in the upcoming years is done simultaneously. This is also referred to as 'working in corridors'. An example of this is the project called '*Oranje Loper*'. By scheduling the closure of all bridges in the project either simultaneously or in quick succession, the important and often used route would only be inaccessible for a minimal period. This would reduce the hindrance to the surroundings and traffic. Figure 3.2 shows the 9 bridges selected for maintenance in the project 'O*ranje Loper*'.

3.1.3. Knowledge acquisition

After maintenance on the first few bridges and quay walls, it was found that the state of many civil structures was better than expected. New methods were developed to investigate the state of the civil structures. For example, samples from the timber pilings used as foundation of bridges and quays were taken to determine the real constructive load-bearing capacity left. This was often much higher than conservative assumptions about the constructive load-bearing suggested. As a result, many assets did not have to be fully renewed, but could be renovated to extend the lifespan with at least 30 years. In particular, many of the bridges within the scope of PBK were expected to last at least 30 years, sometimes achieved by changing their function. A research done by PBK together with V&OR focused on this last option by investigating the impact of removing heavy traffic from bridges on the accessibility of the city (Van der Sluijs et al., 2023). In addition to looking at whether closures would lead to inaccessible parts of the city, they also ranked all bridges based on the additional travel time the closure caused to the system. For the determination of the additional travel time caused they used the Urban Strategy Tool of TNO.

Another interesting development was the new monitoring methods. In addition to trigonometric measurements, also satellites are used to measure the vertical and horizontal deformations. These are very helpful in maintenance planning, since an acceleration of the deformation is a good indication that the civil structures are soon in need of safety measures or maintenance.

Besides gaining knowledge about the state and degradation of the civil structures, new and more efficient methods for renovation were developed. In the beginning of the project in 2018, the renovation and replacement methods for quay walls were inadequate. The speed at which the quay walls were renovated was too slow. To renovate all quay walls in Amsterdam that way would take too long resulting in many emergency measures like steel sheet piles covering the beautiful historic quay walls. Moreover, the current methods resulted in long closures of the quays for traffic, noise and vibration pollution, and the necessary removal of houseboats and monumental trees (G-Kracht, n.d.). The Municipality issued a tender for the development of innovative methods to replace the quay walls faster, cheaper and with less hindrance or disruptions to the surroundings, called *Innovatiepartnerschap Kademuren* (IPBK, translated 'Innovation partnership quay walls'). New methods did not require to close of the roads for traffic.

3.1.4. Current method of scheduling maintenance

Currently, fewer bridges are selected for maintenance due to their state being better than expected and the implementation of alternative measures, such as function change. By avoiding unnecessary renewals before the end of a structure's lifespan, the approach is more sustainable. Unfortunately, this reduction in maintenance is also forced by a limited budget. For the renovation of 8 bridges and 2 kilometres of quay wall per year, an annual of 150 million euro is needed. However, the council has allocated only 83 million euro for the renovation project in their coalition agreement (Programma Bruggen en Kademuren, 2022). Given the significant amount of work required to maintain the city's safety and accessibility, maintenance that can be deferred for up to 30 years has been postponed. This decision helps spread out the initially anticipated peak in maintenance projects, as illustrated by the dotted line in Figure 3.1. However, these developments make it impossible to continue using an area-based approach for bridge maintenance. For the project '*Oranje Loper*' this also meant that not all bridges would be fully renewed anymore. Instead of selecting all bridges in one route for maintenance to reduce hindrance, bridges are now selected based on innovative monitoring and investigation methods. This is known as the condition-based approach.

Using the condition-based approach, the selected bridges and quay walls are included in the main schedule ('*hoofdprogrammering*'). The main schedule shows the strategic route for PBK for the next 10 years and is updated every 2 years. It is used for strategic planning with Stadsregie and within PBK (Jongejans, 2023). A more detailed schedule, which identifies the bridges and quays selected for maintenance in the next 2 years, is established in the block scheduling ('*Blokpgrogrammering*'). Without the closing of bridges in corridors, the effect of the closures on traffic is less obvious. Due to the condition-based approach, selected maintenance projects can be spread throughout the whole city. Given the complex network with high travel demand, simultaneous bridge maintenance projects of different locations can lead to unexpected congestion and detours. Therefore, sometimes the traffic simulation model of Amsterdam (Het verkeersmodel Amsterdam, VMA) is used to calculate the effects of a certain combination of closures and test whether the configuration is possible (Verkeer en Openbare Ruimte, 2022). However, the VMA is very complex due to the size of network including the whole Amsterdam region and uses many iterations, resulting in long computation times. The traffic assignment already takes 4 hours to run, the full model run can take up to 1.5 days (Verkeer & Openbare Ruimte, 2015). This makes it difficult to evaluate the effects of multiple different configurations of road closures.

As part of the municipality of Amsterdam, PBK is dependent on the policy decisions made by the current coalition formed after every municipal election. The coalition not only determines the budget allocated for renovation projects but also sets the vision for how construction work should be managed within the city. Road construction work can cause various disruptions to the network(Municipal Executive of the Municipality of Amsterdam, 2021a). For instance, roads can be fully closed, have restricted passage, or be converted from two-way to one-way traffic, with optionally alternating directions. These obstructions impact various modalities: car, freight, bicycle, pedestrians, public transport and others such as emergency services. Not only the type of obstruction has to be chosen, also the duration of the obstructions should be determined. Currently, the municipality favours a 'short and heavy' maintenance approach. In the past, the goal was to minimise traffic disruption at construction sites, often by maintaining partial road access. However, the current practice generally involves full road closures. Although full closures cause more immediate traffic disruption compared to partial closures, they allow for quicker completion of work. Additionally, full closures are considered safer for construction workers and can reduce overall maintenance costs. The municipality has stated that in case of significant disruptions, the use of night shifts or weekend work has to be considered to shorten the obstruction. However, these alternative working hours are more expensive than regular hours, leading to higher maintenance costs. Balancing the benefits of reduced traffic disruption against the increased costs of accelerated work schedules is a complex challenge for the municipality.

3.1.5. A future towards optimisation

Within the municipality, increasing attention is being given to the potential of optimisation in maintenance scheduling, particularly through data-driven methods. Multiple research initiatives are currently focused on optimising the maintenance of bridges and quays, each exploring various aspects of the topic. In collaboration with AMS (Amsterdam Institute for Advanced Metropolitan Solutions), several research directions have emerged. Optimisation of maintenance planning was mostly focused on finding the best maintenance intervention and year of execution based on the state of the structures.

At TNO a few researchers saw an opportunity in using TNO's fast digital twin called the Urban Strategy Tool (UST) for the optimisation of bridge and quay wall maintenance in Amsterdam. In a collaboration with PBK based on a mutual exchange of knowledge, a first version of an optimisation of quay wall maintenance planning tool was developed, which will be discussed in the next section. Within PBK, there is growing enthusiasm for the concept of the fast digital twin of TNO and there is significant interest in further refining the optimisation tool. Seeing many possibilities in using the UST for their maintenance planning, PBK plans on funding the further development of TNO's maintenance planning optimisation.

3.2. Quay wall maintenance optimisation tool of TNO

At TNO a planning-maintenance module for quay walls in Amsterdam has been developed to support the decision-making process for cities. The focus of this project was to combine traffic effects and the technical aspects of the civil structures to be maintained in one module. That way, a planning can be made that is both cost-effective and causes as little hindrance as possible to services and inhabitants as possible. Using an optimisation algorithm, different configurations of maintenance are generated and selected based on performance indicators for traffic effects and technical aspects of civil structures.

The planning-maintenance module is built in the UST framework, which is also developed by TNO and can be used to simulate traffic on network level very quickly. Due to parallel programmed algorithms and state-of-the-art hardware and lightweight models, the computation can be done within minutes. The traffic module of the UST uses the VMA as a basis, copying its road network and OD matrices. Since UST is very fast in computing traffic effects, it is extremely suitable for testing the traffic impact of many configurations of maintenance.

The objective of the optimisation was minimising city impact, the risk of failure and the maintenance costs. In order to optimise on the city impact, risk and maintenance cost, the impacts on the city and the risk of the quays are translated to costs. To determine the costs for impact on the city, the traffic model of UST is used. Within UST the effect of maintenance is computed for 1 morning peak period, consisting of 2 hours. This effect is multiplied with the time period for which the maintenance is happening. The additional travel time for cars and freight, or vehicle loss hours, caused by the road closures are multiplied with the value of travel time corresponding to car and freight. In addition to delay, also emissions caused by vehicles are considered within the optimisation.

For each year, five measures were considered in the maintenance schedule: Do nothing, close quay wall for parking, close quay wall for traffic, a lifetime extension or a full renovation. Closing the quay wall for parking and traffic are both methods to reduce the pressure on the quay wall to hold off renovation or renewal. After life time the quay wall is expected to be good for another 30 years, and after full renewal for 100 years. For lifetime extension and full renovation, the road of the quay wall was assumed to be closed during the year of maintenance. For every quay wall in the optimisation, first the effects of all possible actions were simulated using the UST. Based on the state of the quay wall and predicted possible future states, the optimal plan for each individual quay wall was generated. In the last step, the individual plans were coordinated using a hard constraint to apply renovation actions in the same quay wall cluster in consecutive years.

This optimisation focused on optimising quay wall maintenance by selecting the best year for certain measures. The result is an automatically generated sequence of actions with time steps of one year for every quay wall for 20 years. Figure 3.3 shows the coordinated planning for a subset of the optimised quay walls.

While the optimisation is focused on minimising city impact by simulating the additional travel time caused by maintenance using the UST, there are some important limitations in addressing hindrance to road users. Lifetime extension and full renewal are planned in a certain year and the closure of the streets are assumed to have a duration of the full year. There is no detail in how multiple maintenance projects in one year can be configured in a smart way to reduce hindrance. Moreover, only vehicles and freight are included in the research, while cyclists have a large share in the traffic in Amsterdam.

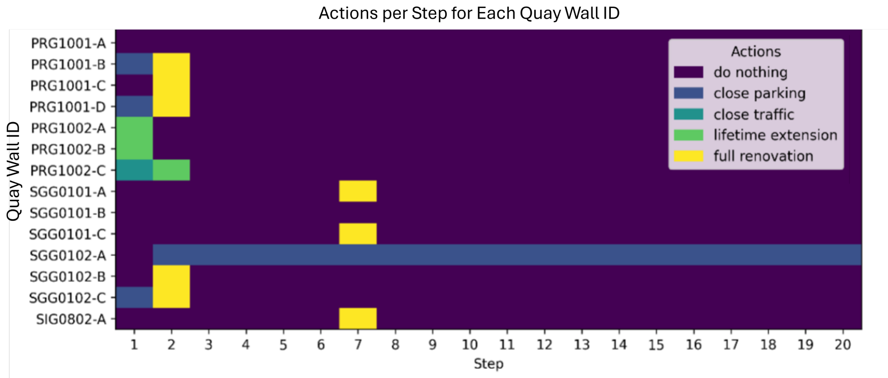


Figure 3.3: Coordinated planning per quay wall per year (showing a subset of the optimised quay walls of the TNO research)

However, the most important limitation of the research is how simultaneous closures of roads are assessed. In this version of the quay wall maintenance optimisation tool, the effect of combined closures is estimated by summing the effect of individual closures. This estimation method neglects network effects, possibly leading to sub optimal solutions.

3.3. The need for improvements considering maintenance planning in Amsterdam

In this last section, the second sub-question will be answered: ”What is currently known about optimising bridge maintenance in practice and what are the needed improvements?”

Amsterdam is facing an increased pressure on the accessibility within the city due to many maintenance projects closing of parts of the road network. With the new task for PBK, more bridges than ever before have to be renovated or renewed in the upcoming years. The condition-based approach for selecting maintenance causes bridge maintenance projects to be spread across the whole city causing hindrance on many different routes. Since bridges serve as crucial links in the road network and typically have longer detour routes, their complete closure due to 'short and heavy' maintenance can significantly increase the travel times for road users. In cases of severe traffic disruptions, the municipality considers more costly maintenance executions to reduce the traffic impact, such as reducing the duration by employing more costly work hours per week. The municipality therefore needs to weigh the traffic impact of maintenance plans with their costs.

However, with high impact projects spread throughout the whole city and a complex network with high traffic density, is it difficult to foresee the effect of maintenance planning on hindrance caused to road users. Currently, maintenance planning still heavily relies on expert judgement to reduce hindrance to road users, as the VMA is too slow to evaluate numerous potential solutions. In the quay wall optimisation of TNO, the faster traffic simulation of the UST was only used to asses the effect of individual closures. By summing these individual closure effects to estimate the effect of simultaneous closures, network effects were neglected. Additionally, the quay wall optimisation did not consider the detailed configuration of multiple maintenance projects within the same selected year. Given the importance of bridges in the road network, it is all the more important to take into account the effects of combined closures when optimising their maintenance planning, since the wrong combinations of closures could lead to a lot of additional travel time.

Based on the above, the municipality is in need of a better scheduling of bridge maintenance to reduce hindrance caused to road user. Given the limitations in the current scheduling, the bridge maintenance optimisation needs to:

- Minimise additional travel time for more than only cars as road users
- Assist in weighing the traffic impact of maintenance plans with their costs
- Consider the effect of simultaneous bridge closures
- Provide more detailed configurations for maintenance in the upcoming years

4 Methodology for bridge maintenance planning optimisation using simulations

This section explains the chosen methodology for this research. First, the basic concept of bridge maintenance optimisation using simulations is introduced to provide a foundation for understanding the following sections. Then, the selected meta-heuristic for optimisation, a genetic algorithm, is explained into detail. Following that, the simulation UST and its use are described. Finally, after covering both the simulation and the optimisation separately, a look will be taken at how these components can be combined. This results in a new proposed estimation method for the effect of simultaneous closures. The proposed method for optimising bridge maintenance aims the last two sub-questions: "*How can bridge maintenance planning be optimised while minimising both additional travel times as well as maintenance cost for the municipality?*" and "*How can the additional travel times caused by simultaneous maintenance of several bridges be taken into account using simulations?*"

4.1. Optimising maintenance planning using simulation

The objective of this research is to optimise bridge maintenance planning in order to minimise additional travel time by improving the assessment of the effect of combined closures. With optimisation, the goal is to find the best solution from a set of candidate solutions (El-Halwagi, 2006). Figure 4.1 shows a conceptual design of candidate solutions considered in this research. It is assumed that the long-term planning has already made the selection of bridges to be scheduled for maintenance in the short-term planning for the upcoming few years. Additionally, it is assumed that the long-term planning has decided on the type of maintenance work, such that only the execution plan for each maintenance project in the short-term planning has to be determined. The execution plan determines the duration of the bridge maintenance. The short-term planning optimisation thus needs to determine for every bridge in the defined bridge set when maintenance should take place and how long the execution should be. This means that the starting date, duration and thus costs of maintenance can vary.

To evaluate the quality of each solution and see which is best, an objective function is used. This objective function, e.g. costs, can be minimised or maximised depending on the problem. It is also possible to have multiple objectives which can even be conflicting (L. Chen & Bai, 2019). This is also the case for the municipality. On the one hand, the municipality wants to minimise hindrance to road users, while on the other hand they want to keep maintenance cost as low as possible due to limited budget. To minimise hindrance to road users, all maintenance should have minimal obstruction of the road traffic, leading to as short as possible maintenance or keeping the flow of traffic as high as possible on the construction site. To minimise maintenance cost, however, it would be better to execute maintenance without the use of expensive working hours or additional safety measures. The optimisation should assist the municipality with their decision making and provide better solutions balancing these different objectives.

Given the complexity of the problem, which involves multiple bridges, many potential starting points

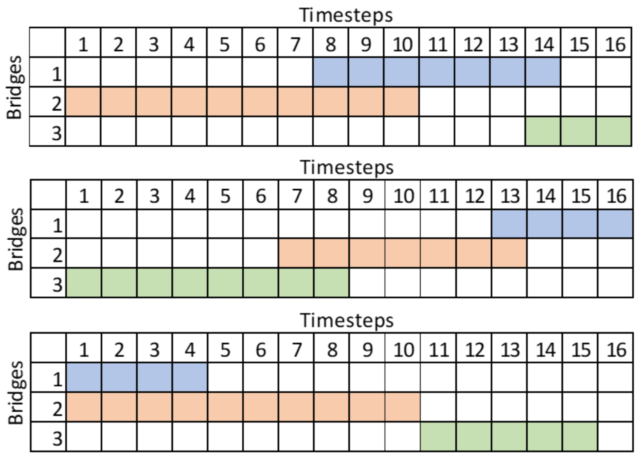


Figure 4.1: Conceptual design of a set of possible bridge maintenance plannings. The colours represent the three different bridges. For every bridge, different starting points and durations are possible

for maintenance, and various maintenance execution options, the number of possible solutions quickly runs into the millions. This makes finding the optimal solution through exhaustive search impractical. Additionally, the evaluation of the solutions is difficult due to non-linear traffic dynamics. In such cases, as explained in the literature review, meta-heuristic methods such as Genetic Algorithms (GA) can be used. GA is a powerful optimisation technique that has been effectively applied in similar research contexts, which has been covered in section 2.2. It effectively searches for better solutions by simultaneously exploring multiple solutions and generating better solutions over time. Section 4.2 will explain more about this chosen optimisation method.

For the evaluation of solutions it is important to accurately determine the traffic hindrance caused by bridge closures due to maintenance. Traffic simulations play a key role in quantifying the additional travel time resulting from these closures. This research uses the simulation tool UST developed by TNO to model the urban traffic network and assess the impact of different maintenance scenarios on travel time, which will be discussed and elaborated on in section 4.3.

By combining optimisation methods and traffic simulations, this research aims to develop an effective bridge maintenance planning strategy that balances the dual objectives of minimising travel disruption and reducing maintenance costs. Figure 4.2 shows how the genetic algorithm and simulation work together to generate and evaluate solutions with the goal of finding the optimal bridge maintenance schedule. The operation will be explained in more detail in the following sections.

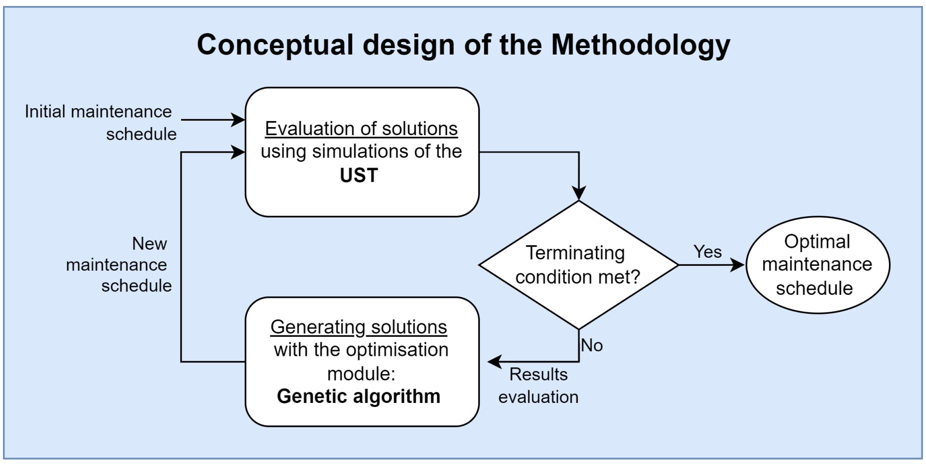


Figure 4.2: Conceptual design of the methodology

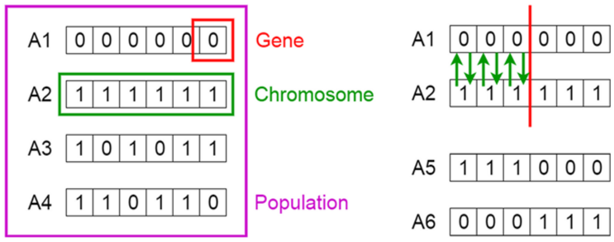


Figure 4.3: The representation of solutions using chromosomes and the process of cross-over (Mallawaarachchi, 2023)

4.2. Optimisation: Genetic Algorithm

In genetic algorithms, the coding of solutions for the population is inspired by how genetic information is stored in DNA. A chromosome consisting of a series of symbols, usually a string of numbers or characters, represents the decision variables of a potential solution. Chromosomes can exist of binary, integer, or real-valued vectors, which can be seen as the genes. Figure 4.3 shows the representation of a solution consisting of only binary variables. The process of evolution over time by natural selection, reproduction and genetic variation is imitated by selecting solutions and applying cross over and mutation. During the cross over process, parts of the chromosome between two parents are exchanged. In the mutation process, a gene can be selected for a random change in value. It is important that the construction of the chromosome is chosen carefully, since small changes in the chromosome should result in small changes of the fitness function (Meyer-Nieberg et al., 2020). In the following subsections, the chromosome design and evaluation are shown.

4.2.1. Chromosome design

As stated before, for every bridge (b) in the bridge set (B) a starting point and maintenance duration choice need to be made. Thus, the chromosomes containing the decision variables for possible solutions have a gene identifying the starting point (s_b) and a gene for the selected maintenance duration for every bridge (m_b). Figure 4.4 shows how a possible schedule is transformed in a chromosome.
The time period wherein all maintenance should take place can be split up in days, weeks, months or other time steps. The genes representing the starting point for every bridge have a lower bound of 1 and an upper bound of the maximum time step within the maintenance period (N_t). The genes representing the maintenance duration choice have a variable bound based on the number of different maintenance duration options considered. Every maintenance duration choice, corresponds to a certain duration and cost. The options can be the same for all bridges, or be bridge specific. Since all bridges need to have a starting point and a maintenance option selected in the chromosome, it is made sure that all bridges get assigned maintenance during the time period. Both the starting point and the maintenance option can only be integers.

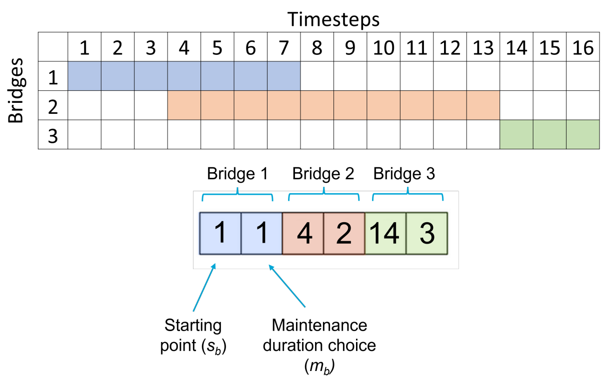


Figure 4.4: Design of the chromosome for a possible schedule

4.2.2. Evaluation of solutions

To evaluate every solution generated by the GA, an objective function is used. As stated before, the municipality has contradicting objectives. The municipality wants to minimise hindrance to road users, but also keep maintenance cost as low as possible. To deal with these two conflicting objectives, both are included in a single optimisation's objective function using a weight. Equation 4.1 shows the objective function (Z), containing a factor for the maintenance costs (C) and a factor for road user hindrance (H) with a specific weight (w).

$$min(Z) = w \cdot H + C + P \tag{4.1}$$

Also a penalty (P) is included in the objective function for maintenance exceeding the time period given for maintenance. Instead of using a penalty, a constraint could be used to prevent the GA of choosing a starting point in combination with a certain maintenance option that would result in completion of maintenance after the latest possible time step in the period. However, using a penalty will help guide the optimisation to a better solution and keeps the option of scheduling maintenance after the given time period for when nothing else is possible.

Penalty calculation

The penalty for a certain solution is calculated by multiplying the number of time steps exceeding the time period with a certain penalty value (p). For every bridge the end timestep is calculated by adding the duration of the chosen maintenance option (d_{m_b}). The highest end time step in the schedule is found by taking the maximum of all end time steps for all bridges. This results in the following equation:

$$e_{\text{max}} = \max_{b \in B}(s_b + d_{m_b}) \tag{4.2}$$

The penalty gets higher for every time step exceeding the maintenance time period. The number of time steps exceeding the period can be calculated by subtracting the number of time steps within the maintenance period (N_t) from the highest end time step in the schedule (e_{max}). Of course no penalty is given when there are no timesteps exceeding the maintenance period. This results in the following equation:

$$P = max(0, e_{max} - N_t) \cdot p \tag{4.3}$$

Maintenance cost calculation

For determination of the maintenance cost of a certain schedule, the cost for every duration option of maintenance should be known (c_{m_b}). The total costs for maintenance can then be calculated by summing the maintenance cost for every bridge for all bridges, see the following equation:

$$C = \sum_{b=1}^{B} c_{m_b} \tag{4.4}$$

Hindrance cost calculation

The hindrance to road users is measured in additional travel time caused by the maintenance closures. As can be seen in Figure 4.5, a schedule consists of multiple unique situations in terms of traffic impact. Every unique situation (u) has a different effect on the additional travel time for the road users caused by the maintenance closure(s). The frequency of the unique situation determines the total amount of additional travel time in one unique situation ($TATT_u$). The total hindrance to road users for a certain schedule (H) can then be captured by the sum for all unique situations, following the next equation.

$$H = \sum_{u=1}^{U} TATT_u \tag{4.5}$$

As explained before, the effect of traffic cannot be easily calculated for complex road networks. To determine the additional travel time of each unique situation, traffic simulations of the UST of TNO are used. First a deeper look in the UST is needed before looking into how these simulations can be used in the optimisation process.

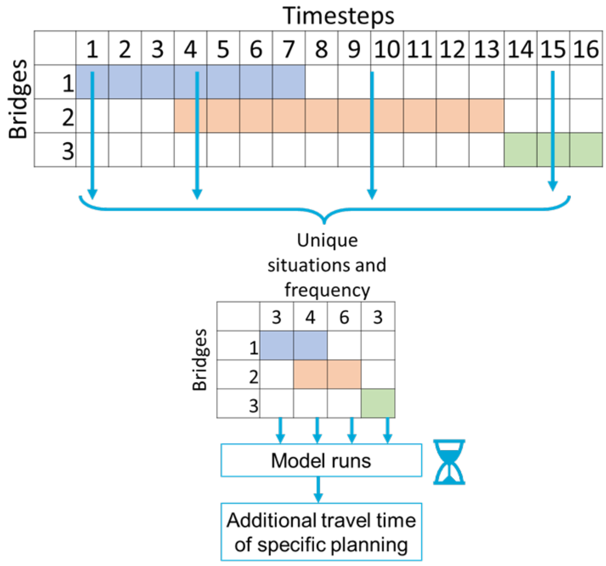


Figure 4.5: Unique situations in a maintenance schedule in terms of traffic impact that need to be evaluated

4.3. Simulation: Urban Strategy Tool

To determine the additional travel time of each unique situation in terms of traffic impact in bridge maintenance schedules, the UST developed by TNO will be used. This section will first provide a short description of this digital twin. Next, the traffic simulations within the tool are explained, followed by the possible interventions. Finally, it will be shown how these simulations can be used in calculating the additional travel time of maintenance closures.

4.3.1. A digital twin of Amsterdam

Scenario-based analysis helps stakeholders study the effects on, for example, traffic intensities, air quality, and noise before and after an intervention. However, many existing software packages do not provide the integral view of interrelated domains needed to make such complex and time-consuming strategic decisions. In a case study for Amsterdam, Lohman et al. (2023) created a digital twin platform called 'Urban Strategy'. A digital twin is a realistic digital replica of the real world where data, analytics and visualisation are combined. Because of its virtual representation, digital twins are the perfect environment to test different interventions without having the consequences of those actions in real life (Madni et al., 2019). Moreover, Digital twins often use lightweight models. By simplification of high-fidelity and complex models, the behaviour of these models can be quickly studied with minimal computational costs and enough fidelity to answer the right questions (Madni et al., 2019).

The Urban Strategy Tool (UST) is an integral and interactive multi-modal approach for urban planning and consists of multiple different models, which are combined with data and visualisation. One of these models is the Traffic+ model for the traffic assignment for cars, bicycles and freight, which is used for this research. In their research, Lohman et al. (2023) used the Traffic+ model to study the effects of temporary road closures for construction work. They emphasised the importance of studying the joint effects on traffic when simultaneous construction work takes place at multiple locations within the city.

Due to parallel programming and state-of the-art hardware, the traffic model in UST is much faster than the traditional transport model VMA. A simulation run only takes 3 minutes, while the traditional traffic model VMA takes hours for a full run. This increases the potential to use the simulation for optimisation. In the optimisation of quay wall maintenance for the city of Amsterdam done by TNO, the UST is already used to see the effects of changes in the road network due to maintenance on traffic and noise- and air pollution (Swaalf et al., 2023). For this research, only the Traffic+ model will be used to measure the effects of maintenance on hindrance to road users.

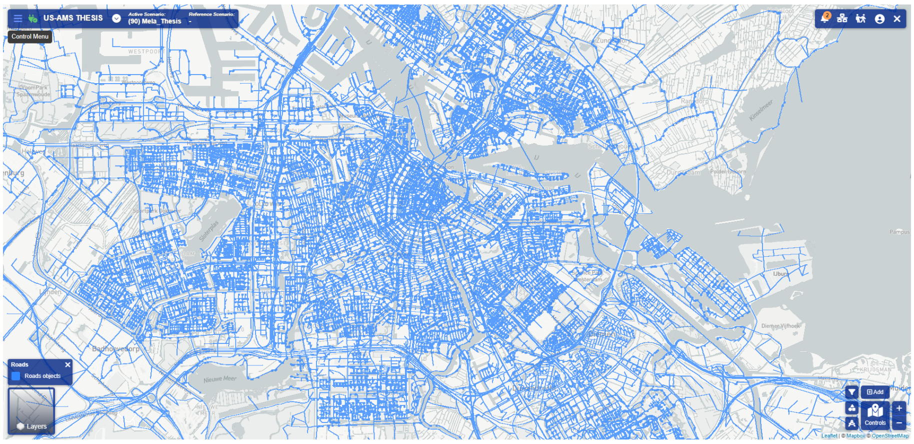


Figure 4.6: Overview of the road network of Amsterdam in Urban Strategy Tool

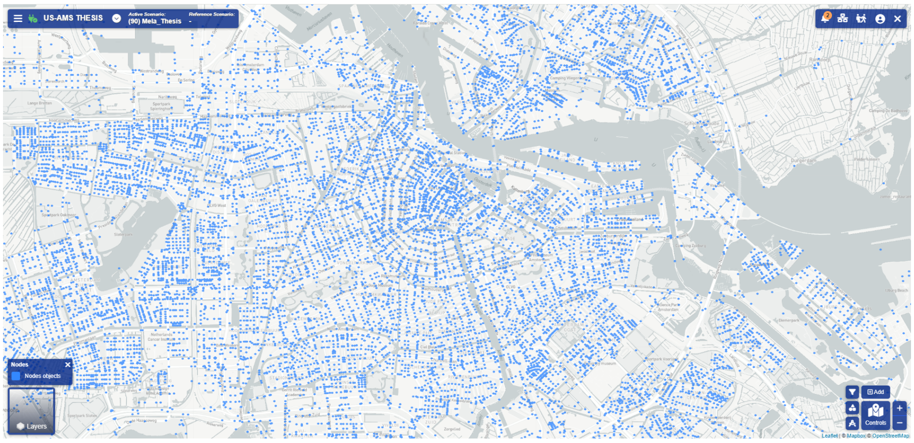


Figure 4.7: Overview of zones of Amsterdam in Urban Strategy Tool

4.3.2. Traffic simulation within UST

The Traffic+ model is built upon the existing VMA model 4.0, using historical data from 2019 (Verkeer en Openbare Ruimte, 2022). From the VMA model the whole road network of Amsterdam is already implemented, as can be seen in Figure 4.6. The road network also consists of roads outside of Amsterdam, to correctly model traffic using the roads within the municipality (Verkeer en Openbare Ruimte, 2022). For every road in the system, the free-flow speed for both directions, the length and the capacity per time period is given (TNO, 2023).

For the traffic assignment, the VMA's origin-destination (OD) matrices are imported. The OD matrix contains the frequency of all passenger trips starting from a postal code (origin) and ending at another postal code (destination) for a certain time period. The time period used in this research is the morning peak, consisting of two hours. All different postal codes, or zones, of the OD matrices can be seen in Figure 4.7.

Based on the number of trips form the OD-matrices, Traffic+ allocates car, freight, and bicycle traffic to the network using a static traffic assignment. The static assignment assumes that the number of trips on a link will be constant during the considered time period of time, representing the average conditions for the specific period. The model can be used to simulate the route-choice of existing trips and also allows to explore the impact of infrastructure alterations on route choice. Two different ways of assigning traffic to the network are used: All-Or-Nothing and Volume Averaging.

The Traffic+ model uses All-Or-Nothing assignment for freight and cyclist, assuming everyone drives the shortest route. Within All-Or-Nothing, the shortest route is determined by travel time, which is a direct result of the roads distances divided by the roads free-flow speed. This method ignores the effects of congestion, but is therefore a quick way to determine the shortest paths. Car traffic is assigned to the network using a Volume Averaging method, meaning that traffic is distributed over different routes using an equilibrium assignment accounting for congestion. Volume averaging is an iterative method, where each iteration the loaded trips onto the network influence the travel times on the road links. To determine the travel time on each link, the capacity and speed is used. If the intensity increases on a certain road, delays will occur due to approaching the capacity. This relation between travel time (T), intensity (I), capacity (C) and free-flow travel time (T_0) is captured in the BPR function, developed by the American Bureau of Public Roads and used in UST to calculate travel time, see Equation 4.6. Alpha and beta are road type specific function coefficients which are imported from the VMA.

$$T = T_0 \cdot \left(1 + \alpha \left(\frac{I}{C}\right)^{\beta}\right) \tag{4.6}$$

The first iteration starts with an AON assignment. After multiple iterations using the BPR, an equilibrium is reached where no car can be better of by taking another route. To increase the speed of the simulation, the number of iterations after which the assignment is terminated used in the VMA is reduced from 30 to 10 in the traffic assignment of UST. This reduces the accuracy of the model, but keeps it functional for many applications by still providing simulations with enough fidelity for many applications.

4.3.3. Interventions in the road network due to maintenance

Maintenance on bridges impacts the road network. To see the effect of these interventions on traffic, changes should be made to the road network in UST. The road network of Amsterdam is split up into different road segments in UST. For every road segment, mode specific road links store the mode specific data. The input data are the capacity and the maximum (or free-flow) speed for both directions. The output data computed by the Traffic+ model is also stored for every mode specific road link. This consists of the intensity and travel time for both directions.

The input data for each mode specific road link can be changed by creating and activating *controls* in UST. These controls are temporary data adjustments, which can be enabled and disabled when needed (Curley et al., 2024). Controls can be made in the web interface of UST, or by using a REST API. For this research, TNO has provided useful python code to send and retrieve information to the UST. By changing the input data of the road segment, the following maintenance interventions could be simulated:

- Full road closures: By setting the speed and capacity to zero for all modes in both directions, a road can be fully closed, such that no traffic can pass.
- Mode specific road closures: Only closing the road for a certain mode or modes, such that other modes can still use the roads. This can also be used to mimic the use of temporary bicycle bridges being built next to the bridges under construction.
- One-way traffic: Only closing the street for one direction, the other direction can still use the road.
- Reduced passage: Traffic can pass in both directions, but the capacity is significantly reduced. Reduced passage can also be used to model alternating traffic, where only one direction at a time is allowed to pass. This is typically controlled by traffic lights or traffic controllers.

After a change has been made to the road data, the Traffic+ model starts computing the new intensities and travel time for all road links. The intensities show how many vehicles or bikes use the specific road link during the time period of the simulation. The travel times show the time it takes to cover the specific road link. Within the web-interface, the effects can be seen by comparing the scenario with change to a selected base scenario. Figure 4.8 shows the effects on the total intensity after closure of the Rijckerbrug in the city centre of Amsterdam. Using the REST API, the output data for all road links can also be retrieved. This data is used to calculate the effects of maintenance on additional travel time.

Figure 4.8: Changes in the total intensity after closure of the Rijckerbrug in the UST interface

4.3.4. Calculating additional travel time resulting from bridge maintenance based on the traffic simulation

The output data for all road links retrieved using the REST API can be used to compute the additional travel time in the whole system after an intervention on a bridge. The additional travel time of unique situation of closures can be computed by the difference between the travel time in the system in the base scenario and in the scenario with maintenance. To compute the travel time in the whole system (TT) for a certain scenario, the travel times spend on every mode specific link should be summed, see Equation 4.7. For ever mode specific link i,the computed intensities I_i are multiplied with the corresponding computed travel time (t_i,) to compute the total travel time spend on that road link during the time period. To compute the additional travel time (ATT) during the simulated time period, the difference between the total travel time spend in the scenario with interventions TT_u and the total travel time spend in the base scenario TT_0 is calculated, see Equation 4.8.

$$TT = \sum_i I_i \cdot t_i \tag{4.7}$$

$$ATT_u = TT_u - TT_0 \tag{4.8}$$

The additional travel time (ATT_u) calculated only reflects the impact of maintenance during the simulated period, which is a 2 hour morning peak for this research. A study of the Dutch travel behaviour indicates that the 2 hour morning peak accounts for approximately 20% of the daily mobility (Cloïn, 2013). For the purpose of this research, it is assumed that the travel behaviour in the morning peak is representative for the entire day and that the daily additional travel time can also be used for weekend days.
The total additional travel time for a unique situation ($TATT_u$) can thus be calculated by first scaling the ATT_u of the morning peak to the entire day, extrapolating it to the duration of one timestep in days (d_t), and multiplying it by number timesteps corresponding to the duration of the unique situation (d_u):

$$TATT_u = ATT_u \cdot \frac{100}{20} \cdot d_t \cdot d_u \tag{4.9}$$

4.4. Integration of UST with the GA optimisation

In the previous section it is shown how the additional travel time of a certain unique situation can be computed using the UST traffic simulation. However, the direct use of simulations in the evaluation of optimisation such as GA can be computational difficult due to the large number of simulations and often long simulations times (Meyer-Nieberg et al., 2020). To find out how the UST simulations can be integrated with the GA, three different options are explored: Simulation based optimisation, precomputations and the use of meta-models.

4.4.1. Simulation based optimisation

With simulation based optimisation, simulation is used to evaluate every solution candidate provided by the optimisation approach (Meyer-Nieberg et al., 2020). The optimisation model chooses the new input parameters for the simulation model. The simulation model uses these input parameters to return the performance measures, which are used to steer the optimisation. Every evaluation this loop is repeated.

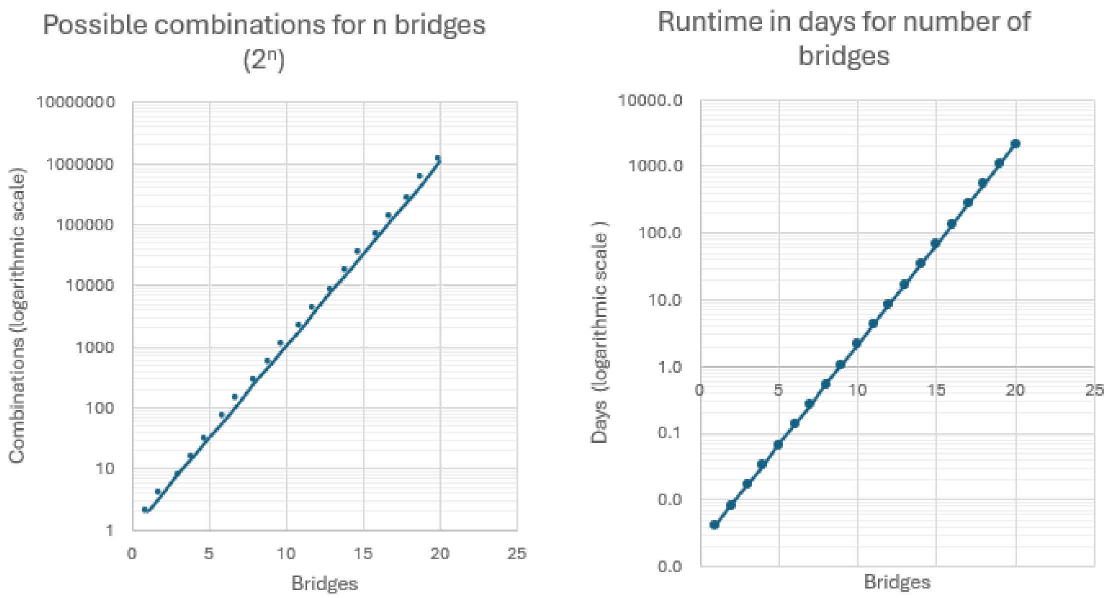


Figure 4.9: The exponential increase of combinations and corresponding computation time

The problem with simulation based optimisation is that it can be computational hard. Especially with approaches like the genetic algorithm, where several solutions are tested at each generation. Many simulation models take a long time to do a full run. Even though the UST is very fast compared to other traffic simulations, it still takes a while to compute the effect on traffic for a certain schedule. As can be seen in Figure 4.5, a schedule consists of multiple unique situations. To calculate the effect on traffic of the whole schedule using the UST, every unique situation must be run separately. One run in the UST takes around 3 minutes. In the simple schedule example shown in Figure 4.5, already 4 runs are needed. With a simulation time of more than 10 minutes, it would take way too long to calculate all different solutions of the optimisation model. Therefore, simulation based optimisation is unfeasible for this research.

4.4.2. Precomputations

Another way of using the simulations of the UST to optimise the maintenance planning of bridges is to precompute all possible combinations of bridges. That way the optimisation algorithms does not need to run multiple simulations every evaluation. Instead, it looks up the additional travel for all unique combinations in the planning. This would speed up the genetic algorithm, but precomputing all combinations can take a lot of time too. Computation time can easily get out of hand as the possible combinations increase. The number of all possible combinations of all sizes for a set of n values, can be calculated with 2^n. In Figure 4.9 the exponential increase of combinations can be seen as well as the corresponding runtime of those combinations in UST with an runtime of 3 minutes per combination.

With only 15 possible closures, full precomputations would take more than 2 months. When multiple types of closure for each bridge are considered, this increases the number of possible combinations even more. So for small bridge sets full precomputations could be an option, but for cases with more possible closures, precomputations are no option.

The number of precomputations could be reduced by limiting the numbers of bridges that can be in maintenance at the same time. If only a certain number of closures at a time are allowed, the number of possible combinations can be computed with the binomial coefficient. This coefficient gives the number of combinations of subset size k out a set with size n. The binomial coefficient is defined by Equation 4.10 (Olive, 1979). For example, if only five bridge closures at a time are allowed, the combinations for all sizes up to and including five need to be computed according to Equation 4.11. For the subsets from 2 to 5 bridges the runtime graph can be seen in Figure 4.10. This graph shows

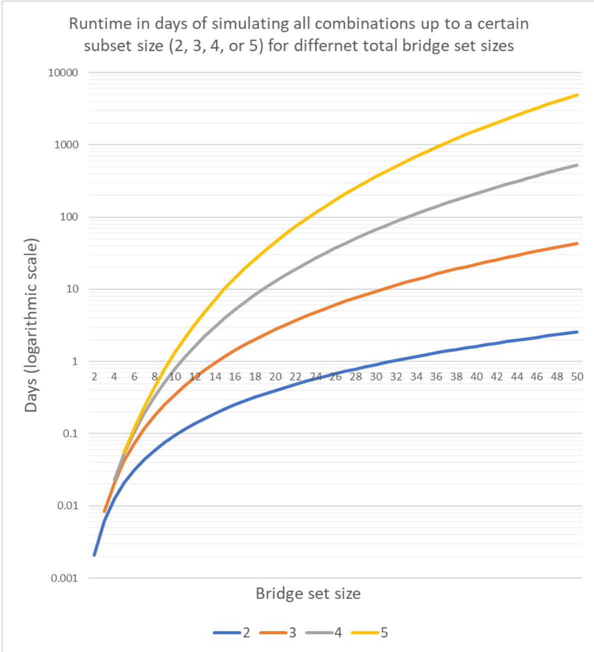


Figure 4.10: Runtime in days of simulating all combinations up to a certain subset size (2,3,4 or 5) for different total bridge set sizes

that also computing all combinations with a maximum bridges simultaneous in maintenance easily gets out of hand. Furthermore, setting a maximum on the number of simultaneous closures would restrict the algorithm in finding the optimal solution.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{4.10}$$

$$\sum_{i=0}^{k} \binom{n}{i} \tag{4.11}$$

4.4.3. Meta-models

The computational difficulty of simulation based optimisation as explained before, has resulted in the introduction of meta-models or surrogate models to reduce the computational load. Meta-models and surrogate models are terms for approximation models. Especially population-based algorithms are often coupled with approximation models (Meyer-Nieberg et al., 2020). Evolutionary algorithms operate with a population of solutions and it commonly takes many generations before finding a good solution. By using approximation of the simulation outputs for the evaluation of solutions, computational times can be reduced. One way of using meta-models is to use the approximation as a complete substitution of the original simulation model for the fitness evaluation of solutions. This is called functional approximation, where an alternative expression is constructed for the objective function (usually called fitness function in evolutionary computation) (Jin, 2003).

It should be noted that one concern with using approximation models for the fitness evaluation is that it is difficult to construct a model that is globally correct. It will be very likely that the evolutionary algorithm will converge to a false optimum, meaning that with the original fitness function it will not be an optimum (Jin, 2003). However, the original fitness function based on simulations also has some error in the evaluation of the solutions, since the simulation is an abstract representation of the system (Meyer-Nieberg et al., 2020). Nevertheless, using an approximation of the simulation to determine the effects of candidate solution in the optimisation is a good start to get to better solutions for the municipality.

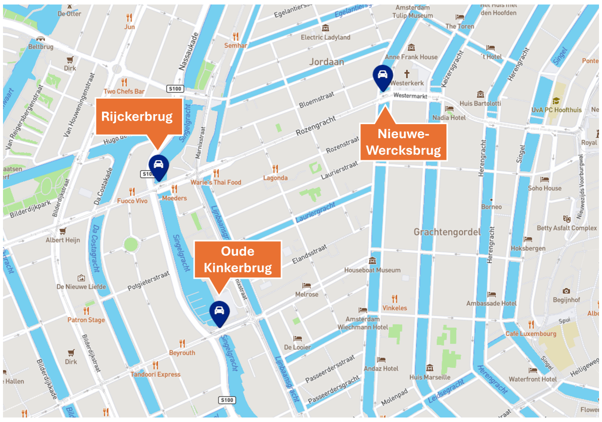


Figure 4.11: Location of the 3 bridges: Rijckerbrug, Nieuwe-Wercksbrug and Oude Kinkerbrug

However, it is important to assess how well this approximation method performs. Therefore, the first step is to evaluate the approximation method used by TNO in its quay wall maintenance optimisation.

A simplistic meta-model using only simulations of individual closures

The researches of TNO working on the optimisation tool for quay wall maintenance planning also encountered the problem that optimisation using the UST would be computational difficult. Within their project they considered 25 different closures. From the beginning, direct simulation based optimisation was disregarded since every simulation took about 1 minute. With 25 possible closures the number of combinations would get out of hand very fast. Due to these the computational limitations they decided to only simulate every road closure separately. The stored results for every road closure were used to compute the effect of combination of closures. For a combination of closures, the individual closure effects were added up. This disregards the potential dependencies between road closures in different locations.

The summation of individual closures for the estimation of combined closures can be regarded as a very simple meta-model for the simulation. However, Lohman et al. (2023) already emphasised the importance of studying the joint effects on traffic when planning simultaneous construction work at multiple locations within a city. With the summation of the individual effects, the different closures are assumed to be isolated from each other in the system, neglecting network effects. This can lead to wrong conclusions for the optimal maintenance schedule. For some combined closures, the effect on traffic is expected to be smaller than the summation of the individual closures. An example is two roads in line of each other. On the other hand, closing a road on the detour of another closed road would lead to an larger effect than the sum of two closures.

To demonstrate the effect of neglecting the potential dependencies between roads, this is tested on three bridges in Amsterdam: the Rijckerbrug, Nieuwe-Wercksbrug and the Oude Kinkerbrug (see Figure 4.11). As can be seen, the Rijckerbrug and the Nieuwe-Wercksbrug are located in the same street. Most traffic going over the Rijckerbrug would also go over the Nieuwe-Wercksbrug and vice versa. As expected, Table 4.1 shows an overestimation of the effects on the additional travel time for all of the 4 categories when the Rijckerbrug and Nieuwe-Wercksbrug are closed simultaneous. When looking at the combined closure of the Rijckerbrug and the Oude Kinkerbrug in Table 4.2, an underestimation of the effect can be seen.

By using this summation method, or isolation method, it does not matter which bridges are combined, as the individual effect of all closures is not dependent on other simultaneous closures. This could result in solutions where bridges are closed together, when in reality the combination of the closures leads to worse outcomes than the sum of individual closures. Moreover, the opportunity of closing consecutive bridges is not taken advantage of. To better incorporate the minimisation of road users hindrance in the optimisation, this simple meta-model will be improved by taking network effects into account.

Table 4.1: Difference in additional travel time in the system between simulation and summation when closing the Rijckerbrug and Nieuwe-Wercksbrug for the total system and for car, freight and bicycle only (in hours during the morning peak)

	Additional travel time (h)		Difference (%)
	Simulation	Summation	
Total	46.5	62.4	+34.3 %
Car	21.7	27.8	+28.2 %
Freight	5.8	7.4	+27.8 %
Bicycle	18.9	27.1	+43.3 %

Table 4.2: Difference in additional travel time in the system between simulation and the summation method when closing the Rijckerbrug and Oude Kinkerbrug for the total system and for car, freight and bicycle only (in hours during the morning peak)

	Additional travel time (h)		Difference (%)
	Simulation	Summation	
Total	59.4	48.0	-19.2 %
Car	21.1	20.8	-1.3 %
Freight	9.1	8.0	-12.1 %
Bicycle	29.2	19.1	-34.4 %

4.5. The proposed method for estimation of combined bridge closures effects

Neglecting the dependencies between closures results in non-optimal schedules with respect to hindrance reduction. Since hindrance reduction is important for the municipality in planning maintenance, a new method is needed to better take the network effects into account. Therefore, it is important to understand how bridge closures impact each other, without having to do many simulations runs. Almost all bridges in the road network of Amsterdam are in some way connected to each other and thus do not function in full isolation. The closure of one bridge will have a direct or indirect effect on other bridges. This means there is always a certain form of interdependence between the bridges. The proposed method uses the interdependencies between bridge pairs to improve the estimation of the network effects of combined closures. This section first shows how these interdependencies between bridges can be captured. Then, the use of these interdependencies in the estimation of combined closures shown.

4.5.1. Capturing the interdependencies between bridges

Multiple options have been tested to capture the interdepenencies between bridges. A first idea was to use shared OD-pairs. During the last step in traffic models, trips of OD pairs are assigned to a certain route. During the traffic assignment in the VMA, information is provided about the volume of the OD-pairs passing over a certain road link. With this information the percentage of shared OD-pairs can be computed. Bridges with a high percentage of shared OD pairs will have a similar impact if closed. If the commonality based on the shared OD pairs is low, no relation between the bridges can be assumed and the individual travel times can just be added up. Unfortunately, the information about the OD volumes per link is not stored in the UST during the simulation, unlike in the VMA. Therefore, the shared OD-pairs cannot be used to compute the commonalities between bridge closures.

Another option considered was the use the true distance between bridges as an estimation for the interdependencies between bridges. However, this does not say anything about the way the bridges are connected in the network. This again could lead to wrong conclusions due to neglecting network effects.

One thing that is possible to retrieve from the simulation runs is the effect of a certain closure on other bridges. After the simulation of a closure, the complete road database can be retrieved. This matrix contains all the intensities and travel times for all links in the system, so also for the other bridge links in need for maintenance. That way it is possible to see the effect of the closure of bridge A on the

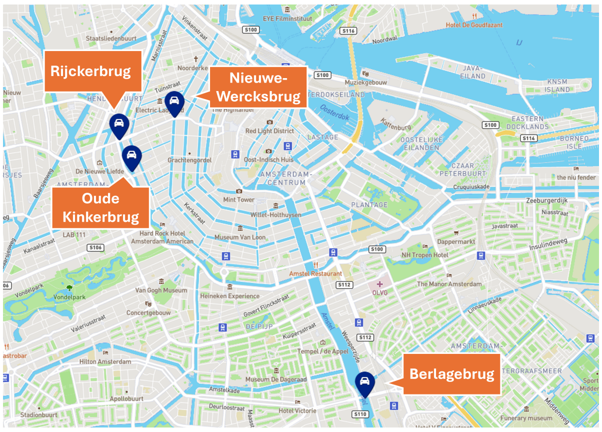


Figure 4.12: Location of the 4 bridges: Rijckerbrug, Nieuwe-Wercksbrug, Oude Kinkerbrug and Berlagebrug

Table 4.3: The additional travel time for the individual closures of the Nieuwe-Wercksbrug, Rijkersbrug, Oude Kinkerbrug & Berlagebrug

	Initial travel time in the system (h)	Additonal travel time for each closure (h)			
		Nieuwe-Wercksbrug	Rijckerbrug	Oude Kinker-brug	Berlagebrug
Total travel time	228412.11	32.43	29.77	15.33	184.24
Travel time car	115966.15	15.36	12.29	5.25	30.80
Travel time freight	20893.00	4.32	2.89	3.49	2.08
Travel time bike	91552.95	12.74	14.60	6.59	151.35

intensity and travel time on bridge B. If bridge B's intensity decreases due to the closure of bridge A, it can be said that these bridges have a negative interdependence. Closing them together is expected to be better for the total additional travel time, since traffic is already reduced. Otherwise, if the intensity on bridge B increases due to the closing of bridge A, it would not be wise to close them off together, since bridge B serves as an alternative bridge for bridge A. To look into these interdependencies based on the intensity and travel time change, the same 3 bridges as before are investigated. The bridges in the previous used bridge set are all close to each other, therefore a fourth bridge that is located in a different part of the city was added. Figure 4.12 shows the location of the previous bridge set together with the new selected bridge, the Berlagebrug.

First, all individual closures are simulated, which results in the additional travel times that can be found in Table 4.3. There are different ways of looking at the effect of the individual closure on the other bridges. For both the intensity and travel times on the road links the absolute and percentage change can be calculated for every mode and for all traffic in total. It can be seen in the individual closure results that the effect of closing the Berlagebrug is far greater than closing one of the other three bridges. This can be explained by the fact that the Berglagebrug is an important bridge in the Amsterdam traffic network. Even if the effect on the Berglagebrug of another closure is small, this could lead to larger absolute change in intensity. This could lead to wrong conclusions about the interdependencies between the bridges, thus it is better to look at the percentage change. Since the objective of the optimisation is to minimise additional travel time, the travel times on the link are used instead of the intensity. Table 4.4 shows the percentage change in total travel time on the bridges after an individual closure. The choice has been made to use total travel time instead of one of the specific mode travel times. As can be seen in the percentage change table, the table is almost symmetrical. The effect of closing bridge A on bridge B is almost the same as closing bridge B on bridge A. To work with one interdependence value for all bridge pairs, the average of the two values is computed. Table 4.5 shows the interdependency values of all bridge pairs.

Table 4.4: The percentage change in total travel time on a bridge after the closure of another

	Closing bridge			
Effect bridge	Nieuwe-Wercksbrug	Rijckerbrug	Oude Kinkerbug	Berlagebrug
Nieuwe-Wercksbrug	-	-44.61%	6.75%	-0.003%
Rijckerbrug	-45.66%	-	29.20%	0.0067%
Oude Kinkerbrug	16.66%	33.77%	-	0.0305%
Berlagebrug	0.0010%	-0.0003%	0.00008%	-

Table 4.5: The interdependencies between bridge pairs captured by the average percentage change in total travel time after closure of one bridge of each pair

	Rijckerbrug	Oude Kinkerbrug	Berlagebrug
Nieuwe-Wercksbrug	-45.138 %	11.703 %	-0.001 %
Rijckerbrug	-	31.485 %	0.003 %
Oude Kinkerbrug	-	-	0.015 %

4.5.2. The use of interdependencies between bridges in the estimation of combined closures effects

By looking at the change in travel time on a bridge after a closure of another, the interdependencies between bridge pairs are computed. A negative percentage change would mean that closing the other bridge too would lead to less additional travel time than closing them separately. A positive percentage change would mean that closing the other bridge too would lead to more additional travel time than closing them separately. The larger the absolute percentage change, the more important it is to take into account the relation between these bridges when calculating the additional travel time of the combined closure. When evaluating a unique situation in a candidate solution of the algorithm, at least the effects of the bridges with the highest absolute relation should not just be summed up.

To estimate the effect of multiple bridge closures together based on the relations between the bridges, an option would be to use a function. This is however very difficult to fit due to the network effects. Traffic simulations are a better way of estimating the joint effect. Since the runtime of all combinations of two closures for even a bridge set of 50 stays within a few days (see Figure 4.9), these simulations can be used in the evaluations. To use those simulations in the evaluation of the algorithm, in addition to all individual closures, all possible combinations of two must be precomputed. An new estimation method has been proposed that effectively uses the simulations of all combinations of two closures in the assessment of the effect of multiple simultaneous bridge closures, based on the interdependency values. The process of the proposed interdependency evaluation method of unique situations can be seen in Figure 4.13 and is explained in detail below.

Potential maintenance schedules consists of multiple unique traffic situation due to bridge closures. All these unique evaluations have to be assessed separately. Based on a given unique situation, the number of simultaneous bridge closures is determined. These simultaneous closures form the set of closures to be evaluated. When the set is larger than two, the bridge pair with the highest absolute interdependency value is searched. Since it is most important that these bridges are evaluated considering their interdependency, the additional travel time based on their combined closure is added to the travel time of the unique situation. After this, the assessed bridge pair is removed from the set of closures to be evaluated. If there are still two or more bridges left to evaluate, the same process of searching the bridge pair with the highest interdependency value is repeated. If one bridge is left over, the simulation of the individual closure is added to the unique situation's additional travel time. After the whole process has been completed, the additional travel time of the unique situation is found.

There are still some limitations to this method. Take for example the three bridges located close to each other: the Rijckerbrug, Nieuwe-Wercksbrug and Oude Kinkerbrug. If in a certain situation they are evaluated together, the Rijckerbrug and Nieuwe-Wercksbrug will be selected for a combined simulation evaluation first, since they have the highest absolute interdependency (-45.1), see Table 4.5. This

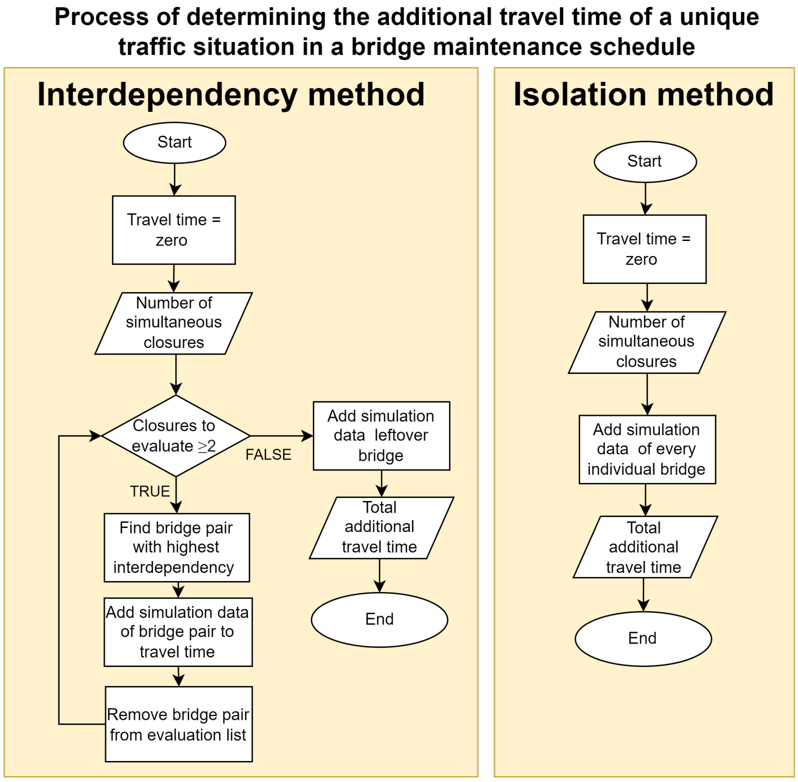


Figure 4.13: Process of determining the additional travel time of a unique traffic situation in a bridge maintenance schedule with the interdependency and isolation method

leaves the Oude Kinkerbrug to be evaluated. Even though this bridge has a very high interdependency with the Rijckerbrug (+31.5) and a positive interdependency with the Nieuwe-Werckbrug (+11.7), it's effect will just be summed with the effect of the Rijckerbrug and Nieuwe-Wercksbrug together. This creates the illusion that the Rijckerbrug is isolated of the other bridges. Since both closures result in a positive percentage change of travel time on the Oude Kinkerbrug, it will be better to not have the Oude Kinkerbrug in maintenance together with the other two bridges. However, due to the summation of the effect, there will be no difference between the evaluation of the three bridges together and the evaluation with the Oude Kinkerbrug separately. The same applies to the situation where a third bridge is negative correlated with both bridges, resulting in a missed opportunity if the bridge is not closed simultaneously.
Even though this method will not be able to perfectly evaluate the additional travel time of multiple simultaneous closures, it is expected to perform better than using only summation. If so, it will be a better alternative to capture the hindrance caused to road users in terms of additional travel time. Using this method in optimising maintenance planning of bridge closures would lead to better results regarding the Genetic Algorithm. To test this, the evaluation with summation will be compared to the evaluation with interdependencies.

4.6. Overview of the proposed methodology

This chapter has proposed a method for optimising bridge maintenance considering hindrance to road users. In this last section an overview is provided by showing the overview and answering the sub-questions related to the methodology. The overview of the proposed method for optimising bridge maintenance considering hindrance to road users can be found in Figure 4.14. Next, the last two sub-questions are answered. The third sub-question was: "*How can bridge maintenance planning be optimised while minimising both additional travel times as well as maintenance cost for the municipality?*". To answer this question, a optimisation objective has been proposed that consisted of both minimising additional travel time as minimising the total costs for the execution of the maintenance

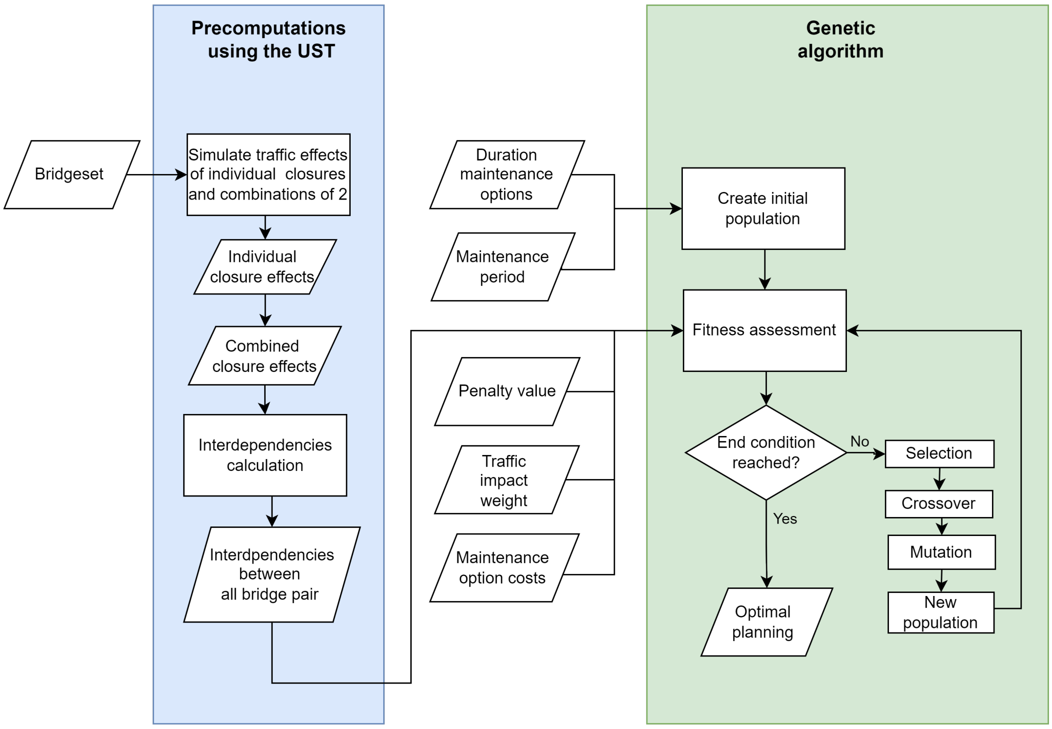


Figure 4.14: Overview of the proposed method for optimising bridge maintenance considering hindrance to road users

plans. A genetic algorithm was used to generate maintenance plans and evaluate possible solutions. To determine the additional travel times, traffic simulations from the UST are used for an accurate estimation of the overall additional travel time in the network. The maintenance costs were dependent on the chosen execution speed. By including a weight for the effect on traffic, the balance between the two parts of the objective can be found using the preferences of the decision-maker.

Next, the fourth sub-question was answered, which reads as follows: ”*How can the additional travel times caused by simultaneous maintenance of several bridges be taken into account using simulations?*”. To answer the last sub-question, first the possibilities of using simulation in optimisation have been analysed. This resulted in the conclusion that meta-models had to be used in order to keep computational times low. A too simplistic meta-model, such as estimating the effect of simultaneous closure by assuming isolation of bridges, could lead to wrong conclusions for combining maintenance projects. An improved estimation method has been proposed that uses interdependencies between bridges to improve the assessment of the combined closure.

To determine the quality of the methods proposed to answer the sub-questions, the methodology first needs to be tested. In the following chapters, the application on a test case of Amsterdam is introduced, after which the results will be examined. The quality of the proposed methodology will be discussed in the final conclusions.

5

Application of the methodology: a test case of Amsterdam

In this section the proposed methodology for optimising bridge maintenance using simulations of the UST is applied to a test case. First the definition of input values will be given, containing the bridge set, maintenance period, maintenance durations and maintenance cost. Then, the parameters used in the GA are stated. Lastly, the generated controls for simulating the closures and the precomputations results will be shown.

5.1. Definition of input values

In this section the problem specific input values will be show. First, the bridge set and maintenance period will be defined for which the short-term maintenance planning has to be made. Then, the chosen maintenance duration options with corresponding cost for this application will be explained. Lastly, the parameters used in the optimisation using the genetic algorithm will be given.

5.1.1. Bridge set and maintenance period

As explained in the subsection 3.1.4, a certain set of bridges is selected for maintenance for a period of 2 years based on the state of the bridges. This selection of bridges is taken as a given for the short-term maintenance optimisation algorithm. The set of bridges is input for the algorithm and can be of any size as long as the bridges are within the scope of PBK and the precomputations do not take too much time.

For this implementation of the optimisation, the previous four selected bridges to test the effect of ignoring the interdependencies between bridges are used, which have been shown in Figure 4.12. Since the Rijckerbrug, Nieuwe-Wercksbrug and Oude Kinkerbrug are close together, the optimisation can be tested on taken into account those mutual effects. The Berlagebrug is located in another part of the city centre, so the closures of the other three bridges are expected to have less effect.
The time period of 2 years is implemented using time steps of one month. This means that the maintenance period in which all bridge maintenance should be scheduled consists of 24 time steps. To calculate the additional travel time, it is assumed that each month has an average of 30,5 days.

5.1.2. Maintenance duration and costs

The municipality has for every bridge a specific maintenance plan. Some bridges only need renovation, while some require full renewal. The characteristic of the bridge normally determine the duration and cost of the chosen maintenance plan. For this optimisation application it is presumed that the costs and duration options are the same for all bridges.

As discussed in the section about the municipality, there are many ways of closing a bridge for maintenance. Lanes can be closed for all traffic or for specific modes. It is also possible to make a bridge temporarily one-way. However, in section 4.4 is shown that the number of combinations gets out of hand with only having two options per bridge (open or closed), so adding more options would increase the computational complexity of this optimisation. Since the MoA is also putting more focus on 'short and heavy' maintenance, the decision has been made to only look at full closures for all modes within this research.

Based on conversations within the municipality, the average cost and duration of full renewal for bridges within the cite centre has been estimated on 8 million euro for a maintenance period of 6 months. The municipality strives for short maintenance to minimise the hindrance caused to the city with 'short and heavy' maintenance. If certain maintenance is expected to cause lots of hindrance to the surroundings and/or road users, the municipality can consider to let the construction work go on for more hours a day or for more days a week (Municipal Executive of the Municipality of Amsterdam, 2021a). A faster execution of construction work typically results in higher costs. This is largely due to need for labour outside the standard work hours, such as evenings, nights and weekends. These hours are compensated at higher rates compared to regular working hours.

The maintenance optimisation algorithm needs to be able to properly assess when it is worthwhile to invest more money in faster maintenance to reduce the hindrance caused to the road users. To model the impact of faster maintenance on cost and hindrance, three options are considered:

- Normal maintenance with standard working hours.
- Twice as fast by adding non-standard working hours.
- Three times as fast by adding more non-standard working hours.

Based on the current standard working hours and the current higher rates for evening, night and weekend hours, the cost for faster maintenance have been estimated. Table 5.1 shows the duration options with corresponding cost selected for this application.

Table 5.1: Maintenance options within the optimisation algorithm with corresponding duration and costs

Maintenance	Duration	Costs (in euro)
Normal	6 months	8 million
Twice as fast	3 months	8.8 million
Three times as fast	2 months	10.4 million

5.1.3. GA parameter settings

The GA will be implemented and solved using Python, specifically with the *pymoo* library version 0.5.0. Pymoo is a multi-objective optimisation framework in Python, which offers state of the art single- and multi-objective optimisation algorithms which can easily be customised (Blank & Deb, 2020). One of the optimisation algorithms provided is the genetic algorithm. This algorithm can be easily customised with different evolutionary operators and applies various categories of problems, including single objective problems.

The parameter values selected for the customisation of the GA used for the application of the test case are shown in Table 5.2. In section A.6 the code can be found that is used to execute the GA, containing the evaluation for every part of the objective function. For assessment of hindrance to road users, both the evaluation for the isolation method as the evaluation of the interdependency method can be found there as well.

Initially, a population size of 4 and a termination criterion of 100 generations are selected. This means that in each of the 100 generations, 4 solutions will be evaluated. While smaller population sizes allow for faster evaluation, larger populations provide greater diversity to avoid local optima. The GA is terminated after 100 generations, which corresponds to 400 solution evaluations. The number of generations needs to be large enough for the GA to converge effectively. The convergence behaviour of the GA will first be analysed to determine whether these values are sufficient. The first population is selected using the predefined get_sampling("int_random") method offered by pymoo. This selects random integers given the lower and upper bounds of the decision variables. For the crossover between

Table 5.2: Parameter settings for the optimisation using GA. *predefined methods offered by Pymoo

Parameter	Value
GA population size	4
GA maximum number of generations	100
Sampling of first population	get_sampling(”int_random”)*
Crossover	get_crossover(”int_sbx”)*
mutation	get_mutation(”int_pm”)*
eliminate_duplicates	True
Weight for hindrance (w)	10
Penalty size (p)	100,000

parent solutions a simulated binary crossover is used. The eliminate_duplicates check ensures that the mating produced offsprings are different from themselves and the existing population. The mutation of solutions over time is executed using polynomial mutation for discrete variables.

The problem to be solved with the GA contains 8 decision variables, namely a starting point and duration choice for each of the four bridges. The starting points can vary from 0 to 23 corresponding to the number of time steps, while the decision variables for the maintenance duration can vary from 0 to 2, each referring to one of the three duration possibilities. The solutions of the problem are evaluated using a single objective consisting of three parts, as has been shown in the Methodology chapter. The first part evaluates the traffic impact, which is based on total additional travel time in hours caused by the bridge closures. The second part evaluates the maintenance costs, expressed in euros. The last part acts as a constraint by heavy penalising maintenance projects that overrun the maintenance period of 24 months.

However, the maintenance costs and additional travel time in hours are not directly comparable because they are not in the same order of magnitude. Maintenance costs are typically much higher in absolute terms than the quantified impact of additional travel time, which would cause the optimisation to prioritise minimising costs over reducing travel time. To address this, a weight is applied to the traffic impact to bring both elements onto a comparable scale. This weight can be adjusted to reflect the priorities of policymakers. If reducing additional travel time is a higher priority than minimising costs, a higher weight can be applied to the traffic impact.

For this first application, it has been chosen to set the weight of the additional travel time in such a way that both travel time and costs are at least on an equal footing in the optimisation process. Therefore, the value has been based on a recent study to the current national value of travel times of the Netherlands. According to this study, the average value of travel time for car and cycling are €10.42 and €10.39 per hour, respectively (Kouwenhoven et al., 2023). By using a weight of 10, the influence in the optimisation process of the traffic impact will be aligned more closely with that of maintenance costs. The penalty for overrunning the maintenance period is set to 100,000, since the penalty has to be large enough to ensure no maintenance projects are exceeding the maintenance period.

5.2. Creating controls

For each of the selected bridges for maintenance, controls need to be created to be able to simulate the closures in UST. This first requires finding the corresponding road link numbers in UST for each bridge in the bridge set.

5.2.1. Finding the corresponding road links for each bridge

The municipality uses bridge codes to identify bridges, but these codes do not exist in UST. In the previous accessibility study of the Municipality of Amsterdam, bridge codes have been linked to their corresponding road links in the UST (Van der Sluijs et al., 2023). This included 687 bridges within the scope of PBK, all accessible for vehicles. The database has been made available by Carlijn van der Sluijs for this research. However, some bridges exist of multiple road links in the UST, which is also the case for the Berlagebrug, see Figure 5.1. Since only complete closures are used, closing one of the

Figure 5.1: Three road sections for the Berlagebrug in UST

Table 5.3: Bridge dataset for the selected bridges linking the bridge codes to the corresponding road link numbers and controls in UST

Bridge code	Road link nr.	Name	Control
BRU0063	22426	Nieuwe-Wercksbrug	337
BRU0167	21017	Rijckerbrug	335
BRU0169	218859	Oude Kinkerbrug	334
BRU0423	218884	Berlagebrug	336

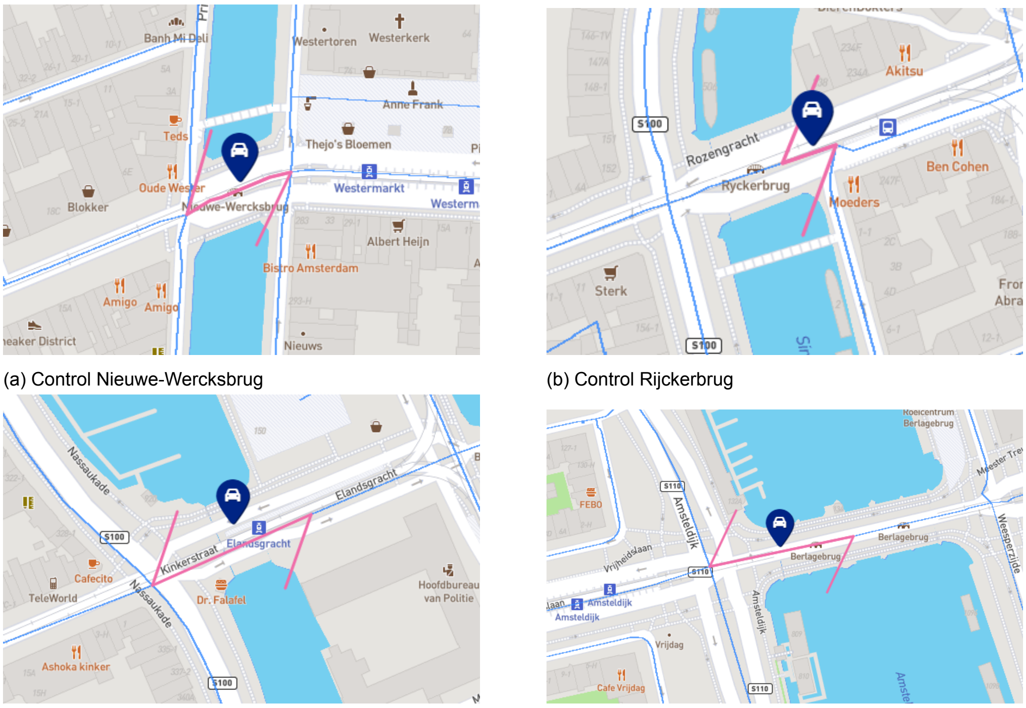
links will be enough to make it impossible for traffic to use the bridge. Therefore, a new bridge dataset has been created containing one UST road link only for every bridge code. This dataset has been expanded with bridge names, since it is sometimes easier to identify bridges based on their name. Of the 687 bridges within the scope of PBK, 347 bridges are documented by L.A.M. Reniers and P. Korrel and linked to the database of bridge codes and UST road links (Reniers & Korrel, n.d.). Given the selected bridge set in bridge codes, the corresponding UST road links and names are retrieved from the dataset.

5.2.2. Creating bridge specific controls for complete closure

To simulate the complete closures for all traffic for a specific bridge, the capacity and speed for both directions and all modes are set to zero for the corresponding UST road link. Instead of manually adding the controls in the interface of UST for each of the bridges, the controls are automatically generated and pushed to the simulation using the REST API. The code used to initialise a connection with the UST can be found in section A.1. The code used to create controls can be found in appendix section A.2. The created controls and corresponding road links can be seen in Figure 5.2. After the controls are created, the control numbers are stored together with the bridge codes, UST road links and names. That way, a dataset is created which contains all information needed of the selected bridge set in later steps of the optimisation. Table 5.3 shows this dataset for the selected bridges in this application of the optimisation.

5.3. Precomputations

For the optimisation using the isolation method, the additional travel times of all individual closures are needed. For using the interdependency method, also the additional travel times and interdependency values of all bridge pairs need to be computed beforehand. To compute the effect of all individual closures and all combination of closures on the total additional travel time, the UST needs to be simulated with certain controls activated. After retrieving the road data of these simulations, the additional travel time caused by the closures can be calculated by comparing the data from the simulation to a reference scenario. Therefore, an initial scenario is used based on a simulation run with no controls activated. The calculated additional travel times for all scenarios and the interdependency values will be used in the optimisation algorithm.



(a) Control Nieuwe-Werckbrug

(b) Control Rijckerbrug

(c) Control Oude Kinkerbrug

(d) Control Berlagebrug

Figure 5.2: Controls and corresponding road links of the 4 selected bridges. In dark blue the activated controls, in pink the selected road links. The half arrow tips on both sides show that the control operates on both directions. In Figure 5.2b and Figure 5.2d it can be seen that the bridge consists of multiple road links. By closing one of the road links, the bridges are still completely closed.

5.3.1. Retrieving road data

First the road data will be collected of all the simulation runs. For every run, the right controls are activated using the python function shown in subsection A.3.1. After a change in the activated control list, the UST automatically starts recalculating the output data for the traffic simulation. Using the rest API, the road data collections of the UST can be retrieved, containing the intensity and travel times for every road link in the system for both directions and all three modes (car, freight and bike). The road collection containing all road data is saved for each simulation run. In section A.3 the code used to simulate and retrieve the road data for the initial scenario, all individual closures and all combined closures can be found.

5.3.2. Calculating travel times

For the initial scenario and for all individual and combined closures, the travel times in the system are calculated using the code in section A.4.

If the total travel times of all different runs are known, the additional travel times can be calculated by comparing the travel times in the system to the initial situation. This results in an overview of the additional hours spent travelling in the system during the morning period simulated in the UST for the total system and for the three specific modes. For the individual closures the results of the additional travel time has already been shown in Table 4.3. The additional travel time in the system for the combined closures can be seen in Table 5.4.

5.3.3. Interdependencies computation

With the interdependency method, the simulations of two combined closures are used in the evaluation of additional travel time based on the interdependency between the bridges. The methodology has shown how these interdependency values are determined. To see the code that has been used to calculate these interdependencies for the given bridge set, see section A.5. For the results, see Table 4.4.

Table 5.4: Additional travel times for all combinations of two closures in hours for one morning peak simulation (NW = Nieuwe-Wercksbrug, R = Rijckerbrug, OK = Oude Kinkerbrug, B = Berlagebrug)

	Initial travel time in the system (h)	Additional travel time for each combined closure (h)					
		NW+R	NW+OK	NW+B	R+OK	R+B	OK+B
Total	228412.11	46.56	50.62	216.69	59.36	214.15	199.67
Car	115966.15	21.77	20.99	46.20	21.10	43.24	36.16
Freight	20893.00	5.71	8.35	6.53	8.94	5.09	5.69
Bike	91552.95	19.08	21.29	163.96	29.32	165.82	157.81

6 Results

This chapter will discuss the results of the application of the methodology on the test case. First, a look will be taken at the convergence of both optimisation methods. Then, the optimal solutions retrieved using the interdependence method will be compared to the isolation method for different seeds of the algorithm. For both methods, the estimated traffic impact of the optimal solutions will also be compared to the fully simulated traffic impact of those schedules, to asses the quality of the traffic impact estimations. Lastly, the objective values of the optimal solutions using the UST will be compared, to see which method leads to better maintenance schedules.

6.1. Convergence

In Figure 6.1 the convergence behaviour can be seen of the objective value for both the isolation and interdependency optimisation methods across different runs, using 20 different seeds. A seed determines the sequence of random numbers generated by the random number generator, which in turn affects various stochastic processes of the genetic algorithm, such as the initialisation of the population, the selection of individuals for crossover, and the mutation process. In order to assess the general performance of both optimisation methods and avoid the influence of any specific random sequence, it is important to perform multiple runs with different seeds.

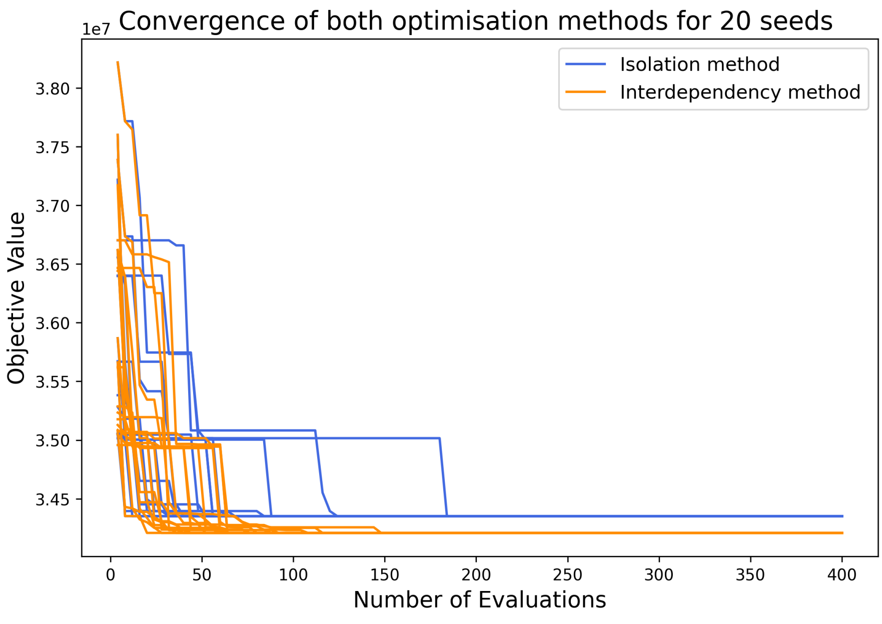


Figure 6.1: Convergence of the objective value for the two different estimation methods for 20 seeds

The genetic algorithms has been run for 100 generations with a population of 4, which means 400 evaluations have been done for every run. The figure shows that across all runs, both methods have reached a stable objective value after 200 evaluations, which suggests that running the genetic algorithm for 100 generations with this population size is sufficient to find an optimal solution. The isolation method converged to an objective value of 34352312.17 and the interdependency method to an objective value of 34209165.61. However, the GA is a heuristic which does not guarantee that the true global optimal solution over all possible solutions will be found. Even though the optimisation seems to be converged, the algorithm could be stuck in a local optimum. For the optimisation method a genetic algorithm was chosen, since the time it takes to evaluate all possible solutions increases exponentially with the number of bridges. With 24 possible starting points, 3 maintenance options and 4 bridges the total number of possible solutions is $(24 \cdot 3)^4$, which is 26,873,856. The evaluation of the 20 solutions for the different seeds of the interdependency method takes 0.13 seconds[1]. While full enumeration of all solutions with 4 bridges can thus be done in 1,5 days, evaluating all solutions for 5 bridges would take 3 months and for 6 bridges even 22 years. This emphasises the need for a meta-heuristic search-based optimisation like the genetic algorithm. The small case study of four bridges makes it possible to check whether the GA has found the global optimal solution. All 26.8 million possible solutions have been evaluated with the interdependency method. Figure 6.2 shows the distribution of the objective values for all possible solutions determined by the interdependency method. This shows that the lowest objective value found by the Genetic Algorithm using the interdependency method is indeed the global optimum of 34209165.61.

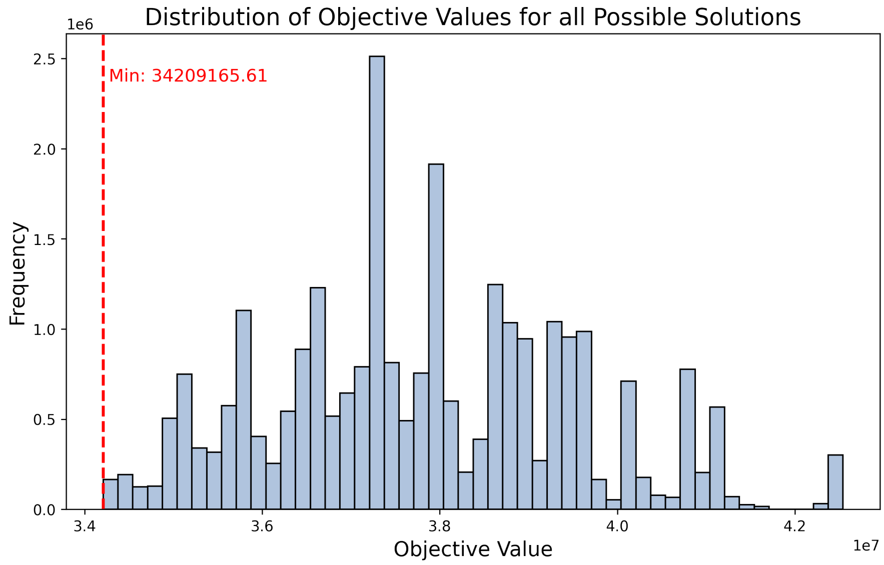


Figure 6.2: Histogram for the objective values using the interdependency method of all possible solutions

Looking at the differences between the two optimisation methods in convergence, it can be seen that the interdependency method reaches a lower objective value then the isolation method. The next section will take a closer look at the optimal results of both optimisation methods.

6.2. Analysing the optimal solutions for the isolation and interdependency method

The convergence graph shows that both solutions reach their optimal solution within the 100 generations and that both methods converge to a consistent objective value for all 20 seeds of the optimisation method. Table 6.1 shows the optimal objective value and separate scores of the three individual components within the objective value for both methods. For the full list of solutions and scores for every seeds, see Appendix B. The objective value shown in the table is calculated with a weight of 10 for the

[1]For these evaluations, a computer with the following specifications was used: Intel Core i5-6200U CPU @ 2.30GHz 2.40 GHz and installed memory of 8 GB (RAM)

Table 6.1: Results of all optimisation runs

	Isolation method	**Interdependency method**
Additional travel time (h)	155231.21	140916.56
Maintenance cost (€)	32800000	32800000
Penalty for overrun	0	0
Total objective value	34352312.17	34209165.61

additional travel time in the schedule.

Looking at the results in the table, a few things can be noted. First, in none of the schedules generated by both optimisation methods the maintenance plans overrun the maintenance period of 24 months, so no penalty was given. This means that the penalty size that has been selected is large enough to avoid overrun in the maintenance schedule. Second, the maintenance cost for both methods are the same, since in all optimal results of both methods for every bridge the same duration is chosen. And third, the additional travel time score of the interdependency method is lower than the additional travel time score of the isolation method. This can be explained when looking at the visualisation of a few of the optimal solutions of both methods. In this section a few specific schedules will be shown. A visualisation of all 20 seeds for both methods can be found in Appendix B.

Figure 6.3 shows four different optimal schedules of the isolation method. With this method the effect of bridge closures are assessed as if the bridges are isolated from each other. For the evaluation of additional travel time, it therefore does not matter how the maintenance plans are coordinated. This can also be seen in the 4 schedules shown. The schedule for a seed of 2 shows that, with this method, all bridges can be in maintenance at the same time. While for a seed of 6, the maintenance plans are almost all done separately. Both solutions have the same additional travel time according to the estimation using the isolation method. This randomness in distribution of maintenance plans over time can coincidentally lead to good or bad combinations of bridge maintenance plans. The schedule for a seed of 5 shows that the maintenance plans of the Oude Kinkerbrug and the Rijckerbrug almost completely overlap. In Table 4.5 has been shown that the combination of the Oude Kinkerbrug and Rijckerbrug is the worst combination that can be made based on their closure effects on each other. While this solution may be optimal using the isolation method, the actual traffic effects will be larger. On the other hand, the schedule for a seed of 18 shows that it is also possible for the optimisation to make good combination of maintenance. Even though it is only a small overlap, the combination of the Rijckerbrug and Nieuwe-Wercksbrug will results in lower additional travel times according to their closure effects on each other in Table 4.5.

When the optimal results of the isolation method are compared to those of the interdependency method in Figure 6.4, it can be seen clearly that with the interdependency method, the maintenance plans of the Rijckerbrug and Nieuwe-Wercksbrug are always aligned. Those two bridges have the highest absolute interdependency value and are thus always evaluated with their combined simulation, which result in lower additional travel times. The schedules for seed 9 and 17 however show that those two bridges can also be paired with a third bridge. In case for seed of 9 even creating the unwanted combination of the Oude Kinkerbrug and the Rijckerbrug. This can be explained by the estimation method of the interdependency optimisation. When a group simultaneous closures is evaluated, pairs of two are formed based on their interdependency value. For every evaluated pair in the group of closures, their combined closure simulation is used. However, the simulations of the different pairs within the group of simultaneous closures are added together as if the pairs are isolated from each other. In case of an uneven group of closures, this is also the case for the last bridge evaluated. This explains why the Berlagebrug or the Oude Kinkerbrug can be simultaneous with the combination of the Rijckerbrug and the Nieuwe-Wercksbrug. For this bridge set no combinations of 4 are made in all of the optimal solutions using the interdependency method. When all the bridges would be evaluated together, the Berlagebrug and Oude Kinkerbrug would form a pair after the Rijckerbrug and Nieuwe-Wercksbrug. Since the Berlagebrug and Oude Kinkerbrug are not completely isolated according to Table 4.5, there combination would be avoided.

In all optimal solutions of both methods, the maintenance of the Berlagebrug has an execution of

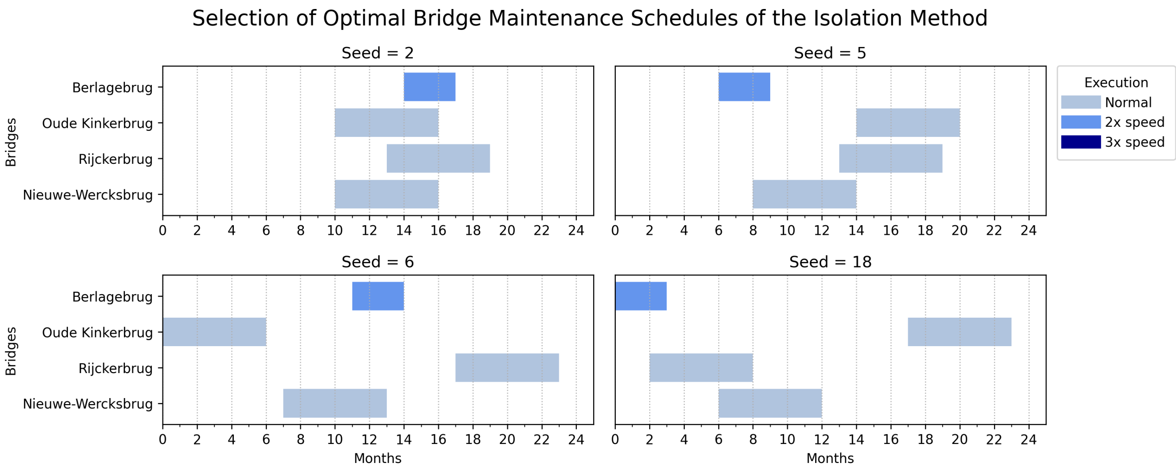


Figure 6.3: Four alternative optimal bridge maintenance schedules of the isolation method for selected seeds

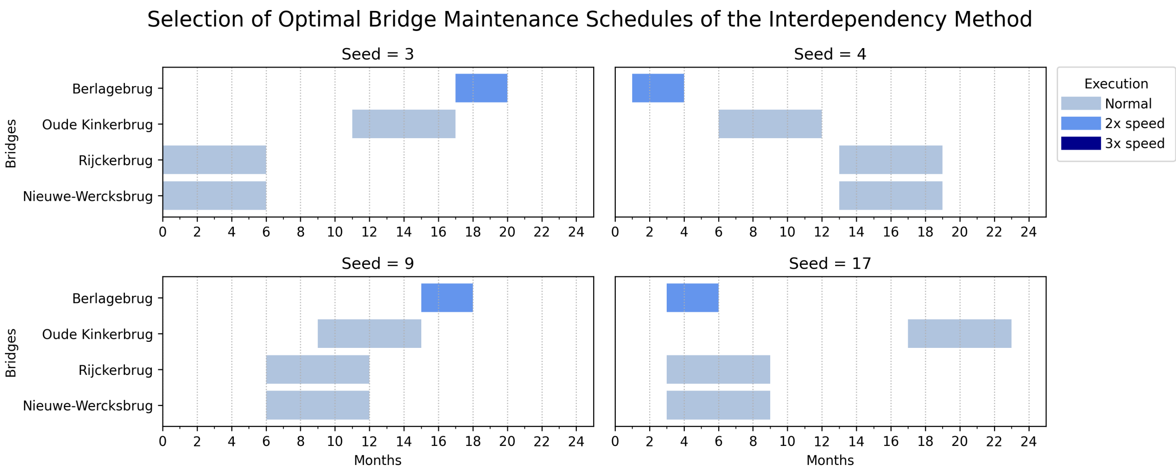


Figure 6.4: Four alternative optimal bridge maintenance schedules of the interdependency method for selected seeds

two times faster than normal, while the other bridges will be in maintenance for the normal execution duration of 2 months. The faster the maintenance, the darker the maintenance plan colour of the bridge in the schedule visualisations. This can be explained by the fact that the additional travel time after the individual closure of the Berlagebrug is already 6 times larger than that of the individual closures of the Nieuwe-Wercksbrug and the Rijckerbrug, and even 12 times larger than the closure of the Oude Kinkerbrug . With a weight of 10 for the traffic effect, increasing the execution of the maintenance of the Berlagebrug will result in a better overall solution using these optimisation methods. The sensitivity of this weight has been investigated by analysing the optimal schedules of both optimisation methods for a consistent seed using different weights. The transition weights found are shown in Table 6.2 and the corresponding schedules can be found in Appendix C. Given the additional travel times of the individual closures in Table 4.3, it can be expected that the transitions to faster maintenance for bridges with higher additional travel time happen at a lower weight. It is however interesting to see that the transitions of the Rijckerbrug and Nieuwe-Wercksbrug for the interdependency method happen at a higher weight and simultaneous compared to the isolation method. Taking interdependencies between bridges into account can thus avoid using unnecessary expensive and intensive maintenance execution by effectively combining the closures.

Only based on the objective values of both optimisation methods, not much can be said about which method is better, since both methods have a different evaluation of the effect of bridge closures on traffic. Both methods estimate the effect of the combination of closures. To assess the level of quality of these estimations, the optimal schedules generated by both methods need to be fully simulated in

Table 6.2: Transition weights for bridge maintenance duration for both optimisation methods (Isolation = Iso., Interdependency = Int.)

Transition weights for bridge maintenance duration				
	Normal >2x speed		2x speed >3x speed	
	Iso.	Int.	Iso.	Int.
Berlagebrug	10	10	57	57
Oude Kinkerbrug	115	115	685	685
Rijckerbrug	59	76	353	451
Nieuwe-Wercksbrug	54	76	324	451

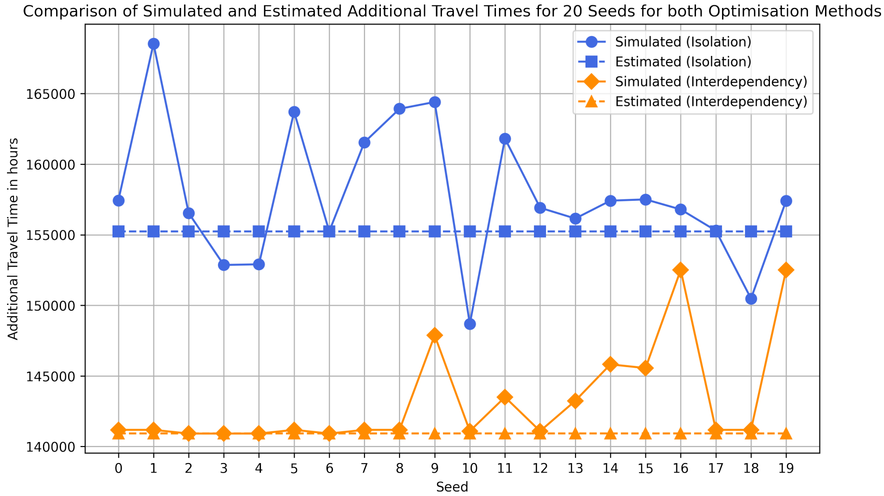


Figure 6.5: Comparison of the simulated and estimated additional travel times for 20 seeds for both optimisation methods

UST.

6.3. The accuracy of the additional travel time estimations

In this section of the results the traffic effect estimations of both optimisation methods will be compared to the fully simulated traffic effect of the optimal schedules. For both methods, all 20 optimal schedules of the 20 different seeds are put back in the UST. Instead of estimating the effect of combined closures, all combinations that occur in the optimal schedules are now simulated using the UST. Figure 6.5 shows the UST simulated additional travel time of the optimal maintenance schedules for both methods for 20 seeds compared to the estimations of both methods.

The estimated and simulated additional travel time are shown in blue for the isolation method and shown in orange for the interdependency method. The exact differences can be found in Appendix B. The blue lines for the isolation method show that the optimal schedules can result in more and less additional travel time than estimated. This has already been shown in the visualisation of a few optimal schedules in Figure 6.3, where with a seed of 5 a bad and with a seed of 18 a good combination of closures is made. As can be seen in Figure 6.5, the optimisation is indeed underestimating the additional travel time for the optimal schedule with a seed of 5, and overestimating the additional travel time for a seed of 18. The estimation of the additional travel time of the optimal schedule for a seed of 6 comes closest to the simulated additional travel time for the same schedule. Looking at the visualisation of that schedule in Figure 6.3, it can be concluded that the Berlagebrug can indeed be evaluated as isolated from the Nieuwe-Wercksbrug, since the simulated additional travel time does not differ that much from the isolated estimation. This is expected as the combination of the Berlagebrug and the Nieuwe-Wercksbrug has the lowest absolute interdependency value in Table 4.5.

While the simulated additional travel time for the isolation methods fluctuates around the estimated

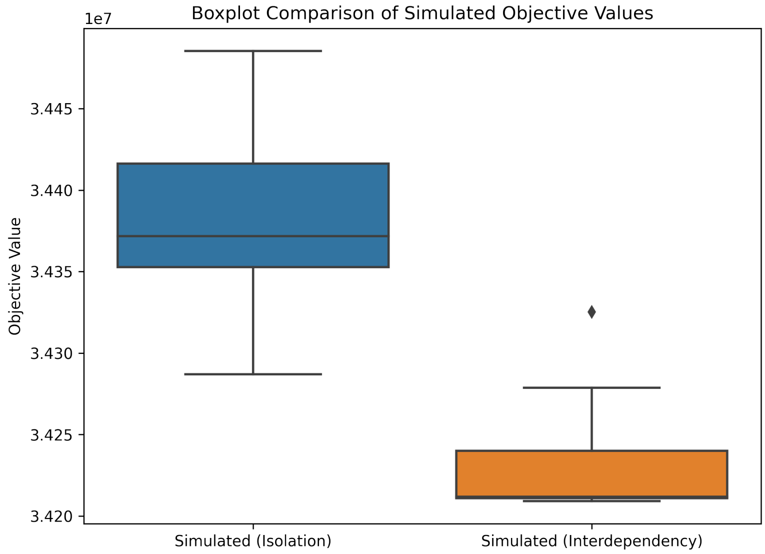


Figure 6.6: Boxplot of the simulated objective values of the optimal solutions of both methods

additional travel time, the estimation of additional travel time for the interdependency method is always the same or lower than that of the simulated additional travel time. In optimal schedules where there are only combinations of 2 closures, the estimation always uses the simulation run of the combined closure. The total additional travel time estimated by the optimisation in those cases is thus automatically the same as the simulated additional travel time. Only when more than 2 bridges closures are combined in the maintenance schedule, the estimation deviates from the full simulation. In all cases where the Berlagebrug is combined with the Rijckerbrug and Nieuwe-Wercksbrug pair, such as for seed 17, the simulated additional travel times are only slightly higher than estimated. This again shows that it can be assumed that the Berlagebrug is isolated from the other bridges and can be assessed that way. In the other cases where there is a significant difference between the estimation and the simulation, the maintenance plans of the Oude Kinkerbrug somewhat overlaps with the combined closures of the Rijckerbrug and Nieuwe-Wercksbrug. The greater the overlap, the higher the simulated additional travel times, and the greater the underestimation of the optimisation. Both the Berlagebrug and the Oude Kinkerbrug have a positive interdependency value with the bridge pair Rijckerbrug and Nieuwe-Wercksbrug, meaning their combination would lead to more additional travel times. If there would be another bridge in the bridge set that would have a negative interdependency value with the bridge pair, it could be that the interdependency method overestimated the additional travel time when assessing the three bridges together.

Looking at the differences between the two methods, it can be seen that the optimal solutions generated with the interdependency method are almost always better in terms of additional travel time compared to the optimal solutions of the isolation method. However, the optimal schedules of both methods are different and to determine which method leads to better schedules, the full objective function should be taken into account.

6.4. Comparing the estimation methods results using the UST

All optimal solutions generated by the isolation and interdependency method have been evaluated with the objective function using the UST simulation runs for the determination of the additional travel time caused by the maintenance schedule. This means that the additional travel time is given a weight of 10 and the maintenance cost and penalty for maintenance period overrun are also taken into account. Figure 6.6 shows the boxplot of the simulated objective values of the optimal solutions of both methods.

The boxplot of the isolation method shows a wider distribution of objective values than that of the interdependency method, showing a higher overall variability. The median in the middle of the isolation box is near the centre of the box and the first and fourth quartile (the whiskers) are almost the same length

, indicating a relatively symmetric distribution. This can be explained by the randomness in the spread of the different maintenance plans over the maintenance period. The boxplot of the interdependency method shows a more compact distribution of objective values, with the median almost as low as the minimum value. This suggest a more consistent performance of the optimisation method. Except for one outlier, all schedules generated by the interdependency method perform better than those of the isolation method.

The fact that the interdependency method scores better on the objective is in line with the conclusion that the interdependency method scored better on the simulated travel times. Since both methods have the same maintenance cost for all optimal schedules and no penalties for period overrun, the differences in objective function are entirely due to the differences in additional travel time caused by the optimal schedules.

Based on these boxplots, it can be concluded that the optimisation using the interdependency method consistently produces lower objective values and thus is the better optimisation method for creating good maintenance schedules based on the objective function in this research. The isolation method leads to higher and more variable results, making it less effective for the planning of bridge maintenance in order to reduce hindrance to road users in terms of additional travel time.

7 Conclusions

This research, focused on bridge maintenance planning optimisation considering hindrance to road users, has proposed an improved optimisation method using simulations of the UST to reduce additional travel time caused by the planning of upcoming bridge maintenance. Additionally to minimising additional travel time caused by the bridge maintenance schedule, the research aim was to improve the existing assessment of the effect of combined closures used in TNO's research about the quay wall maintenance optimisation. To conclude this research, an answer will be provided to the following main research question:

How can additional travel time caused by bridge maintenance be minimised combining traffic simulation and optimisation?

To answer the research question, the supporting sub-questions on what is known about this topic in literature and practice have been answered at the end of chapter 2, 3 and 4. To answer sub-question 1 and 2, relevant literature and current practice were explored. The literature review in chapter 2 showed an gap in optimising short-term maintenance with the goal of minimising hindrance to road users. While genetic algorithms proved to be an effective optimisation method for optimising long-term maintenance of road structures, combining the genetic algorithm with traffic simulation had not been much researched. The often long computation time of these simulations made them impractical for the use in optimisation. However, the needs of the municipality of Amsterdam discussed in chapter 3 called for an optimisation of upcoming bridge maintenance where the effect of multiple simultaneous closures was assessed accurately. The fast simulations of the Urban Strategy Tool by TNO, which only take approximately 3 minutes, provided new possibilities for combining optimisation with simulation. However, its use in maintenance optimisation still neglected network effects for simultaneous closures. Despite the fast simulations, using exact simulations for all combinations of closures in maintenance optimisation would lead to high computation times due to the exponential increase of possible combinations with the number of bridges. In a study optimising long-term quay wall maintenance, the effect of multiple simultaneous closures was therefore estimated by summing the effects of individual closures. It was shown in chapter 4 that this method, which assumes that different closures are isolated from one another, can lead to wrong conclusions about the best combination of simultaneous closures to minimise additional travel time.

This study proposed an improved estimation method for assessing the effect of combined closures for bridge maintenance optimisation, while keeping the computational time low for easy use in short-term maintenance planning. This new estimation method uses the interdependencies between bridges to come to a more realistic estimate of the effect of simultaneous closures. The interdependencies between bridge pairs are determined by looking at the effect of individual bridge closures on the traffic at the other bridges. Based on the interdependencies between bridge pairs, the simulations of the UST for two combined closures are used. Both the isolation estimation method and the new proposed interdependency estimation method are used in the designed genetic algorithm for optimising bridge maintenance. The genetic algorithm minimises the estimated additional travel time caused by the closures of bridges by selecting the starting point and duration of each bridge maintenance project, while also minimising the costs for the maintenance execution. This proposed method answered sub-question 3 on how to optimise bridge maintenance planning while minimising both additional travel times and maintenance costs and sub-question 4 on how the additional travel times caused by simultaneous closures can be taken into account using simulations.

The optimisation algorithm was run for a test case of the city of Amsterdam, consisting of four bridges in the city centre. With both methods the designed genetic algorithm converged to a stable objective value for multiple different runs within 200 evaluations rounds. While the isolation method only needs the simulation of all possible closures individually, the interdependency method also needs to precompute all combinations of 2 closures. Even though the precomputational times are slightly higher, the runtime of the algorithm does not increase using the interdependency method. This improves the accuracy of the estimation of additional time while keeping the optimisation useful in bridge maintenance planning.

The optimisation using the isolation method produced solutions of varying objective values. Sometimes this resulted in relatively good solutions, but sometimes closures were combined in a way that would increase the additional travel time more than necessary. The optimisation using the interdependency method produced consistently better solutions, due to a more accurate estimation of the additional travel times caused by combined closures. The interdependency method made sure that at least pairs of bridges with high interdependencies were evaluated using a simulation run. This prevented bad combinations of closure more often and took advantage of the good combinations to produce better bridge maintenance schedules. Not all bad combination could be avoided, since the interdependencies were only calculated for bridge pairs. When assessing more than two simultaneous bridge closures, it could be that high interdependencies between two bridges of different pairs within a simultaneous closed bridgeset are neglected due to one bridge having a higher interdependency with another bridge. Despite this limitation, the proposed interdependency estimation method improved the accuracy of the evaluation of additional travel time in bridge maintenance optimisation compared to the isolation method.

Based on the results, it can be stated that the proposed method provides an answer to the main research question:” *How can additional travel time caused by bridge maintenance be minimised combining traffic simulation and optimisation?*”. By using the interdependency method for estimating the additional travel time caused by bridge maintenance, a way has been found to incorporate traffic simulations in optimisation to take network effects into account, while avoiding the usual computational difficulties of combining these two. The interdependency method improved the accuracy of the evaluation of additional travel time due to bridge maintenance compared to previous estimation methods. The ability of this method to combine closures in a way to reduce additional travel time and consider faster maintenance execution in case of severe hindrance helps minimising the additional travel time caused by bridge maintenance.

The focus on optimising short-term maintenance in order to reduce hindrance to road users by this new method of combining fast traffic simulations with a genetic algorithm is not only a scientific contribution, it can also be very helpful in the planning of bridge maintenance in Amsterdam, being a fast way of showing the effects of different priorities on bridge maintenance planning.

8
Recommendations

Within this section recommendations will be made for further research, based on the limitations of this research and other interesting directions for research found during the conducting of this study. Additionally, special attention is given on providing recommendations for the Municipality of Amsterdam regarding optimisation of maintenance, interesting opportunities using the UST and needed improvements for better decision making.

8.1. Model improvements

The proposed optimisation for short-term bridge maintenance planning improved the assessment of the effect of combined closures on additional travel time, providing good solutions for the municipality of Amsterdam to reduce hindrance to road users. Still there are some limitations to the current method. Therefore, some model improvements are suggested to address these limitations and to increase the extent to which the optimisation can reduce hindrance to road users.

8.1.1. Additional travel time determination

First the limitations and possible model improvements for assessing the hindrance to road users will be presented. In this study, the hindrance to road users is measured in additional travel time together for the modalities car, freight and bicycle by using simulations of the morning peak in UST.

- **Expanding the included modalities**
  The addition of bicycles has already been an improvement compared to the current literature in taking into account hindrance to road users in maintenance planning. This is certainly important for a city like Amsterdam, where cyclists make up a large part of the traffic in the city. To improve the assessment of hindrance to road users, the number of modalities taken into account could be expended. For this research, public transport and pedestrians have been excluded. One of the reasons for this was the dependency on the UST model made available for this research, which did not include these modalities. An additional public transport model exists within the UST environment and could be included in the optimisation to include the effect on bus and tram users. However, due fixed timetables, stops and routes for public transport, closures have to be reported much earlier and not all closures will be possible. Before public transport can be added to the optimisation, it must be investigated how this can be taken into account.
- **Including mode specific additional travel time**
  The objective function minimises the total amount of additional travel time. By splitting the additional travel time into mode specific additional travel times, different weight to car, freight and bicycles can be given. This makes the optimisation more useful for taking into account different policy priorities, for example when the policy is more focused on reducing hindrance for cyclists. The code for the proposed model has already been prepared for this split in travel time, by calculating the mode specific additional travel times in addition to the total additional travel times when evaluating solutions. The next step would be to explore the effect of mode specific additional

travel times with variable weights on the generated solutions.

- **Simulating for other than the morning peak**
  The optimisation is based on traffic simulations of a 2 hour morning peak. For this research it is assumed that these effects can be extrapolated to calculate the effect of the entire duration of the traffic situation. However, the behaviour of traffic during the morning peak can differ considerably from traffic during the evening peak or outside peaks in the middle of the day, at night or during the weekend. Typically, during the morning and evening peaks, traffic travels in opposite directions, as people commute between residential and working areas. Outside these peaks, different traffic patterns can apply. Only simulating the morning peak can therefore lead to wrong conclusions about the optimal bridge maintenance schedule. Especially the existence of one-way bridges or roads could lead to different conclusions between the morning and evening peak. To overcome this limitation, the effects of different simulation periods should be included in optimisation.

- **Including the additional effect of other traffic simulation**
  The UST is a static macroscopic traffic model. In the macroscopic model the traffic volumes are determined by the speed and the density and the traffic flows are assumed to be in equilibrium. It predicts long term average steady state of the road network and assumes everyone has found their fastest route. Time dependencies and the effects of spill-back due to congestion are not considered. Dynamic models do take these effects into account and therefore could improve the assessment of hindrance to road users. Other improvements could be made for the initial phase of bridge closures, when road users are often unaware of the new situation. Because the road users have not yet adjusted their routes to the new situation, their detour routes can be unnecessarily longer. This effect cannot be captured using the UST simulations, so the additional travel time determined by the optimisation in this study is probably underestimating the effect of the initial phase of closures. Additional simulations using other models like dynamic traffic models or agent-based simulation models can improve the assessment of the extra additional travel time caused by the initial phase of unexpected bridge closures. However, these models often require significantly more computational time to run, which may limit their integration into the optimisation. Therefore, further investigation is needed to explore the feasibility of combining these models within the bridge maintenance optimisation.

8.1.2. Expanding the maintenance options

The short-term bridge maintenance planning optimisation produces a maintenance schedule by selecting the starting point and the duration of the complete closure of every bridge in the provided bridge set. For every bridge only complete closures are considered and the possible duration are the same for every bridge in the bridge set.

- **Different types of closures**
  Based on the municipalities policy of having 'short and heavy' maintenance, the decision has been made to only include complete closures of bridges. However, there are other options possible for closing roads on bridges to make maintenance possible. Mode specific road closures, one-way traffic or reduced passage are all possibilities of reducing traffic on bridges to create space for maintenance work. The choice for different types of closures impacts the additional travel time caused to road users by affecting the passage of traffic and the duration of the maintenance project. Expanding the possible types of closures could help in reducing the overall additional travel time. With the UST it is possible to mimic these types of closures, but the proposed estimation method is built on the interdependencies between complete closures. Calculating the interdependencies between bridges for multiple types of closures, increases the possible combinations and therefor the computational time. The effectiveness and accuracy of the current estimation method after expanding the closure options, or the possible adaptions, should be further investigated.

- **Bridge specific maintenance options**
  The genetic algorithm selects a starting point and maintenance option for each bridge. When applying the proposed method, the maintenance options were the same for all 4 bridges, namely complete closure for a duration of 6, 3 or 2 months. But in reality, each bridge requires a different type of renovation and the differences in the bridge structure provide different levels of difficulty. It

would therefore be better to also include bridge-specific maintenance options in the optimisation. The design of the chromosomes of the genetic algorithm can be unchanged, but the way the chromosomes are translated to schedules in the code should be adapted.

8.1.3. The evaluation of simultaneous closures

This study has proposed an improved method of assessing the effect of simultaneous closure in bridge maintenance optimisation on additional travel time for road users. The method uses the interdependencies between bridge pairs to improve the estimation of the simultaneous closure of multiple bridges.

- **Determining the interdependency**
  The interdependencies are determined by the effect of the complete closure of a bridge on the traffic on other bridges, measured in travel time. For more understanding of the affected OD-pairs, the interdependencies between bridges could be expressed in terms of overlap in OD-pairs using the bridges. In traffic simulations like OmniTrans, information about the volumes of the OD-pairs using each road link is stored. This feature is not available in UST, which makes it more difficult to get insights in the affected OD-pairs. To get this insight, the optimisation could be tested with OmniTrans or the feature should be added in UST.
- **Improving the avoidance of wrong combinations of closures**
  The application of the optimisation showed that not all bad combination could be avoided using the interdependency estimation method. When assessing more than two simultaneous bridge closures, it could be that high interdependencies between two bridges of the simultaneous closure are neglected when one of the bridges has a higher interdependency with another bridge of the simultaneous closure. Using simulations of more than 2 combined closures, increases the number of combinations and thus the precomputation time. To avoid wrong combinations in simultaneous closures of more than 2 without increasing the number of precomputations, the assessment should be improved. When assessing a set of combined closures, for example a penalty could be given for when a third bridge is combined with bridge pair while showing positive interdependencies with both bridges of the bridge pair. This needs further research.
- **Exploring the accuracy of the estimation further**
  The proposed evaluation method has been tested on the optimisation of bridge maintenance for a case with 4 bridges in the city centre of Amsterdam. Since the methods form pairs of two bridges based on interdependencies, the full potential of the estimation method could be further explored when tested on a larger dataset. With four bridges, the maximum number of pairs evaluated for a simultaneous closure is two pairs. With more bridges and thus larger groups of simultaneous closures, the error of the interdependency estimation method for additional travel time compared to the true simulation value is expected to increase. However, the error of the isolation method compared to the true simulation value is expected to increase even further. A larger dataset of bridge closures is needed to showcase the accuracy of the estimation method when multiple pairs are formed due to the simultaneous closure of more than 4 bridges.

8.2. Future research on reducing the hindrance caused by maintenance

In the previous section improvements for the proposed model are suggested to increase the potential of the short term bridge maintenance optimisation in order to reduce additional travel time caused to road users. The research objective emerged from the need to consider hindrance to road users when planning maintenance, as they are expected to face an increase in experienced hindrance in the coming years due to the rise in maintenance projects to catch up on the national backlog. This study proposed a method to reduce the total additional travel time caused to road users, but this is not the only way of capturing the experienced hindrance. Therefore, this section suggest several future research options to reduce hindrance caused by maintenance.

A study for the Municipality of Dordrecht researched the perception of road closures among road users, residents and businesses for different road closures in their city. In addition to increased travel times, factors such as the detours themselves, increased traffic and unsafe traffic situations were often mentioned as types of hindrance (Damen et al., 2020). Clear communication before and during the closures, as well as understanding the importance of the closures, are also mentioned as important factors influencing the experience of hindrance. However, considering all these different hindrance factors for various groups can complicate optimisation, as the interests of these groups may conflict. For example, road users may prefer continuous, around-the-clock construction work to minimise the duration of closures, while residents might experience increased noise hindrance from work during the night. Research into the use of multi-objective optimisation could help balance these competing interests. By expanding the focus beyond just additional travel time to include the impact of improved communication, better detours, and enhanced safety in temporary traffic situations, it may be possible to further reduce the overall hindrance experienced during maintenance projects.

The experience of hindrance can also be aggravated by its repetition. When people are faced with multiple road closures in a short time, their perception of hindrance can intensify. The Municipality of Amsterdam is already trying to take this into account by aiming for 10 years of '*graafluwte*' (dig-free period) after long-term road closures (Municipal Executive of the Municipality of Amsterdam, 2021b). In this way, residents and business owners are assured a maintenance free period. For road users however, this '*graafluwte*' does not assure that they will not experience hindrance again in a very short time, since they can experience hindrance of every part of their route. To reduce hindrance experienced from repetition, it would be interesting to optimise the bridge maintenance planning while considering the additional hindrance experienced form repeated maintenance on routes. This could be done by aiming for maintenance free periods on frequently used routes after major closures, or by preventing having multiple detours at the same time on often used routes. The extent to which these types of repetition of hindrance aggravates the perception of hindrance is also an interesting direction for further research.

Given the anticipated increase in hindrance due to the large number of upcoming maintenance projects, it is crucial to consider hindrance when optimising planning. This study has taken a first step by focusing on reducing the total additional travel time caused by bridge maintenance. However, by only looking at total travel time, it remains unclear who experiences the hindrance and to what extent. It could be that many people experience only minor delays, while a small group may suffer from significant travel disruptions. The next step, therefore, is to investigate who is affected by the hindrance and to what degree. Distributing the impact of hindrance more evenly across different road users, modes, areas, or other categories could enhance equity. In the UST environment, a distinction is already made between different areas in Amsterdam. In addition to using this subdivision to analyse hindrance per area, the generated OD matrices could be studied to see which routes are most affected by maintenance.

8.3. Recommendations for the Municipality of Amsterdam

This study was commissioned by the Municipality of Amsterdam. During the execution of this research, there was a lot of interaction with the innovation team of PBK and other relevant departments within the programme. Based on the findings of this research and acquired knowledge during the work within the municipality, recommendations will be given on the potential use of UST and the use of optimisation for decision making for the city of Amsterdam.

8.3.1. The potential of the UST

The fast simulations of the UST offer many new opportunities for the Municipality of Amsterdam, which go beyond their use in optimisation. With the current traffic simulations of the VMA, it is not possible to quickly assess the effects of changes in the city's network. Because of the speed of the simulations, experts who previously had to rely solely on their experience can now perform quick checks using the UST. For example, it can assist in deciding whether to install a temporary bicycle and pedestrian bridge during a bridge closure. By simulating the effects of the closure on additional travel times for cyclists and pedestrians, more informed decision can be made on whether the cost of installing the temporary bridge is justified.

The way TNO has built the UST makes the simulation highly adaptable for integration with other data and models. This adaptability allows for the optimisation of bridge maintenance to be linked with real-time monitoring data. This extension of the model has already been identified as a future research area in the quay wall maintenance study conducted by TNO and the Municipality of Amsterdam. Together with the proposed optimisation from this study, alarming changes in the condition of bridges could be quickly incorporated into the short-term planning of maintenance. Additionally, the UST can be connected to other models. While this research has focused on the impact on road traffic, bridge maintenance also disrupts water traffic. TNO and the Municipality are currently already working on a water traffic model, which could, in the future, be integrated into the optimisation of bridge maintenance.

Moreover, the UST's interface makes the simulation very user-friendly. Demonstrations of UST usage to the municipality have learned that clear visualisations provided by the interface led to better insights into the effects of network changes. This visualisation aids discussions about various options and assists in decision-making. Additionally, users can easily make adjustments within the UST and study the effects of these changes within a few minutes.

8.3.2. The use of optimisation in decision-making

Fast optimisation, such as the one developed in this study, is very useful for planning maintenance activities. It offers a quick way to test numerous solutions, leading to better maintenance planning. This approach also allows for the testing of different perspectives. Since the city's coalition changes every four years, the municipality's vision may also shift. Fast optimisations with adaptable objective functions can demonstrate the effects of varying policy positions. For example, if cycling becomes a priority, the focus should shift towards reducing hindrance to cyclists. Therefore, adding mode-specific weights to the objective function is a wise extension of the proposed optimisation method to assist with changing policies. Hence, it is also important to incorporate public transport and pedestrian in the optimisation model, in order to also include the effects on these modes. The optimisation of maintenance activities can be broadened by expanding it to other types of road closures. There are many other causes of long-term road closures that could be included in the optimisation process, such as replacing sewer or water pipes underground. This research on bridges clearly demonstrated the potential of the method, but the same methodology can be applied to all types of maintenance projects. These extensions of the optimisation process make it even more attractive for use in planning activities to minimise disruption to the city. However, it is important that the optimisation remains fast by efficiently incorporating the quick simulations of UST, as proposed in this study. Close collaboration with TNO remains crucial for this, given the reliance on their fast computing capabilities and their knowledge about the possibilities with the UST.

However, it is also important to note that not everything can be captured in optimisation. During this research, knowledge-sharing sessions have been attended between various municipal departments, PBK, and TNO on optimising maintenance activities. These sessions revealed that planning maintenance involves much more than just considering traffic and the condition of bridges. Other factors, such as utility companies' maintenance schedules, tree preservation, and additional constraints like ensuring emergency services' accessibility, must also be considered. Since some of these interests may conflict, it is impossible to incorporate all these different aspects into a single optimisation model to create one perfect maintenance schedule. However, optimisation provides better insight into the effects of various trade-offs. Therefore, a fast optimisation approach is highly suitable for visualising the effects of competing interests. The proposed optimisation model can be best used as a building block for a more comprehensive optimisation approach, where this proposed model handles the calculation of additional travel time for road users. Other factors that determine the level of hindrance for road users, as well as important constraints and requirements, should be considered separately. While optimisation will play a more significant role in decision-making, the final decisions will still need to be made after carefully weighing the various interests.

Bibliography

Abdelkader, E. M., Moselhi, O., Marzouk, M., & Zayed, T. (2022). An exponential chaotic differential evolution algorithm for optimizing bridge maintenance plans. *Automation in Construction*, *134*. https://doi.org/10.1016/j.autcon.2021.104107

Alsharqawi, M., Saleh, ; Dabous, A., Zayed, T., Asce, F., & Hamdan, S. (2021). Budget Optimization of Concrete Bridge Decks under Performance-Based Contract Settings. *Journal of Construction Engineering and Management*, *147*(6). https://doi.org/10.1061/(ASCE)CO.1943-7862.0002043

Blank, J., & Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*, *8*, 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567

Bleijenberg, A. (2021, April). *Instandhouding civiele infrastructuur proeve van landelijk prognoserapport vervanging en renovatie* (tech. rep.). TNO.

Bukhsh, Z. A., Stipanovic, I., & Dorée, A. (2020). Multi-year maintenance planning framework using multi-attribute utility theory and genetic algorithms. *European Transport Research Review*, *12*. https://doi.org/10.1186/s12544-019-0388-y

Cantarella, G. E., & Vitetta, A. (2006). The multi-criteria road network design problem in an urban area. *Transportation*, *33*(6), 567–588. https://doi.org/10.1007/s11116-006-7908-z

Chen, L., & Bai, Q. (2019, April). Optimization in decision making in infrastructure asset management: A review. https://doi.org/10.3390/app9071380

Chen, W., & Zheng, M. (2021, October). Multi-objective optimization for pavement maintenance and rehabilitation decision-making: A critical review and future directions. https://doi.org/10.1016/j.autcon.2021.103840

Choi, J. h. (2019). Strategy for reducing carbon dioxide emissions from maintenance and rehabilitation of highway pavement. *Journal of Cleaner Production*, *209*, 88–100. https://doi.org/10.1016/J.JCLEPRO.2018.10.226

Cloïn, M. (2013, November). Met het oog op de tijd. https://repository.scp.nl/handle/publications/611

Curley, A., Nandakumar, A., & Oirbans, L. (2024, February). Urban strategy manual: San diego.

Damen, M., Oostlander, S., Deursen, W. v., & Sijbrandi, F. (2020). *Een zure appel op z'n tijd* (tech. rep.). https://raad.dordrecht.nl/Documenten/Bijlage-1/RKC-Rapport-Wegafsluitingen-1.pdf

El-Halwagi, M. M. (2006). Overview of optimization. *Process Systems Engineering*, *7*(100), 285–314. https://doi.org/10.1016/S1874-5970(06)80012-3

Fani, A., Naseri, H., Golroo, A., Mirhassani, S. A., & Gandomi, A. H. (2022). A progressive hedging approach for large-scale pavement maintenance scheduling under uncertainty. *International Journal of Pavement Engineering*, *23*(7), 2460–2472. https://doi.org/10.1080/10298436.2020.1859506

Fwa, T., Chan, W., & Hoque, K. (2000). Multiobjective optimization for pavement maintenance programming. *Journal of Transportation Engineering*, *126*(5), 367–374. https://doi.org/10.1061/(ASCE)0733-947X(2000)126:5(367)

Galatioto, F., Huang, Y., Parry, T., Bird, R., & Bell, M. (2015). Traffic modelling in system boundary expansion of road pavement life cycle assessment. *Transportation Research Part D: Transport and Environment*, *36*, 65–75. https://doi.org/10.1016/j.trd.2015.02.007

Gao, L., & Zhang, Z. (2013). Management of pavement maintenance, rehabilitation, and reconstruction through network partition. *Transportation Research Record*, (2366), 59–63. https://doi.org/10.3141/2366-07

Gemeente Amsterdam. (2023, December). Uitleg programma - bruggen en kademuren. https://www.amsterdam.nl/verkeer-vervoer/bruggen-kademuren/uitleg-programma/

G-Kracht. (n.d.). *G-Kracht – een krachtige samenwerking*. Retrieved June 28, 2024, from https://gkracht.nl/

Jin, Y. (2003). A comprehensive survey of fitness approximation in evolutionary computation. https://doi.org/10.1007/s00500-003-0328-5

Jongejans, R. (2023, November). Concept nieuwe hoofdprogrammering programma bruggen en kademuren vanaf 2023 (focus op uitvoerbaarheid en goedkoper realiseren).

Kouwenhoven, M., Muller, J., Thoen, S., Willigers, J., & Jong, G. d. (2023, November). *Values of Time, Reliability and Comfort in the Netherlands 2022* (tech. rep.). KiM Netherlands Institute for Transport Policy Analysis. KiM Netherlands Institute for Transport Policy Analysis. https://www.kimnet.nl/binaries/kimnet/documenten/publicaties/2023/12/04/nieuwe-waarderingskengetallen-voor-reistijd-betrouwbaarheid-en-comfort/Significance_Value+of+Travel+Time+in+the+Netherlands+2022_final+technical+report.pdf

Liu, C., Hammad, A., & Itoh, Y. (1997). Maintenance Strategy Optimization of Bridge Decks Using Genetic Algorithm. *Journal of Transportation Engineering*, *123*(2), 91–100. https://doi.org/10.1061/(ASCE)0733-947X(1997)123:2(91)

Lohman, W., Cornelissen, H., Borst, J., Klerkx, R., Araghi, Y., & Walraven, E. (2023). Building digital twins of cities using the Inter Model Broker framework. *Future Generation Computer Systems*, *148*, 501–513. https://doi.org/10.1016/j.future.2023.06.024

Madni, A. M., Madni, C. C., & Lucero, S. D. (2019). Leveraging digital twin technology in model-based systems engineering. *Systems*, *7*(1). https://doi.org/10.3390/systems7010007

Mallawaarachchi, V. (2023, September). Introduction to Genetic Algorithms — Including Example Code. https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3

Meneses, S., & Ferreira, A. (2013). Pavement maintenance programming considering two objectives: Maintenance costs and user costs. *International Journal of Pavement Engineering*, *14*(2), 206–221. https://doi.org/10.1080/10298436.2012.727994

Meyer-Nieberg, S., Leopold, N., & Uhlig, T. (2020). Introduction to Simulation-Based Optimization. In *Natural computing for simulation-based optimization and beyond* (pp. 1–8). Springer International Publishing. https://doi.org/10.1007/978-3-030-26215-0{_}1

Moorman, M., Groot Wassing, R., & van Dantzig, R. (2022, May). Amsterdams coalitieakkoord 2022-2026. https://www.amsterdam.nl/nieuws/coalitieakkoord/

Municipal Executive of the Municipality of Amsterdam. (2021a). Nadere regels wior. https://lokaleregelgeving.overheid.nl/CVDR656374/1?&show-wti=true

Municipal Executive of the Municipality of Amsterdam. (2021b, March). Verordening werken in de openbare ruimte amsterdam 2021. https://lokaleregelgeving.overheid.nl/CVDR656375/1

NOS. (2020). Amsterdam betaalt prijs voor decennia achterstallig onderhoud kademuren. *NOS*. https://nos.nl/artikel/2346373-amsterdam-betaalt-prijs-voor-decennia-achterstallig-onderhoud-kademuren

Olive, G. (1979). Binomial Functions and Combinatorial Mathematics. *JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS*, *70*, 460–473.

Programma Bruggen en Kademuren. (2020). *Programmaplan bruggen en kademuren. herstellen en verbinden. bouwen aan het fundatment van de stad* (tech. rep.). Gemeente Amsterdam. https://www.amsterdam.nl/verkeer-vervoer/bruggen-kademuren/

Programma Bruggen en Kademuren. (2022, December). *Gefundeerd herstellen: Actieplan bruggen en kademuren 2023-2026* (tech. rep.). Gemeente Amsterdam. https://www.amsterdam.nl/verkeer-vervoer/bruggen-kademuren/

Reniers, L., & Korrel, P. (n.d.). *Alle Amsterdamse bruggen*. Retrieved May 14, 2024, from https://bruggenvanamsterdam.nl/

Rijkswaterstaat. (2023, November). *Rapport staat van de infrastructuur rijkswaterstaat* (tech. rep.). Rijkswaterstaat. https://www.rijksoverheid.nl/documenten/rapporten/2023/12/04/bijlage-3-staat-van-de-infrastructuur-2023

Rothlauf, F. (2011). Optimization methods. In *Design of modern heuristics: Principles and application* (pp. 45–102). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72962-4_3

Swaalf, F., Spruijtenburg, S., Colán), A. (, Shina, R., Weijs, H., & de Heer, P. (2023). *Background literature and decisions for the developed planning-maintenance module for quay walls in amsterdam* [Unpublished work].

TNO. (2023). Traffic+ technical manual.

TU Delft. (2024). Mude textbook. https://mude.citg.tudelft.nl/book/intro.html#

Van der Sluijs, C., Borst, L., & Zantema, K. (2023). *Tonnetje minder: Afweging brugherstel op basis van datagedreven impactanalyse voor bereikbaarheid* (tech. rep.). Municipality of Amsterdam.

Verkeer en Openbare Ruimte. (2022). *Uitgangspunten Verkeersmodel Amsterdam 4.0* (tech. rep.). Municipality of Amsterdam.

Verkeer & Openbare Ruimte. (2015). *Infographic: Het VerkeersModel Amsterdam* (tech. rep.). https://openresearch.amsterdam/nl/page/99963/infographic-verkeersmodel-amsterdam

Zhang, W., & Wang, N. (2017). Bridge network maintenance prioritization under budget constraint. *Structural Safety*, *67*, 96–104. https://doi.org/10.1016/j.strusafe.2017.05.001

A

Python code

This appendix contains code used for the optimisation of maintenance using a genetic algorithm and the urban simulaiton tool (UST) of TNO. See the following sections:

- A.1 Initialising connection with the Urban Strategy Tool
- A.2 Creating controls
- A.3 Precomputation of closures
- A.4: Calculating system travel times
- A.5 Interdependency determination
- A.6 Genetic Algorithm

A.1. Initialising connection with the Urban Strategy Tool

This part of the appendix shows how to communicate with the UST It is imported that the right libraries are imported. Most importantly, the python file called 'usrest' is imported. This file has been provided by Bart van der Poel, medior system engineer of the UST at TNO. The file contains the needed functions to communicate with the UST. The specific python file 'usrest' will not be provided in this research, since access requires permission form the original owner. An account is required to communicate with the UST via the REST API, which can also be requested from the UST team at TNO.

```python

#import libraries
import usrest as us                    # Needed for communication with the UST
import pandas as pd
import geopandas as gpd                # Specifically for determining the location of controls in UST
import os
import matplotlib.pyplot as plt
import json
import urllib3
import itertools

## Start API
ustApi = us.api("https://portal.urbanstrategy.nl:8011")
ustApi.login(email="youremail@mail.com",password="yourpassword")
ustApi.set_bin("us_ams_2022.v90")

## Get information about running models
models=ustApi.get_models()['models']['instances']
print(models) # To retrieve information about bin, name, state, desiredstate, combinedstate


if len(models)==0:
    print('No models running')
else:
    # Print for all models running the setup name and bin
    print(f'{len(models)} models running:')
    column_width = 20
    for id in models:
        setup_name = models[id]['setup']['name']
        bin_name = models[id]['bin']
```

```
        # Format the output to ensure the second value starts at the same spot
        print(f"{setup_name:<{column_width}}{bin_name}")
    print(f'\n')

    # Check for Tiler, Publisher and Traffic+
    Tiler_id = None
    Publisher_id = None
    Traffic_id = None

    for id in models:
        if models[id]['setup']['name'] == 'Tiler':
            Tiler_id = id
            #print(f"Tiler running on {models[id]['bin']}")

        if models[id]['setup']['name'] == 'Publisher␣(IMB5)':
            Publisher_id =id
            #print(f"Publisher running on {models[id]['bin']}")

        if models[id]['bin'].lower() == "us_ams_2022.v90" and models[id]['setup']['name'] == "Traffic+" :
            Traffic_id = id
            print("Reusing␣already␣started␣Traffic+␣model",Traffic_id)

    if Tiler_id != None and Publisher_id != None:
        print('Already␣running␣Tiler␣and␣Publisher')
    else:
        print('Run␣next␣block␣to␣start␣up␣tiler␣and␣publisher')
        # Start up Tiler if none
        if Tiler_id == None:
            ustApi.set_bin("us_ams_2022.v0")            # Needs to be started on bin v0
            started_tiler = ustApi.start_model("Tiler",a_setup_node_name='app-usdt02')
            print(started_tiler)
            Tiler_id=started_tiler['models']['instance']
            print("Started␣new␣Tiler␣model",Tiler_id)

        # Start up Publisher if none
        if Publisher_id == None:
            ustApi.set_bin("us_ams_2022.v0")            # Needs to be started on bin v0
            started_publisher = ustApi.start_model("Publisher␣(IMB5)", a_setup_node_name='app-usdt02')
            print(started_publisher)
            Publisher_id=started_publisher['models']['instance']
            print("Started␣new␣Publisher␣model",Publisher_id)

        ustApi.set_bin("us_ams_2022.v90")   # set bin back to own bin v90

        # Start up Traffic+ if none
        if Traffic_id == None:
            started_traffic=ustApi.start_model("Traffic+",a_setup_node_name='app-usdt02')
            print(started_traffic)
            Traffic_id=started_traffic['models']['instance']
            print("Started␣new␣Traffic␣model",Traffic_id)
```

A.2. Creating controls

In this section the code can be seen for creating controls. First the bridgeset is defined with a list of bridgecodes used by the Municipality of Amsterdam. Second, corresponding road links and bridge names are searched in the database. Then, controls are created. Lastly, the controls are pushed to UST.

```

# Define Bridgeset
bridge_set = ['BRU0063', 'BRU0167', 'BRU0169', 'BRU0423']

# Retrieve data bridges
bridge_info = pd.read_csv('bridge_info_file.csv', delimiter = ',')  # Read full bridge dataset
bridge_dataset = bridge_info.loc[bridge_info['bridge_code'].isin(bridge_set)]  # Create dataset based on
    bridge_set
road_links = bridge_dataset['linknr'].tolist()     # Create road links list

# Create controls using the defined function 'create_controls'
df_controls = create_controls(road_links)

# Add controls to bridge dataset
unique_controls = df_controls[['TABLE_OBJECT_ID', 'OBJECT_ID']].drop_duplicates()
unique_controls.rename(columns={'TABLE_OBJECT_ID': 'linknr', 'OBJECT_ID': 'control'}, inplace=True)
bridge_dataset = pd.merge(bridge_dataset, unique_controls, on='linknr', how='left')

# Save bridge_set including controls for easy retrieval
bridge_dataset.to_csv("bridge_dataset.csv", index=False)

# Push controls to control bin in UST
ustApi.set_collection('controls', df_controls.to_dict(orient='records'), use_base_bin=True)
```

Below the defined function 'create_controls' can be seen. Controls are made for the list of road segments given as input. Controls need to be pushed to the UST in dataframe containing specific columns. To place the control in the interface on the same location as the bridge, the location of the road segment is called. The controls generated with this function close off roads completely by setting the capacity and speed of both sides to zero.

```python

def create_controls(target_roadsegment):
    """
    Creates control(s) for the segment(s) given in the list.

    First retrieve roads df for roads you want to make a control for by using a filter. (This method needed to retrieve location for control on map based on road location)

    Example:
    target_roadsegment = [193764, 209782] (input)
    df_road_shape = gpd.GeoDataFrame.from_features(ustApi.get_collection(collection='roads',
                                                                          filter='object_id in (126141)',
                                                                          format="geo",
                                                                          fields="DIMENSION_ID,OBJECT_ID,SHAPE",
                                                                          geometry=4))
    """

    target_roadsegment_str = ','.join(map(str, target_roadsegment)) # Needed to make the list go to a filter option ([193764, 209782] --> '193764,209782')
    df_road_shape = gpd.GeoDataFrame.from_features(ustApi.get_collection(collection='roads',
                                                   filter= f"object_id in ({target_roadsegment_str}) and dimension_id in (1,2,4)",
                                                   format="geo",
                                                   fields="DIMENSION_ID,OBJECT_ID,SHAPE",
                                                   geometry=4))

    # Give temporary control_id to control based on object_id (so one control for each object)
    # Check for highest control number
    # Get dataframe with control ID's and OBJECT_ID. One object_id has multiple control id's.
    control_num = pd.DataFrame.from_dict(ustApi.get_collection(collection="controls",
                                                               fields="OBJECT_ID",
                                                               use_base_bin=True))
    checklist = df_road_shape['OBJECT_ID'].unique().tolist()
    df_road_shape['CONTROL_ID']= df_road_shape['OBJECT_ID'].apply(lambda row:checklist.index(row))+control_num['OBJECT_ID'].max()+1

    # Give the controls the table id's from the object
    df_road_shape['TABLE_OBJECT_ID']=df_road_shape['OBJECT_ID']
    df_road_shape['TABLE_DIMENSION_ID']=df_road_shape['DIMENSION_ID']
    df_road_shape['TABLE_ID']= df_road_shape['id']

    # Set the temporary control_id as object_id (now for control, not road)
    df_road_shape['OBJECT_ID'] = df_road_shape['CONTROL_ID']

    # Give the control the coordinates of the middle of the road
    df_road_shape['X']= df_road_shape['geometry'].centroid.map(lambda p: p.x)
    df_road_shape['Y']= df_road_shape['geometry'].centroid.map(lambda p: p.y)

    # Same for all controls
    df_road_shape['OBJECT_TABLE'] = 'ROADS'
    df_road_shape['CONTROL_TYPE_ID'] = 17
    df_road_shape['PROJECT_ID'] = 'AMS_THESIS'

    # Names for controls
    df_road_shape['NAME']= 'Closure of road: ' + df_road_shape['TABLE_OBJECT_ID'].astype(str)
    df_road_shape['DESCRIPTION'] = 'Control for full closure'

    #What you want to do with controls (here to close completely)
    #df_road_shape['SIDE']= [[-1, 1]] * len(df_road_shape)
    df_road_shape['FIELD'] = [['CAPACITY_L', 'CAPACITY_R','SPEED_L', 'SPEED_R']] * len(df_road_shape)
    df_road_shape['VALUE'] = '0'

    # Make every field a separate control
    df_road_shape = df_road_shape.explode('FIELD')
    side_mapping = lambda x: -1 if x.endswith('_L') else (1 if x.endswith('_R') else None)
    df_road_shape['SIDE'] = df_road_shape['FIELD'].apply(side_mapping)

    # Fix shape of df
    desired_order = ['id', 'CONTROL_TYPE_ID', 'DESCRIPTION', 'FIELD', 'NAME', 'OBJECT_ID', 'OBJECT_TABLE', 'PROJECT_ID', 'SIDE', 'TABLE_DIMENSION_ID', 'TABLE_ID', 'TABLE_OBJECT_ID', 'VALUE', 'X', 'Y', 'geometry', 'CONTROL_ID']
    df_road_shape = df_road_shape.reindex(columns = desired_order)
    df_road_shape.drop(columns=['CONTROL_ID', 'geometry', 'id'], inplace=True)

    return(df_road_shape)
```

A.3. Precomputation of closures

In this section the code used to precompute the effects of all individual and combined closures are shown.

A.3.1. Activating controls

Below the function for activating a certain list of controls is shown. After a change in active controls, the UST immediately starts running.

```
def activate_controls(target_control_object_ids):
    """
    Runs the model with provided controls on.
    First makes sures all controls are off, than turns on every control in the provided list

    target_control_object_ids = list of control numbers

    """
    controlsenabled = ustApi.get_collection("controlsenabled")   # Check which controls are active
    controlsenabled_df = pd.DataFrame(controlsenabled)

    # show previous controls
    print('Active controls were:', list(controlsenabled_df.loc[controlsenabled_df['ACTIVE'] == 1, 'OBJECT_ID']))

    # Set all controls back to inactive
    controlsenabled_df['ACTIVE'] = 0

    # Set all controls in target controls list to active
    for object_id in target_control_object_ids:
        print(object_id)
        if object_id in controlsenabled_df['OBJECT_ID'].values:
            controlsenabled_df.loc[controlsenabled_df['OBJECT_ID'] == object_id, 'ACTIVE'] = 1
            ustApi.set_collection('controlsenabled', controlsenabled_df.to_dict(orient='records'))
        else:
            ustApi.set_collection('controlsenabled', controlsenabled_df.to_dict(orient='records')) # Make sure the previous active controls are back to 0
            new_row= {'ACTIVE': [1], 'OBJECT_ID': [object_id]}
            new_row_df = pd.DataFrame(new_row)
            ustApi.set_collection('controlsenabled', new_row_df.to_dict(orient='records'))
            #controlsenabled_df = controlsenabled_df.append(new_row, ignore_index=True)

    controlsenabled = ustApi.get_collection("controlsenabled")
    controlsenabled_df = pd.DataFrame(controlsenabled)
    print('Active controls are now:', list(controlsenabled_df.loc[controlsenabled_df['ACTIVE'] == 1, 'OBJECT_ID']))

    # push new controls enabled to simulation and run wait until new results are in
    ustApi.set_collection('controlsenabled', controlsenabled_df.to_dict(orient='records'))
    ustApi.get_model_state([Traffic_id],0,300)
```

A.3.2. Initial road data

Here the code is provided for running the simulation without any bridge closures to obtain the reference scenario. The output is saved as a csv-file for easy retrieval of the initial road data. To make sure no closures are present in the simulation, all controls are deactivated.

```
print("Get initial values")

# Make sure all controls are off
activate_controls([])

# Get road values
ini_values=ustApi.get_collection(collection="roads",filter="dimension_id in (1,2,4)",fields="OBJECT_ID,CAPACITY_L,CAPACITY_R,SPEED_L,SPEED_R,DIMENSION_ID,DISTRICT_ID,INTENSITY,INTENSITY_L,INTENSITY_R,TRAVEL_TIME_L,TRAVEL_TIME_R,LENGTH")
df_ini_values=pd.DataFrame.from_dict(ini_values)

# safe intial values as csv
df_ini_values.to_csv("initial_road_values.csv")
```

A.3.3. Running all individual closures

For every bridge in the provided bridgeset a run is saved with the individual closure of the corresponding bridge. First, the control for full closure of the bridge is activated. When the simulation model is done with the calculations, the road data is collected from the UST. After checking if the bridge is indeed closed for all trafic, the data is saved to a specific folder containing all individual closures.

```python
# Create folder path with new date
folder_path = './closure_results/run_20240610'
os.makedirs(folder_path, exist_ok=True)

# Retrieve dataset
dataset = pd.read_csv('bridge_dataset.csv', delimiter = ',')

dict_df = {}

for index, row in dataset.iterrows():
    control = row['control']
    bridge = row['bridge_code']
    link = row['linknr']

    print(f"Check for {bridge}, close {link} with control {control}")
    #Activate control
    activate_controls([control])

    # Get results
    roads_closure=ustApi.get_collection(collection="roads",filter="dimension_id in (1,2,4)",fields="OBJECT_ID,CAPACITY_L,CAPACITY_R,SPEED_L,SPEED_R,DIMENSION_ID,DISTRICT_ID,INTENSITY,INTENSITY_L,INTENSITY_R,TRAVEL_TIME_L,TRAVEL_TIME_R,LENGTH")
    df_roads_closure=pd.DataFrame.from_dict(roads_closure)

    # Put in dict
    dict_df[link] = df_roads_closure

    if df_roads_closure.loc[df_roads_closure['OBJECT_ID'] ==link]['INTENSITY'].sum() == 0:
        print(f"All intensities are zero for object {link}")
        # Write away results to csv
        df_roads_closure.to_csv(f"{folder_path}/{bridge}.csv")
    else:
        print(f"Not all intensities are zero for object {link}")
```

A.3.4. Running all combinations of two closures

Based on the bridgeset, all possible combinations of two are generated. For every combination a simulation run is executed with both bridges closed off. Before saving the road data to the run specific csv file, it is first checked wheter both bridges are indeed closed for all traffic.

```python
# create search dict based on dataset
bridge_dict = dataset.set_index('bridge_code').T.to_dict()

# Define folder path for combined results:
folder_path = './closure_results/combinedrun_20240610'
os.makedirs(folder_path, exist_ok=True)

# Create bridgelist
bridge_list = dataset['bridge_code'].tolist()

# Create df to safe road results
combi_dict_df = {}

# Use bridgelist to make combinations
for bridge_combination in itertools.combinations(bridge_list, 2):
    print(f'run combination of bridges{bridge_combination}')
    controls = [bridge_dict[code]['control'] for code in bridge_combination]
    links = [bridge_dict[code]['linknr'] for code in bridge_combination]
    print(f"Check for bridges {bridge_combination}, close links {links} with controls {controls}")

    # Activate controls
    activate_controls(controls)

    # Get results
    roads_closure=ustApi.get_collection(collection="roads",filter="dimension_id in (1,2,4)",fields="OBJECT_ID,CAPACITY_L,CAPACITY_R,SPEED_L,SPEED_R,DIMENSION_ID,DISTRICT_ID,INTENSITY,INTENSITY_L,INTENSITY_R,TRAVEL_TIME_L,TRAVEL_TIME_R,LENGTH")
    df_roads_closure=pd.DataFrame.from_dict(roads_closure)

    combi_dict_df[tuple(links)] = df_roads_closure


    # Check if roads are indeed closed
    all_intensities_zero = all(df_roads_closure.loc[df_roads_closure['OBJECT_ID'] == link]['INTENSITY'].sum() == 0 for link in links)

    if all_intensities_zero:
        print(f"All intensities are zero for objects {links}")
        # Write results to csv
        filename = '+'.join(bridge_combination) + '.csv'
        df_roads_closure.to_csv(f"{folder_path}/{filename}", index=False)
    else:
```

```
        print(f"Not␣all␣intensities␣are␣zero␣for␣objects␣{links}")
```

A.4. Calculating system travel times

In this section the code is shown used to calculate the travel times of the different simulation runs. First the function to calculate travel times based on a road collection is shown. Then, the code used to retrieve the road collections for each simulation run and use the travel time calculation function is given.

A.4.1. Function for calculating travel times

In this subsection, the function to calculate the travel times in the system based on a certain road collection dataframe is shown. The function returns the updated road collection dataframe including the column with the calculated travel times for each road link. It also returns a summary showing the total travel time in the system as well as the total travel times in the system for the specific modes separately.

```
import pandas as pd

def calculate_travel_times(df):
    """
    Calculate travel times for a given dataframe by considering
    the travel times in left and right directions and their intensities.

    This function assumes consistent column names across all dataframes.

    The expected column names are:
    - 'TRAVEL_TIME_L': Travel time in the left direction
    - 'INTENSITY_L': Intensity in the left direction
    - 'TRAVEL_TIME_R': Travel time in the right direction
    - 'INTENSITY_R': Intensity in the right direction
    - 'DIMENSION_ID': Identifies the mode of transport or dimension

    Returns:
    - pandas.DataFrame: The original dataframe with additional calculated columns.
    - dict: A summary of total travel times for different dimensions.
    """

    # Calculate travel times for each seperate road section
    # For this the time needed to travel that road section needs to be multiplied with the intensity
    df['CALC_TRAVEL_TIME_L'] = df['TRAVEL_TIME_L'] * df['INTENSITY_L']
    df['CALC_TRAVEL_TIME_R'] = df['TRAVEL_TIME_R'] * df['INTENSITY_R']

    # Calculate travel times for whole road section (sum L and R)
    df['CALC_TOTAL_TRAVEL_TIME'] = df['CALC_TRAVEL_TIME_L'] + df['CALC_TRAVEL_TIME_R']

    # Calculate total system travel time
    total_system_travel_time = df['CALC_TOTAL_TRAVEL_TIME'].sum()

    # Calculate travel time for specific dimensions
    travel_time_d1 = df[df['DIMENSION_ID'] == 1]['CALC_TOTAL_TRAVEL_TIME'].sum()
    travel_time_d2 = df[df['DIMENSION_ID'] == 2]['CALC_TOTAL_TRAVEL_TIME'].sum()
    travel_time_d4 = df[df['DIMENSION_ID'] == 4]['CALC_TOTAL_TRAVEL_TIME'].sum()

    # Create a summary dictionary
    travel_times_summary = {
        "travel_time_total": total_system_travel_time,
        "travel_time_d1": travel_time_d1,
        "travel_time_d2": travel_time_d2,
        "travel_time_d4": travel_time_d4
    }

    return df, travel_times_summary
```

A.4.2. Retrieving road collections and calculate travel times

This section shows how the road collections for the initial scenario and for the individual closures are retrieved. Since the individual closures are stored in a separate folder, a function is prepare all CSV files in the folder for the calculation. For this function the selected bridge set and folder path need to be given as input. It should be noted that in all simulation runs, road 209782 has been excluded from the travel time calculations due to an extremely high value.

Initial run

```
f_initial_road = pd.read_csv('initial_road_values.csv', delimiter = ',')
df_initial_road = df_initial_road.loc[df_initial_road['OBJECT_ID'] != 209782] # Excluding certain road
    because of extreme high travel time

# Use calculate def to get df and summary of results
df_initial, summary_initial = calculate_travel_times(df_initial_road)
print('summary of initial df:\n', summary_initial)

# Calculate percentages
for i in summary_initial.keys():
    print(f"Percentage of {i} of total travel time is {(summary_initial[i]*100)/summary_initial['
        travel_time_total']}")
```

Individual closures

```
# Define a function to create a dictionary of dataframes based on a group of CSV files
def CSVs_to_DFs(csv_names, base_path):
    """
    Loads CSV data for a list of csv files from a specified base path.

    :param csv_names: A list/dict of names of the csv's (bridge names/road sections)
    :param base_path: The base directory where the CSV files are located
    :return: A dictionary mapping csv's (bridge names/ road sections) to their DataFrames
    """

    # Create an empty dict for df's
    dict_dfs = {}

    print("Dataframe contains CSV data for:")
    # Loop through all the csv files in the provided name list
    for csv_name in csv_names:
        # construct file path for each csv
        file_path = os.path.join(base_path, f'{csv_name}.csv')

        # check of filepath exists
        if os.path.isfile(file_path):
            df = pd.read_csv(file_path)
            df = df[df['OBJECT_ID'] != 209782]                    # Removing roads with high travel times
            dict_dfs[csv_name] = df
            print(f'CSV data for {csv_name}: {df.shape[0]} rows')
        else:
            print(f'File not found for {csv_name}: {file_path}')

    return dict_dfs

# Retrieve dataset bridges
dataset = pd.read_csv('bridge_dataset.csv', delimiter = ',')
bridge_list = dataset['bridge_code'].tolist()
bridge_dict = dataset.set_index('bridge_code').T.to_dict()
results_path = './closure_results/run_20240610'

bridge_runs = CSVs_to_DFs(bridge_dict, results_path)

# create dict with total travel times with inidividual closures
bridge_tt_summary = {}

for bridgecode, df in bridge_runs.items():
    df, summary_tt = calculate_travel_times(df)
    df = df.merge(df_initial[['id', 'CALC_TOTAL_TRAVEL_TIME']], on='id', suffixes = (None, '_REF')) # Add
        column of initial df to compare
    df['CHANGE_TT']= (df['CALC_TOTAL_TRAVEL_TIME']-df['CALC_TOTAL_TRAVEL_TIME_REF'])/df['
        CALC_TOTAL_TRAVEL_TIME_REF']*100   # calculate procentual change
    df['DIFF_TT'] = df['CALC_TOTAL_TRAVEL_TIME']-df['CALC_TOTAL_TRAVEL_TIME_REF'] # Calculate difference
        in total travel time for each road id
    bridge_runs[bridgecode] = df # update bridge_runs so it containts the df with change tt
    bridge_tt_summary[bridgecode] = summary_tt
```

Combined closures

```
# Retrieve road collections for combined runs
# Function to retrieve names of closed bridges
def list_files_in_folder(folder_path):
    file_names = os.listdir(folder_path)
    file_names = [file_name.split('.')[0] for file_name in file_names]
    return file_names

# Retrieve combined closure simulation results
combined_runs_path = './closure_results/combinedrun_20240610'
combined_bridges_list = list_files_in_folder(combined_runs_path)
combi_clousre_df_dict = CSVs_to_DFs(combined_bridges_list, combined_runs_path)

# Calculate travel times
```

```
combi_summary_dict = {}

for bridgecombi, df in combi_clousre_df_dict.items():
    df, summary_tt = calculate_travel_times(df)
    df = df.merge(df_initial[['id', 'CALC_TOTAL_TRAVEL_TIME']], on='id', suffixes = (None, '_REF'))        # Add column of initial df to compare
    df['CHANGE_TT']= (df['CALC_TOTAL_TRAVEL_TIME']-df['CALC_TOTAL_TRAVEL_TIME_REF'])/df['CALC_TOTAL_TRAVEL_TIME_REF']*100    # calculate procentual change
    df['DIFF_TT'] = df['CALC_TOTAL_TRAVEL_TIME']-df['CALC_TOTAL_TRAVEL_TIME_REF']                    # Calculate difference in total travel time for each road id
    combi_clousre_df_dict[bridgecombi] = df
    combi_summary_dict[bridgecombi] = summary_tt

bridges_pd_combi = pd.DataFrame.from_dict(combi_summary_dict)
initial_pd_combi = pd.DataFrame.from_dict(summary_initial, orient='index', columns = ["initial"])
result_df_combi = pd.concat([initial_pd_combi, bridges_pd_combi], axis=1)
result_df_combi

combi_changes_df = pd.DataFrame.copy(result_df_combi)

for col in combi_changes_df.columns[1:]:        #loop over all columns except the intial column
    combi_changes_df[col]= combi_changes_df[col]-combi_changes_df['initial']
```

A.4.3. Determining the additional travel times

For each scenario of closures, the additional travel times in the system can be calculated by looking at the change in travel times compared to the initial scenario. This is easy done by subtracting the initial travel times from the travel times after closure. The result is a dataframe containing the additional travel times in the system (total, car, freight and bike) for the closure of each individual bridge or combined closure. Using the bridge info dataset, the bridge codes are changed into bridge names for easy understanding. The code is provided below.

individual closures

```
    # Create changes df
    bridge_tt_summary

    bridges_pd = pd.DataFrame.from_dict(bridge_tt_summary)
    initial_pd = pd.DataFrame.from_dict(summary_initial, orient='index', columns = ["initial"])
    result_df = pd.concat([initial_pd, bridges_pd], axis=1)

    # Overview traveltimes initial and with closures
    result_df

    changes_df = pd.DataFrame.copy(result_df)

    for col in changes_df.columns[1:]:        #loop over all columns except the intial column
        changes_df[col]= changes_df[col]-changes_df['initial']

    # def to change bridge codes to bridge names for easier understanding

    def bridgecode_to_name(df):
        # Create a rename dictionary for index (rows)
        rename_index_dict = {code: bridge_dict[code]['Name'] for code in df.index if code in bridge_dict}
        # Create a rename dictionary for columns
        rename_columns_dict = {code: bridge_dict[code]['Name'] for code in df.columns if code in bridge_dict}

        # Rename the DataFrame
        df_renamed = df.rename(index=rename_index_dict, columns=rename_columns_dict)
        return df_renamed

    changes_df_names = bridgecode_to_name(changes_df)
```

Combined closures

```
    bridges_pd_combi = pd.DataFrame.from_dict(combi_summary_dict)
    initial_pd_combi = pd.DataFrame.from_dict(summary_initial, orient='index', columns = ["initial"])
    result_df_combi = pd.concat([initial_pd_combi, bridges_pd_combi], axis=1)
    result_df_combi

    combi_changes_df = pd.DataFrame.copy(result_df_combi)

    for col in combi_changes_df.columns[1:]:        #loop over all columns except the intial column
        combi_changes_df[col]= combi_changes_df[col]-combi_changes_df['initial']

    combi_changes_df
```

A.5. Interdependency determination

The interdependencies between the bridges are determined by the change in travel time on the bridge after another bridge is closed. After closing a certain bridge, all other bridges are checked in the road collection dataset to see if there is an reduction or increase of the total travel time on their road links. Below the code is given that is used to compute the change in travel time, both absolute as percentage change, for all bridge combinations.

```
# Create empty DataFrames for the differences and percentage changes
df_travel_time_total_diff = pd.DataFrame(index=bridge_list, columns=bridge_list)
df_travel_time_total_prc  = pd.DataFrame(index=bridge_list, columns=bridge_list)

# Populate the DataFrames
for bridge, bridge_info in bridge_dict.items():
    for other_bridge, other_bridge_info in bridge_dict.items():
        other_linknr = other_bridge_info['linknr']
        if bridge != other_bridge:
            # Get difference in total travel_time when the bridge is closed
            difference_total = bridge_runs[bridge].loc[
                bridge_runs[bridge]['OBJECT_ID'] == other_linknr, 'DIFF_TT'].sum()

            df_travel_time_total_diff.at[other_bridge, bridge] = difference_total

            initial_tt_total =df_initial.loc[df_initial['OBJECT_ID'] == other_linknr, '
                CALC_TOTAL_TRAVEL_TIME'].sum()

            # Calculate the percentage change
            if initial_tt_total != 0:
                percent_change_total = (difference_total / initial_tt_total) * 100
                df_travel_time_total_prc.at[other_bridge, bridge] = percent_change_total
            else:
                df_travel_time_total_prc.at[other_bridge, bridge] = None

# Print the DataFrames
print("Total␣Travel␣Time␣Difference␣DataFrame:\n", df_travel_time_total_diff)
print("\nTotal␣Travel␣Time␣Percentage␣Change␣DataFrame:\n", df_travel_time_total_prc)
```

For every combination of bridge closures there are two values showing the percentage change in travel time. One for showing the percentage change in travel time on one bridge cause by the other and vice versa. These two values need to be combined into one value to be able to see if the combination of the two bridges is good or bad in the maintenance schedule. For this, the average of the two values is taken, resulting in one average travel time effect for each bridge combination.

```
# Create an empty DataFrame for the average values
average_closure_effect_df = pd.DataFrame(index=df_travel_time_total_prc.index, columns=
    df_travel_time_total_prc.columns)

# Iterate through the upper triangle of the DataFrame
for i in range(len(df_travel_time_total_prc.columns)):
    for j in range(i + 1, len(df_travel_time_total_prc.columns)):
        bridge_1 = df_travel_time_total_prc.columns[i]
        bridge_2 = df_travel_time_total_prc.columns[j]

        # Get the values from both positions
        value_1 = df_travel_time_total_prc.at[bridge_1, bridge_2]
        value_2 = df_travel_time_total_prc.at[bridge_2, bridge_1]

        # Calculate the average
        avg = np.nanmean([value_1, value_2])

        # Store the average in the new DataFrame
        average_closure_effect_df.at[bridge_1, bridge_2] = avg
        average_closure_effect_df.at[bridge_2, bridge_1] = avg

# Print the average DataFrame
print("Average␣closure␣effect␣matrix:")
print(average_closure_effect_df)
```

A.6. Genetic Algorithm

Import libraries:

```
import numpy as np

from pymoo.factory import get_algorithm, get_crossover, get_mutation, get_sampling
from pymoo.optimize import minimize
from pymoo.core.problem import Problem

```

```python
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
from pymoo.config import Config
Config.show_compile_hint = False
```

In the Genetic Algorithm (GA), the solutions are written as chromosomes. These chromosomes consist of a starting time step and a choice for a certain maintenance duration for each bridge of the selected bridges. For evaluation of the travel time caused by the planning, it is needed to know what bridges are in maintenance at the same time and for how long. Therefore, the chromosome needs to be transformed to a schedule showing the bridges in maintenance for each time step. The code below shows how the choromosomes are transformed to a schedule.

```python
# Rewrite choromosome to schedule
def chromosome_to_schedule(chromosome):     # example chromosome: [5, 2, 0, 1, 1, 0]
    start_end_bridges = [(chromosome[i], chromosome[i]+(duration[chromosome[i + 1]]-1)) for i in range(0, len(chromosome), 2)] #create a list with for every bridge the start and end timestep
    print_check(start_end_bridges) # example: [(5, 12), (0, 5), (1, 10)]

    # create information to know for every timestep which bridges are in maintenance.
    schedule_per_timestep = {}
    for timestep in range(0, max(moment[1] for moment in start_end_bridges)+1): #range from 0 to max timestep (not n_timestep because maintenance can exceed that) (! start with 0)
        bridges_in_maintenance = []         #list of bridges in maintenance at one timestep
        for bridge, start_end_bridge in enumerate(start_end_bridges):
            if start_end_bridge[0] <= timestep <= start_end_bridge[1]:
                bridges_in_maintenance.append(bridge)
        # Convert to a tuple for easier comparison
        schedule_per_timestep[timestep] = tuple(sorted(bridges_in_maintenance))
    print_check(schedule_per_timestep) # example: {0: (1,), 1: (1, 2), 2: (1, 2), 3: (1, 2), 4: (1, 2), 5: (0, 1, 2), 6: (0, 2), 7: (0, 2), 8: (0, 2), 9: (0, 2), 10: (0, 2), 11: (0,), 12: (0,)}
    return(schedule_per_timestep)
```

To evaluate the effect of a certain schedule on travel time using the isolation method (or summation method), all individual closure effect are summed. See the code below:

```python
# Evaluate total travel time by summing
def evaluate_totaltraveltime_simple(x):
    F_list = []

    for chromosome in x:
        schedule_per_timestep = chromosome_to_schedule(chromosome)

        # calculate traveltimes for each timestep and total for specific schedule
        tt_effect_per_timestep = {}
        for timestep in range(len(schedule_per_timestep)):
            tt_effect = 0
            for i in (schedule_per_timestep[timestep]):
                tt_effect= tt_effect + (t_bridge[i])
            tt_effect_per_timestep[timestep] =tt_effect
        print_check(tt_effect_per_timestep) #example {0: 1399, 1: 1699, 2: 1699, 3: 1699, 4: 1699, 5: 2699, 6: 1300, 7: 1300, 8: 1300, 9: 1300, 10: 1300, 11: 1000, 12: 1000}
        F=sum(tt_effect_per_timestep.values())
        F_list.append(F)

    print_check('F list time', F_list) #example [19394, 19592, 16592, 16597]
    F_array = np.array(F_list)
    F_array.reshape(len(F_list),1)
    return F_array
```

To evaluate the effect of a certain schedule on travel time using the interdependency method a different evaluation is used. The code used for this can be seen below:

```python
# Evaluate total travel time by using correlations
def evaluate_totaltraveltime_correlation(x):
    F_list = []

    for chromosome in x:
        # STEP 1: Get unique situations and their occurance from planning
        print_check(schedule_per_timestep)
        schedule_list = list(schedule_per_timestep.values())
        unique_situations = set(schedule_per_timestep.values())

        situation_counts = {situation: schedule_list.count(situation) for situation in unique_situations}
        print_check(situation_counts) # example situation_counts = {(1, 2): 4, (0,): 2, (1,): 1, (0, 1, 2): 1, (0, 2): 5}

        # STEP 2: calculate additional travel time for each unique situaion & add to total
        runs_bridges = []
```

```
        tt_total = 0
        for situation, count in situation_counts.items(): # example situation_counts = {(1, 2): 4, (0,): 2, (1,): 1, (0, 1, 2): 1, (0, 2): 5}
            print_check('Check for situation', situation, 'with count', count) #(1, 2), 4
            bridges = [bridge_list[bridge] for bridge in situation]
            print_check('bridges to evaluate:', bridges) #['BRU0167', 'BRU0169']
            runs_bridges.append(bridges)
            tt_situation = 0
            if 0 < len(bridges) <=2:
                search_column = '+'.join(bridges)
                print_check('search tt of: ', search_column) # BRU0167+BRU0169
                tt_bridges = together_df.loc['travel_time_total', search_column]
                tt_situation += tt_bridges
                print_check(tt_bridges, 'travel time 1 day for', bridges)
            elif len(bridges)>=3:
                print_check('use corerlations to get tt')
                bridges_copy = bridges.copy()
                while len(bridges_copy)>2:
                #search for the max pair in bridgelist
                    abs_effects = {abs(average_closure_effect_df.loc[bridge1, bridge2]): [bridge1, bridge2] for bridge1, bridge2 in itertools.combinations(bridges_copy, 2)}
                    print_check(abs_effects)
                    max_pair = abs_effects[max(abs_effects.keys())]
                    print_check('bridgeset with highest correlation', max_pair)
                    search_column = '+'.join(max_pair)
                    print_check('search tt of: ', search_column)
                    tt_bridges = together_df.loc['travel_time_total', search_column]
                    print_check(tt_bridges, 'travel time 1 day for', max_pair)
                    tt_situation += tt_bridges
                    # Remove already evaluated bridges from to evaluate list
                    for bridge in max_pair:
                        bridges_copy.remove(bridge)
                    print_check(bridges_copy, 'bridges left to evaluate')
                    print_check(tt_situation, 'travel time after this set')
                if len(bridges_copy)>0:
                    search_column = '+'.join(bridges_copy)
                    print_check('search tt of: ', search_column) # BRU0167+BRU0169
                    tt_bridges = together_df.loc['travel_time_total', search_column]
                    print_check(tt_bridges, 'travel time 1 day for', bridges_copy)
                    tt_situation += tt_bridges
            print_check(tt_situation, 'travel time 1 day total (the whole situation)')
            tt_timespan = tt_situation * count
            print_check ('total timespan tt', tt_timespan)
            tt_total += tt_timespan
            print_check('\n')
        print_check('tt for whole planning:', tt_total)
        print_check(runs_bridges)
        F=tt_total
        F_list.append(F)
        print_check('\n')


    print_check(F_list) #example [19394, 19592, 16592, 16597]
    F_array = np.array(F_list)
    F_array.reshape(len(F_list),1)
    return F_array
```

Evaluation of total costs for maintenance in a certain schedule:

```
    def evaluate_cost(x):
    F_list = []

    for chromosome in x:
        duration_choices = [chromosome[i] for i in range(1,len(chromosome),2)]
        cost = 0
        for duration_choice in duration_choices:
            cost += cost_function[duration_choice]
        F_list.append(cost)

    print_check('F list cost', F_list)
    F_array = np.array(F_list)
    F_array.reshape(len(F_list), 1)
    return F_array
```

Evaluation of overrunning the time window. A penalty is given for each evaluation

```
    def evaluate_overrun(x):
    F_list = []

    for chromosome in x:
        schedule_per_timestep = chromosome_to_schedule(chromosome)
        last_step = max(schedule_per_timestep)
        overrun = max(0, last_step - (n_timesteps-1))
```

```
        penalty = 10000 * overrun
        F_list.append(penalty)

    print_check('F list overrun', F_list)
    F_array = np.array(F_list)
    F_array.reshape(len(F_list), 1)
    return F_array
```

Method selection for GA. Standard GA from pymoo.

```
# General method for GA
method = get_algorithm("ga",
                       pop_size= n_population,
                       sampling=get_sampling("int_random"),
                       crossover=get_crossover("int_sbx", prob=1.0, eta=3.0),
                       mutation=get_mutation("int_pm", eta=3.0),
                       eliminate_duplicates=True)
```

Input values for GA.

```
#input information
n_bridges = len(bridge_list)
n_timesteps = 24
n_options = 3
t_bridge = list_ttt
duration = [6,3,2]
cost_function = [8000000, 8800000, 10400000]
weight = 10
print_checks = False # Shows in between results of code
n_population = 4
n_generations = 10

#create upperbound list where for every bridge the first value is the max timestep and the second value
    the max option. Since python starts with 0 it's the number of timesteps/options -1
x_highbound = [n_options-1 if i % 2 else n_timesteps-1 for i in range(n_bridges*2)]
```

Running the algorithm with all evaluations:

```
print_checks = True
class MyProblem_three(Problem):

    def __init__(self):
        super().__init__(n_var= n_bridges*2, n_obj=1, n_constr=0, xl=0, xu=np.array(x_highbound), type_var
            =int)

    def _evaluate(self, x, out, *args, **kwargs):

        #Get vectors of all evaluations
        F_traveltime = evaluate_totaltraveltime_simple(x)
        F_cost = evaluate_cost(x)
        F_overrun = evaluate_overrun(x)

        print_check('F_traveltime:', F_traveltime)
        print_check('F_cost:', F_cost)
        print_check('F_overrun:', F_overrun)

        F_sum = weight * F_traveltime + F_cost + F_overrun

        out["F"] = F_sum
        print_check(out["F"]) #example [19394 19592 16592 16597]


res = minimize(MyProblem_three(),
               method,
               termination=('n_gen', n_generations),
               seed=2,
               save_history=True,
               verbose=False)

print("Best solution found: %s" % res.X)
print("Function value: %s" % res.F)
print("Constraint violation: %s" % res.CV)
```

B Optimisation results

In this appendix, the details of the optimal results are shown. First the optimal results of the isolation method are presented. Then the details of the optimal results of the interdependency method are shown. In the last section, the differences in additional travel time between the estimation and simulation for both optimisation methods are presented for all seeds.

B.1. Optimal results of the isolation method

In Table B.1 the solutions and corresponding objective values are shown for every seed. The visualisation of these solutions can be seen in Figure B.1

Table B.1: Optimisation results using the isolation method for 20 seeds

Seed	Solution	Objective value	Additional travel time	Maintenance cost	Penalty overrun
0	[0 0 15 0 10 0 9 1]	34352312.174652405	155231.21746524033	32800000	0
1	[10 0 4 0 4 0 5 1]	34352312.174652405	155231.21746524033	32800000	0
2	[10 0 13 0 10 0 14 1]	34352312.174652405	155231.21746524033	32800000	0
3	[1 0 6 0 12 0 17 1]	34352312.174652405	155231.21746524033	32800000	0
4	[4 0 9 0 17 0 11 1]	34352312.174652405	155231.21746524033	32800000	0
5	[8 0 13 0 14 0 6 1]	34352312.174652405	155231.21746524033	32800000	0
6	[7 0 17 0 0 0 11 1]	34352312.174652405	155231.21746524033	32800000	0
7	[13 0 16 0 16 0 7 1]	34352312.174652405	155231.21746524033	32800000	0
8	[17 0 4 0 6 0 14 1]	34352312.174652405	155231.21746524033	32800000	0
9	[6 0 11 0 10 0 15 1]	34352312.174652405	155231.21746524033	32800000	0
10	[10 0 7 0 15 0 11 1]	34352312.174652405	155231.21746524033	32800000	0
11	[0 0 8 0 11 0 8 1]	34352312.174652405	155231.21746524033	32800000	0
12	[13 0 16 0 18 0 3 1]	34352312.174652405	155231.21746524033	32800000	0
13	[6 0 0 0 10 0 12 1]	34352312.174652405	155231.21746524033	32800000	0
14	[6 0 17 0 12 0 7 1]	34352312.174652405	155231.21746524033	32800000	0
15	[8 0 0 0 9 0 7 1]	34352312.174652405	155231.21746524033	32800000	0
16	[5 0 14 0 8 0 8 1]	34352312.174652405	155231.21746524033	32800000	0
17	[10 0 3 0 17 0 3 1]	34352312.174652405	155231.21746524033	32800000	0
18	[6 0 2 0 17 0 0 1]	34352312.174652405	155231.21746524033	32800000	0
19	[6 0 13 0 7 0 2 1]	34352312.174652405	155231.21746524033	32800000	0

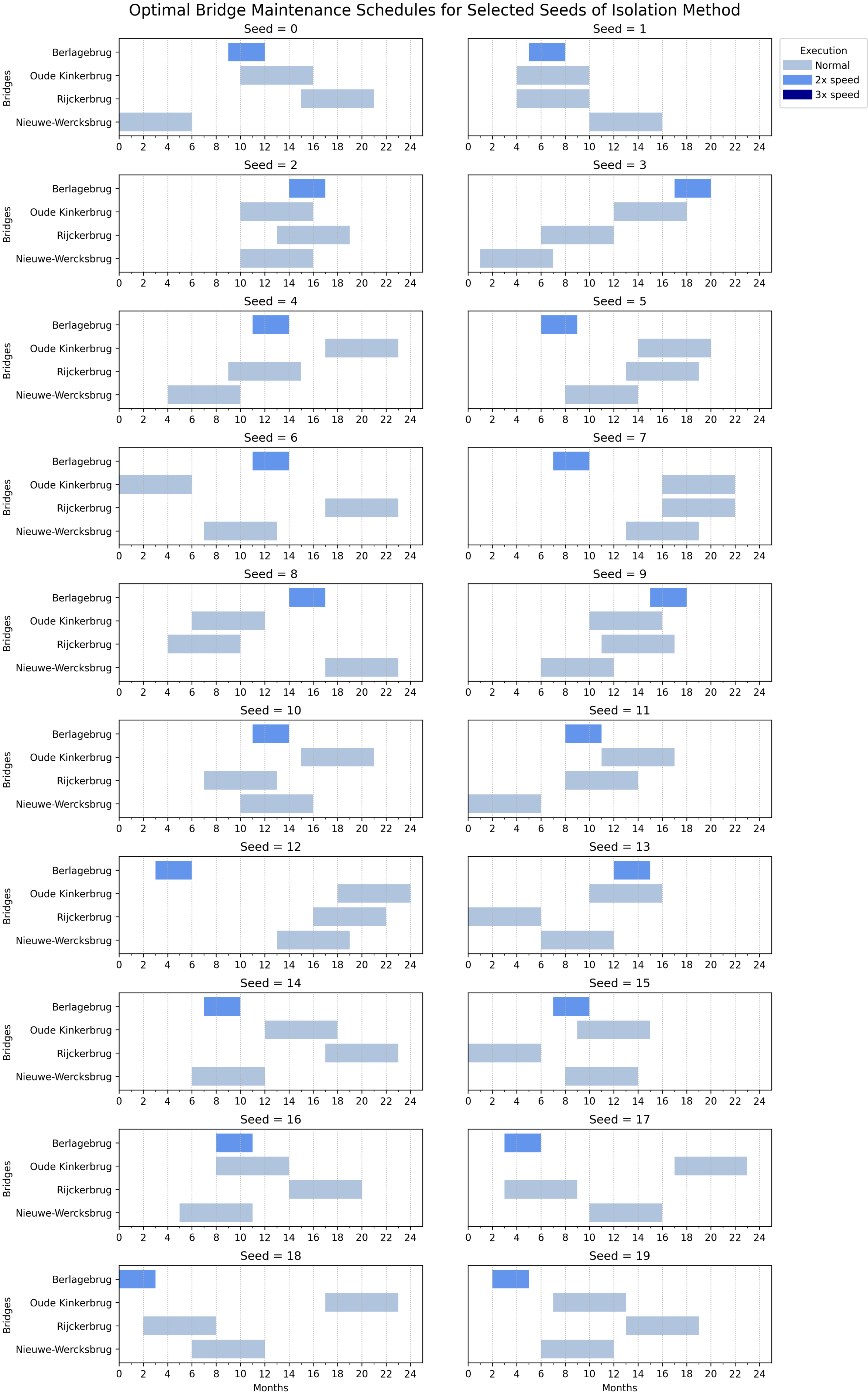


Figure B.1: The visualisation of all optimal schedules of the isolation method for all 20 seeds

B.2. Optimal results of the interdependency method

In Table B.2 the solutions and corresponding objective values are shown for every seed. The visualisation of these solutions can be seen in Figure B.2

Table B.2: Optimisation results using the interdependency method for 20 seeds

Seed	Solution	Objective value	Additional travel time	Maintenance cost	Penalty overrun
0	[1 0 1 0 9 0 3 1]	34209165.605524205	140916.5605524207	32800000	0
1	[6 0 6 0 13 0 9 1]	34209165.605524205	140916.5605524207	32800000	0
2	[12 0 12 0 6 0 3 1]	34209165.605524205	140916.5605524207	32800000	0
3	[0 0 0 0 11 0 17 1]	34209165.605524205	140916.5605524207	32800000	0
4	[13 0 13 0 6 0 1 1]	34209165.605524205	140916.5605524207	32800000	0
5	[7 0 7 0 16 0 9 1]	34209165.605524205	140916.5605524207	32800000	0
6	[17 0 17 0 0 0 13 1]	34209165.605524205	140916.5605524207	32800000	0
7	[5 0 5 0 11 0 7 1]	34209165.605524205	140916.5605524207	32800000	0
8	[12 0 12 0 0 0 13 1]	34209165.605524205	140916.5605524207	32800000	0
9	[6 0 6 0 9 0 15 1]	34209165.605524205	140916.5605524207	32800000	0
10	[8 0 8 0 1 0 12 1]	34209165.605524205	140916.5605524207	32800000	0
11	[8 0 8 0 3 0 9 1]	34209165.605524205	140916.5605524207	32800000	0
12	[4 0 4 0 16 0 3 1]	34209165.605524205	140916.5605524207	32800000	0
13	[6 0 6 0 11 0 17 1]	34209165.605524205	140916.5605524207	32800000	0
14	[13 0 13 0 17 0 13 1]	34209165.605524205	140916.5605524207	32800000	0
15	[10 0 10 0 14 0 7 1]	34209165.605524205	140916.5605524207	32800000	0
16	[1 0 1 0 0 0 14 1]	34209165.605524205	140916.5605524207	32800000	0
17	[3 0 3 0 17 0 3 1]	34209165.605524205	140916.5605524207	32800000	0
18	[7 0 7 0 14 0 9 1]	34209165.605524205	140916.5605524207	32800000	0
19	[5 0 5 0 4 0 19 1]	34209165.605524205	140916.5605524207	32800000	0

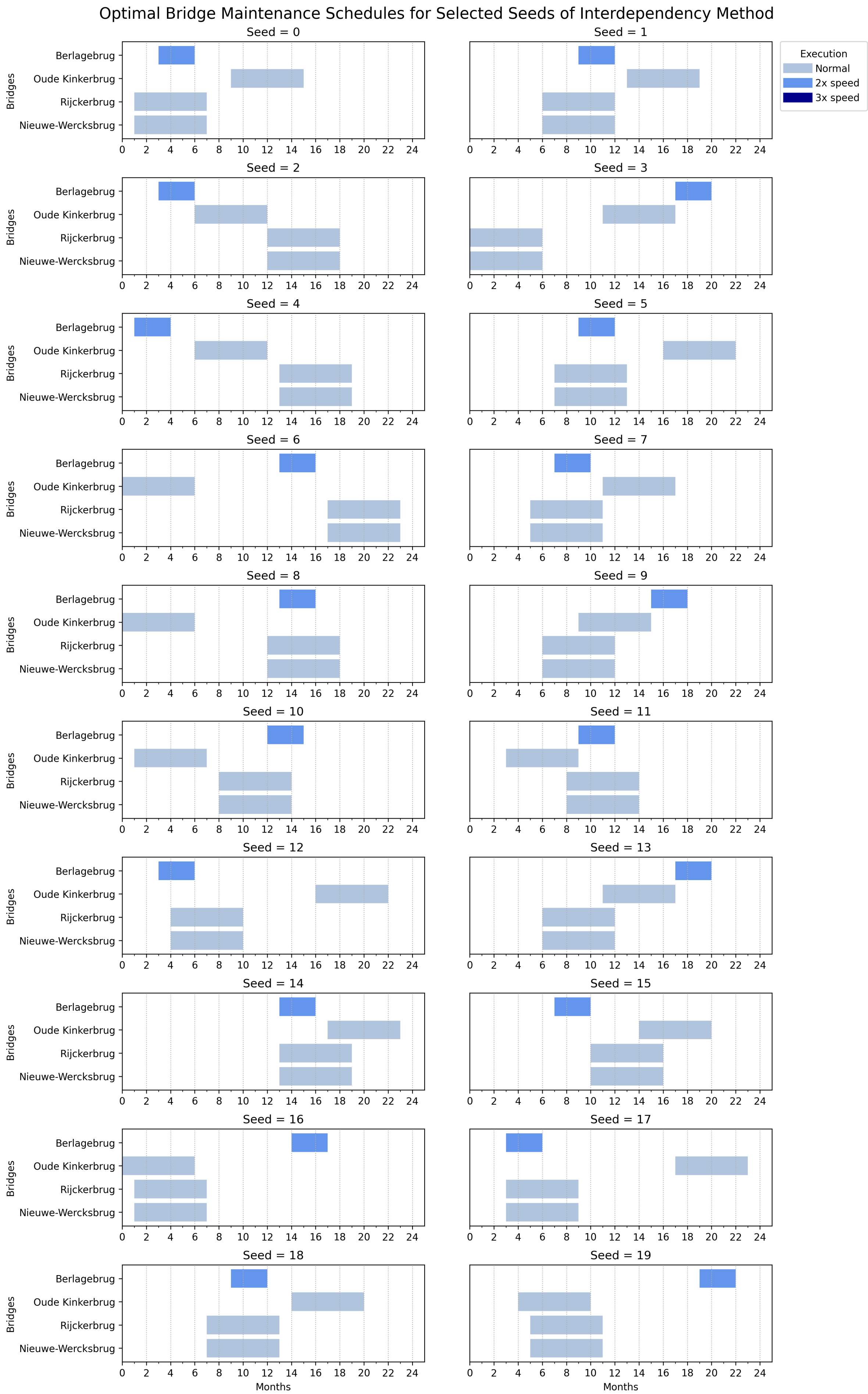


Figure B.2: The visualisation of all optimal schedules of the interdependency method for all 20 seeds

B.3. Differences estimation and simulation of additional travel times

Table B.3: Differences in estimated and simulated additional travel times in hours for 20 seeds (rounded to two decimal places)

	Isolation method			Interdependency method		
Seed	Estimated (h)	Simulated (h)	Difference (h)	Estimated (h)	Simulated (h)	Difference (h)
0	155231.22	157433.76	2202.54	140916.56	141178.23	261.67
1	155231.22	168543.6	13312.38	140916.56	141178.23	261.67
2	155231.22	156538.79	1307.57	140916.56	140916.56	0.0
3	155231.22	152859.99	-2371.23	140916.56	140916.56	0.0
4	155231.22	152907.44	-2323.77	140916.56	140916.56	0.0
5	155231.22	163716.77	8485.55	140916.56	141178.23	261.67
6	155231.22	155239.34	8.13	140916.56	140916.56	0.0
7	155231.22	161552.62	6321.41	140916.56	141178.23	261.67
8	155231.22	163925.03	8693.81	140916.56	141178.23	261.67
9	155231.22	164407.11	9175.89	140916.56	147874.94	6958.38
10	155231.22	148689.58	-6541.64	140916.56	141091.01	174.45
11	155231.22	161813.58	6582.36	140916.56	143497.69	2581.13
12	155231.22	156913.71	1682.49	140916.56	141091.01	174.45
13	155231.22	156149.2	917.98	140916.56	143236.02	2319.46
14	155231.22	157416.86	2185.64	140916.56	145817.15	4900.59
15	155231.22	157505.0	2273.78	140916.56	145555.48	4638.92
16	155231.22	156794.26	1563.05	140916.56	152513.86	11597.29
17	155231.22	155293.22	62.0	140916.56	141178.23	261.67
18	155231.22	150480.33	-4750.89	140916.56	141178.23	261.67
19	155231.22	157417.09	2185.88	140916.56	152513.86	11597.29

C Weight sensitivity

On the next page the corresponding bridge maintenance schedules of the transition weights of maintenance duration found by the sensitivity analysis are shown. The analysis has been performed with a consistend seed, in this case a seed of 3. On the left, the schedules for the isolation method are shown, on the right those of the interdependency method. The interdependency has two schedules less than the isolation method, since the transitions of the Rijckerbrug and Nieuwe-Wercksbrug happen at the same weight.

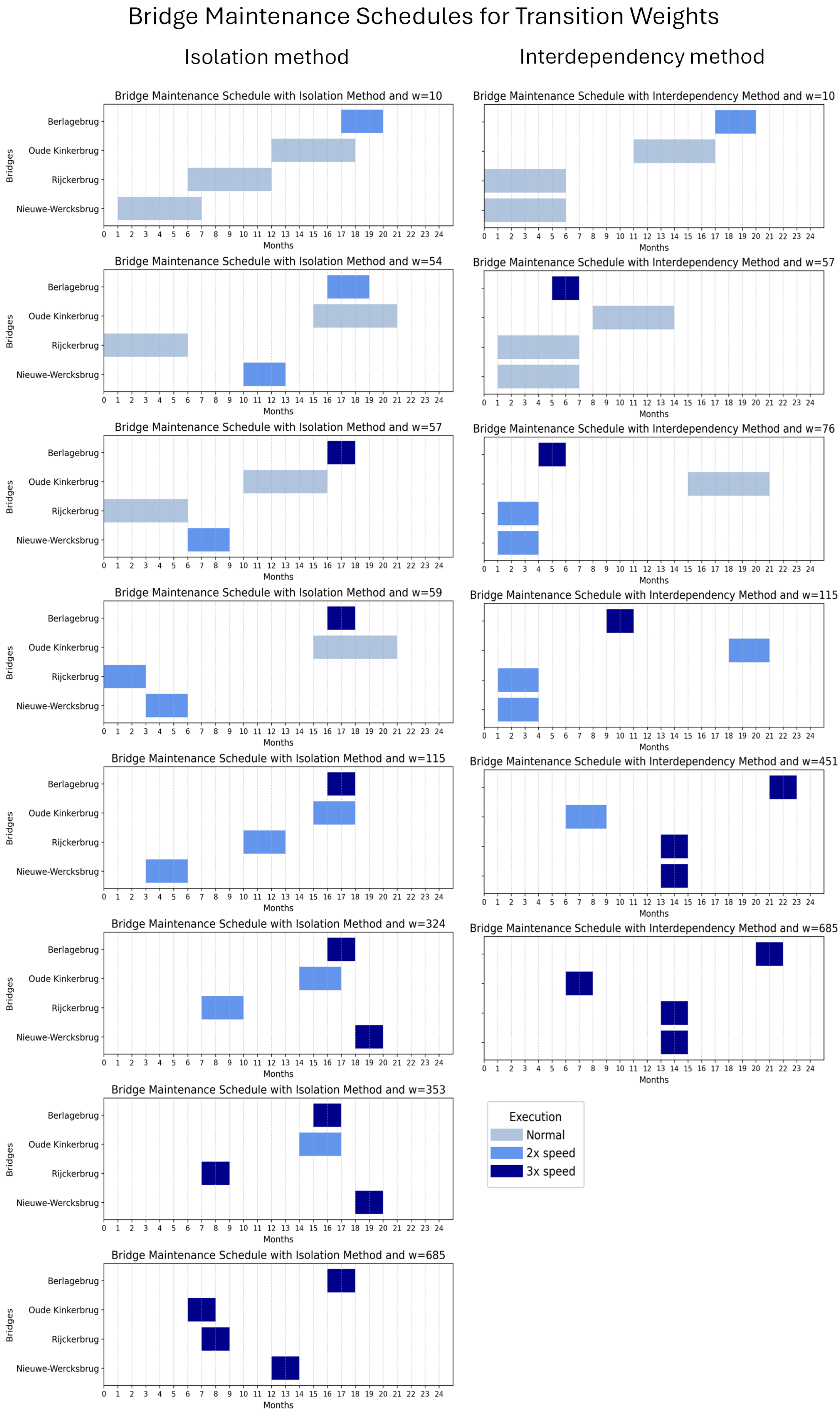


Figure C.1: Bridge maintenance schedules for the transition weights of maintenance duration

D

Scientific paper

The scientific paper is part of the requirements for the thesis of the master's of Transport, Infrastructure and Logistics. Based on the research conducted for this thesis, a scientific paper has been written, which is attached to this document.

Optimising Bridge Maintenance Planning Considering Hindrance to Road Users

An improved method combining optimisation and simulation applied to the city of Amsterdam

M. (Mela) Sagasser[a]

[a]*Department of Transport, Infrastructure and Logistics, Delft University of Technology, Delft, the Netherlands*

Abstract

Amsterdam's extensive network of bridges and quay walls is facing a critical maintenance backlog due to structural underfunding infrastructure maintenance. The many upcoming bridge maintenance projects to ensure safety are expected to cause significant more hindrance to road users. However, complex road networks with high traffic density in urban areas like Amsterdam make it difficult to foresee how the effects of different maintenance projects interact and how the projects can be planned efficiently to reduce hindrance.

This study aims to optimise the planning of a given set of bridge maintenance projects to reduce hindrance to road users, combining a genetic algorithm with traffic simulations using a new estimation method to improve the assessment of simultaneous closures. The genetic algorithm selects the optimal starting time and execution duration for each bridge while minimising maintenance cost and additional travel times for the modalities car, freight and bicycle. Using the fast static traffic simulation of the Urban Strategy Tool of TNO, a new estimation method for evaluating simultaneous closures is proposed, accounting for interdependencies between bridges. Results of a case study on four urban bridges in Amsterdam show that the new method provides more accurate estimates of additional travel time compared to previous methods and generates better solutions. Moreover, the fast proposed optimisation framework makes it possible to evaluate multiple scenario's in reasonable time to assist in decision making of maintenance planning.

Keywords: Bridge maintenance, Maintenance planning, Optimisation, Genetic algorithm, Traffic simulation, Meta-model, Road users hindrance, Travel time

1. Introduction

The city of Amsterdam is currently facing a significant infrastructure crisis due to the poor condition of their extensive network of canals, consisting of many historical bridges and quay walls. Like many countries, the Netherlands has structurally underfunded infrastructure maintenance, leading to a substantial backlog of necessary renovation and renewal work [1; 2]. To ensure safety and catch up on the backlog, many maintenance projects need to be executed in the upcoming decades, resulting in severe hindrance to the city [3].

Currently, structures are selected for maintenance based on their state using a condition-based-approach. This results in maintenance projects spread over the city, causing hindrance to road users on many locations. Bridge closures are particularly problematic due to their vital role as connectors in the road network, significantly increasing travel times. In cases of severe traffic disruptions, the municipality considers more costly maintenance executions that reduce the traffic impact, such as reducing the duration by employing more costly work hours per week to reduce the total obstruction time. The municipality therefore needs to weigh the traffic impact of the maintenance plans with their costs. However, the complex urban road network of Amsterdam with high traffic demand and dependencies between bridges make it difficult to foresee the effect of simultaneous closures on traffic, complicating the efficient planning of bridge maintenance to reduce hindrance to road users.

While recent efforts have been made to optimise maintenance schedules, current methods fail to fully account for the interdependencies between multiple simultaneous closures, which can lead to sub optimal planning and unnecessary hindrance to road users. Traffic simulation can improve the assessment of the effect of simultaneous closures on traffic, but those are rarely combined with optimisation due to computational challenges. The development of faster traffic simulation models such as TNO's Urban Strategy Tool (UST) opens new possibilities for their integration into optimisation. However, previous work using these optimisations to estimate the effect maintenance plans on traffic did not take dependencies between simultaneous closures into account [4].

The objective of this research is to develop a bridge maintenance planning optimisation method that minimise additional travel time for road users while balancing maintenance execution costs. This paper proposes a novel approach using a genetic algorithm combined with an improved estimation of additional travel time using traffic simulations from TNO's UST. The estimation improves the assessment of the traffic impact by considering interdependencies between simultaneous closures. This study contributes to scientific literature by proposing an improved method for combining traffic simulations with optimisation of bridge maintenance. In addition, the developed optimisation will help decision makers to balance minimising hindrance and maintenance cost in bridge maintenance planning.

The remainder of this paper is organised as follows. Section 2 presents a literature review of the existing studies of maintenance optimisation for bridges and similar infrastructure projects. In Section 3 the methodology for optimisation using traffic simulations with an improved assessment of simultaneous closures is presented. This methodology has been used on a case study of Amsterdam in Section 4. Based on the results of the application, conclusions and recommendations are given in Section 5 and 6.

2. Literature Review

Optimisation of maintenance has been widely researched, however, less studies have focused on bridge maintenance or on reducing hindrance to road users. The literature review covers current maintenance optimisation in road networks by reviewing both bridge specific and other infrastructure maintenance optimisation.

2.0.1. Bridge maintenance optimisation

Several studies have been conducted on optimisation of bridge maintenance based on the state of the bridges, given budgetary constraints [2]. An example is [5], where a budget optimisation is proposed for short- and long-term maintenance, repair and replacement (MRR) plans for bridge decks. For the optimisation, a performance model developed to define the current condition of bridges and predict their future deterioration rate. Using a genetic algorithm, optimal MRR plans were generated while maintaining a defined level of service and budget constraints.

Bukhsh et al. [6] developed a multi-objective optimisation for finding the optimal maintenance planning for a network of high-way bridges over a multi-year planning period. The multiple objectives of the optimisation were maximising the performance level of bridges and minimising the maintenance cost. A genetic algorithm was applied to find the optimal maintenance plan, selecting the MRR treatment and year for bridges to be repaired. Bridges were selected for maintenance based on a priority determined by the socio-economic impact of maintenance activities, one of which is user delay costs. This only included the extra travel time due to speed restrictions along the working zone.

According to the literature research in [2] on bridge maintenance optimisation where limited resources have to be distributed strategically to optimise the performance of the whole system, life-cycle cost (LCC) analysis are most often used to select individual bridges for maintenance. While bridges are often connected in the same road network and should therefore be considered collectively, the authors found that only several studies have attempted to obtain the optimal bridge maintenance strategy by maximising the operational performance of a transportation network. Therefore, a bridge maintenance optimisation under budget constraints was proposed where the global objective was minimising total travel time, measured in total travel time for cars of all shortest paths between all possible Origin-Destination (OD) pairs. A binary particle swarm algorithm provided solutions to the mixed integer optimisation problem.

Network levels were also considered in the developed bridge maintenance plan optimisation for a multi-year planning horizon in [7], where the multi-objective optimisation determines which bridge components to repair, what MRR action to apply and when to perform it. The multi-objective optimisation used four objective functions to determine the optimal set of maintenance plans: maximisation of performance condition of the bridge, minimisation of agency and user costs, minimisation of duration of traffic disruption and minimisation of environmental impact. User costs only included travel delay costs, vehicle operating costs and accident costs of crossing the bridges themselves.

2.0.2. Other infrastructure maintenance optimisation

In the field of MRR optimisation, many papers have studied the optimisation of pavement maintenance. Just like with bridges, MRR optimisation aims to determine the optimal treatments for each pavement section at each time to improve the condition of the network within minimal budget [8]. Recent studies have increasingly focused on the application of multi-objective optimisation in pavement MRR [9]. In [10] a genetic-algorithm-based procedure for solving multi-objective pavement maintenance planning problems on network level has been developed. The multi-objective optimisation maximised work production and overall network pavement condition and minimised the total maintenance costs.

In the extensive literature review in [9] of research on multi-objective optimisation in pavement MRR planning, it is found that most research usually only focus on additional fuel consumption due to degrades in flatness when incorporating user costs, as in [11]. Delay costs due to MRR activities are frequently ignored. The authors of [9; 11] note that especially in urban road networks with heavy traffic flow, pavement MRR activities often cause delays and detours. Delay costs should thus be included but are complex to evaluate due to different alternative routes for each closed road. In [12; 13], delay costs have been included, but with the goal of calculating the additional emissions. In [13] micro-simulation modelling of traffic was used to estimate the emissions caused by delays during maintenance for several traffic management options, to extend the system boundary of the life cycle assessment of road pavement.

Authors of [14] have pointed out that pavement MRR planning models usually distribute selected sections of pavement for MRR spatially across the network. By combining adjacent sections with similar MRR needs in one single project, advantage can be taken of economies of scale. This is done by partitioning the network into smaller groups with similar MRR needs before using optimisation to allocate the MRR resources maximising system performance.

2.1. Genetic algorithm in maintenance optimisation

Optimisation methods can mainly be divided into two different categories: exact and heuristic approaches [15]. Exact optimisation approaches guarantee finding the optimal solution to a problem, but suffer from computational complexity [16]. Therefore, the use of meta-heuristic algorithms, such as genetic algorithms, is the main alternative to solve complex optimisation problems in a reasonable time for many real-life optimisation problems, such as network design problems [17].

The Genetic Algorithm (GA) is a heuristic algorithm often used to solve optimisation problems. It belongs to the larger class of evolutionary algorithms, inspired by evolution in nature where populations change over time by natural selection, reproduction and genetic variation. From a optimisation perspective, evolutionary algorithms can be seen as population-based randomised optimisation algorithms [18]. GA's have been widely used as a searched-based optimisation technique to develop accurate yet simple maintenance planning solutions constituting of multiple performance goals and budget constraints [6]. In one of the first studies using GA for the optimisation of bridge maintenance, the need for GA was emphasised [19]. One reason was that in bridge maintenance planning, selecting the optimal maintenance strategy is challenging due the number of possible combinations exponentially increasing with the number of bridges, the planning period and the number of maintenance alternatives. After this, many studies on optimising maintenance planning using GA followed [5; 6; 10].

2.2. Identified literature gaps

All MRR optimisation methods mentioned above are based on allocating limited resources over different maintenance projects to keep the network or structures functioning. Metaheuristics such as the genetic algorithm are often used to provide solutions that perform well on the optimisation objectives. Most methods focus on maximising the performance of the individual structures by selecting the best year and treatment for maintenance. Only several studies have taken the effect of MRR activities on road users into account when optimising maintenance. Often this only included the induced user cost on the bridges or pavement itself, neglecting user costs of route diversion and delays due to the maintenance. The importance of considering connected maintenance projects in complex urban road networks collectively due to high delay costs has been pointed out by several studies. However, existing studies have only relied on calculating the shortest paths for vehicles to estimate the additional travel time across the network caused by maintenance.

Based on the literature review, it is evident that current literature on maintenance optimisation does not effectively take into account hindrance to all road users caused by bridge maintenance. This is particularly crucial in complex urban road networks with high traffic demand, where bridge maintenance can result in significant delays and detours that are challenging to predict and evaluate. With many bridges to be renovated or renewed in such networks, it is important to assess the effect of simultaneous closures. Only calculating the shortest paths to assess the network performance does not take into account the delays due to increased traffic on diversion routes. Despite the promising search properties of GA's for reducing computational time, no studies have been found that combine traffic simulations with bridge maintenance optimisation, which could better assess and minimise the additional travel time experienced by all road users. This research aims to fill this gap by developing a bridge maintenance optimisation using simulations to minimise additional travel time experienced by all road users in the system.

3. Methodology

By combining optimisation methods and traffic simulations, this research aims to develop an effective bridge maintenance planning optimisation that balances the dual objectives of minimising hindrance to road users and maintenance costs. Given a predefined bridge set for maintenance, the proposed optimisation method selects for each bridge the starting point and execution duration of maintenance within a certain maintenance period. The proposed methodology for optimising bridge maintenance planning consists of three key components: 1) Traffic simulations using TNO's UST, 2) Optimisation via a genetic algorithm, 3) A meta model for the assessment of simultaneous bridge closures.

3.1. Traffic simulations

To accurately determine the traffic hindrance caused by bridge closures due to maintenance, traffic simulations of the UST of TNO are used. The Traffic+ model of this Digital Twin allocates car, freight, and bicycle traffic to the network using a static traffic assignment, given the OD matrices. The model used for this research retrieved the road network and OD matrices of Amsterdam from the more complex and computational heavy traffic simulation model of Amsterdam (VMA). The static assignment represents the average conditions for the specific period, in this case for the two-hour morning peak. All-Or-Nothing (AON) assignment for freight and bicycles are used, assuming everyone drives the shortest route. For car traffic a Volume Averaging (VA) method is used, distributing traffic over different routes using an equilibrium assignment. While VA considers congestion, AON does not. Using controls to change road input data, the closures of bridges are simulated. This results in new intensities and travel times for each road link in the system. By comparing this data to a base case scenario, the additional travel time of a scenario with closures can be computed.

Thanks to parallel programming, state-of-the art hardware and lightweight models, the UST can provide accurate simulations in under three minutes, significantly increasing the potential to evaluate many more solutions compared to traditional simulations like the VMA which can take hours.

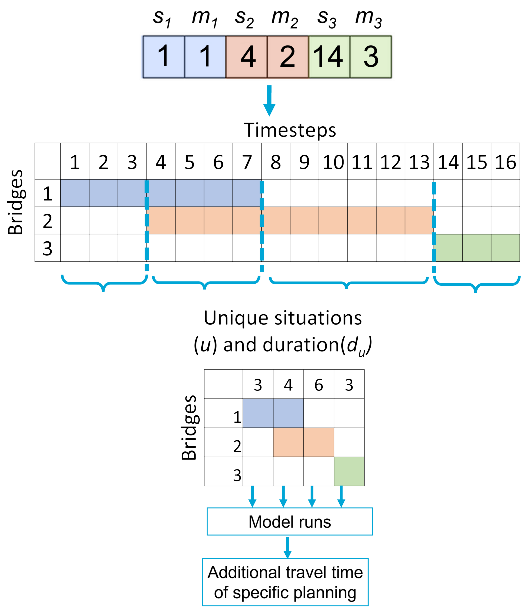


Figure 1: Overview of the evaluation of the additional travel time caused by a certain maintenance schedule solution using simulations

3.2. *Genetic algorithm*

Finding the optimal schedule by enumeration is impractical since the number of possible solutions increases exponentially with the number of bridges considered and increases even faster when more possible starting points and maintenance options are considered [19]. Therefore, this research uses a GA as a searched-based optimisation technique. In GA's coding of solutions is done in populations of chromosome-like strings containing the decision variables, which evolve by crossover and mutation, creating better solutions over multiple generations [18]. The solutions are evaluated using an objective function.

3.2.1. *Decision variables*

For every bridge (b) in the bridgeset (B) a starting point (s_b) and maintenance duration choice (m_b) need to be made, creating chromosomes for potential schedules consisting of two decision variables per bridge, see Figure 1. The possible starting points are determined by the number of time steps of the maintenance period considered in which all maintenance should take place(N_t). The possible maintenance duration choices in set M all have a corresponding maintenance cost (c_{m_b}).

3.2.2. *Parameters*

Parameters are input for the optimisation problem and used in the determination of the objective value. Table 1 provides an overview of all parameters that are input for the optimisation method.

3.2.3. *Evaluation*

The solutions produced by the GA are evaluated using an objective function. The overall objective is to minimise the total additional travel time caused by the bridge maintenance planning while also minimising the maintenance costs. A penalty will be included in the objective function to avoid maintenance plans exceeding the maintenance period. Figure 1 shows that a maintenance schedule consist of multiple unique closure situations, which all have to be assessed separately to determine the additional travel time of the whole schedule.

Symbol	Parameter
N_t	Number of timesteps in maintenance period
d_t	Number of days in timestep
d_{m_b}	Duration of maintenance m_b
c_{m_b}	Cost of maintenance m_b
p	Penalty per timestep overrun
w	Weight for hindrance component H

Table 1: Maintenance options within the optimisation algorithm with corresponding duration and costs

The objective function:

$$min(Z) = w \cdot H + C + P \tag{1}$$

It consists of three components: Hindrance to road users (H), maintenance cost (C) and the penalty for period overrrun (P). The weight w influences the balance between the H and C component and can be adjusted to reflect the priorities of policymakers.

Hindrance to road users H:

$$H = \sum_{u=1}^{U} TATT_u \tag{2}$$

$$TATT_u = ATT_u * \frac{100}{20} * d_t * d_u \tag{3}$$

$$ATT_u = TT_0 - TT_u \tag{4}$$

$$TT = \sum_i I_i \cdot t_i \tag{5}$$

The hindrance to road users is determined by the sum of the additional travel times caused by all unique situations u in the maintenance planning. The additional travel time of a unique situation $TATT_u$ is determined by extrapolating the additional travel time of the simulation for that unique situation ATT_u for the duration of the unique situation d_u. The simulated morning peak of two hours accounts for approximately 20% of the daily mobility [20]. TT_0 and TT_u are the total travel time in the simulation system for the base case and situation u determined by the intensities I_i and travel times t_i of all road links i.

Maintenance cost C:

$$C = \sum_{b=1}^{B} c_{m_b} \tag{6}$$

Penalty for overrun P:

$$P = max(0, e_{max} - N_t) \cdot p \tag{7}$$

$$e_{max} = \max_{b \in B}(s_b + d_{m_b}) \quad (8)$$

A penalty will be given very every timestep that the latest end timestep of maintenance e_{max} exceeds the number of timesteps of the maintenance period.

3.3. Assessment of simultaneous closures

The direct use of simulations in the evaluation of optimisation such as GA is computational difficult due to the large number of simulations and long simulations times [18]. This is true even for the relatively fast simulations of the UST that take 3 minutes to evaluate one unique situation. The number of unique situations in a maintenance schedule that all need to be evaluated separately, increases exponentially with the number of bridges, making it impractical to use direct simulations for all situation evaluation ATT_u of solutions. To overcome this computational problem, population-based algorithms are often coupled with approximation models, also called meta-models [18]. This has also been applied in a previous study on quay wall maintenance using the UST [4] to determine the effect of maintenance on traffic. In their simple meta-model, simulations of individual closures were used to compute the effect of simultaneous closures by summing the additional travel times. This disregards the potential dependencies between closures. Evaluating simultaneous closures as if they are isolated from each other can lead to wrong conclusions about the optimal configurations in maintenance schedules, potentially increasing hindrance to road users.

The proposed method in this research uses the interdependencies between bridge pairs to improve the estimation of the network effects of combined closures. The interdependency value of a bridge pair is determined by the change in travel time on a bridge after a closure of another. A negative percentage change would mean that closing the other bridge too would lead to less additional travel time than closing them separately. A positive percentage change would mean that closing the other bridge too would lead to more additional travel time than closing them separately. The larger the absolute percentage change, or interdependency value, the more important it is to take into account the relation between these bridges when calculating the additional travel time of the combined closure. Figure 2 shows the proposed estimation method using interdependencies to evaluate the additional travel time of a unique situation ATT_u. It evaluates a set of simultaneous closed bridges by pairing the bridges in order of the highest absolute interdependency value, using the simulation of the combined closure for each pair. This requires only all individual closures and combinations of two to be computed in advance, which can be done within a few days for up to 50 bridges.

4. Application

In this section the applicability of the proposed bridge planning optimisation using interdependencies to evaluate the effect of simultaneous closures is demonstrated with a case study of four bridges in the city centre of Amsterdam.

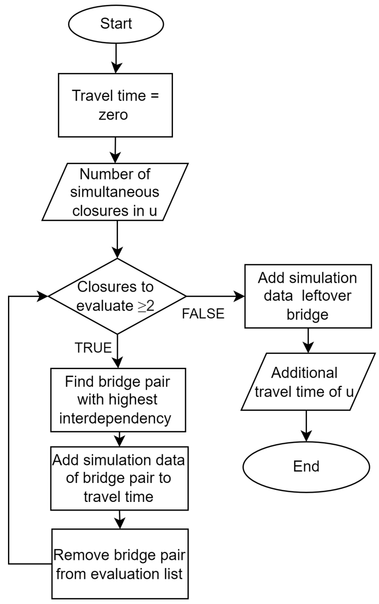


Figure 2: Process of determining the additional travel time of a unique traffic situation ATT_u in a bridge maintenance schedule using the interdependency method

4.1. Case description

To show the benefits of the proposed optimisation method using interdepedencies to evaluate the effect of simultaneous closures, this approach is tested on an illustrative case study of four bridges in Amsterdam. With a high traffic demand in a complex urban road network, this city is well suited to test the new evaluation method of simultaneous closures. The results of the optimisation using the interdependency estimation method are compared to the evaluation method used in the previous maintenance planning optimisation of [4] using the traffic simulation of UST. The latter does not take into account network effects when assessing multiple simultaneous closures and is therefore referred to as the isolation method. The four selected bridges for his case study are the Rijckerbrug, Nieuwe-Wercksbrug, Oude Kinkerbrug and Berlagebrug, of which the locations can be seen in Figure 3. With three out of four bridges being close together, the optimisation method can be tested on taking into account the network effects.

For the maintenance planning, a period of two years is considered with time steps of one month. This means that the maintenance period in which the four bridges should be scheduled consists of 24 time steps. During maintenance a complete closure of the bridge is assumed and for all four bridges, the same maintenance duration and corresponding costs are considered. The three used duration options and costs in Table 2 show that a faster execution of construction work results in higher maintenance costs.

To assess the effect of single and simultaneous closures in the evaluated maintenance schedules, all individual and combinations of two bridge closures have been simulated using the static traffic simulation of UST. Based on the road data of the single closure simulations, the interdependencies between all bridge

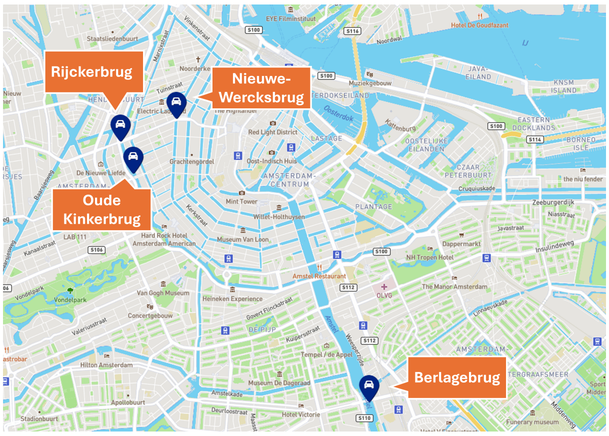


Figure 3: Location of the 4 bridges: Rijckerbrug, Nieuwe-Wercksbrug, Oude Kinkerbrug and Berlagebrug

Maintenance	Duration	Costs (in euro)
Normal	6 months	8 million
Twice as fast	3 months	8.8 million
Three times as fast	2 months	10.4 million

Table 2: Maintenance options within the optimisation algorithm with corresponding duration and costs

pairs are computed, which can be seen in 3. The additional travel times for the modes car, freight and bicycle retrieved from the simulations and the determined interdependencies are used in the assessment of traffic hindrance in the objective function of the GA.

The objective value used to evaluate solutions consists of three parts, as has been shown in the Methodology. To balance the hindrance to road users with the maintenance cost, a weight for hindrance of 10 is selected. While this weight can be adjusted to reflect the priorities of policymakers, this value has been chosen for this case study to have both the hindrance as cost at least on an equal footing in the optimisation process. The weight of 10 is based on the value of travel time for car and cycling in the Netherlands of €10.42 and €10.39 respectively to have both the hindrance to road users as maintenance cost evaluated in euro's [21]. The third part of the objective value acts as a constraint for maintenance period overrun. Therefore, a large penalty value of 100,000 has been selected to make sure the optimisation avoids exceeding the given maintenance period of 24 months.

For the optimisation, the GA is implemented and solved using Python, specifically with the *pymoo* library version 0.5.0 [22]. A population size of 4 is selected, which allows for faster evaluation, while also providing enough diversity. The initial population is randomly generated within the boundaries of the number of time steps and maintenance duration options. The GA is terminated after 100 generations, which corresponds to 400 solution evaluations. Every generation new solutions are generated using a simulated binary crossover and polynomial mutation for discrete variables.

	Rijckerbrug	Oude Kinkerbrug	Berlagebrug
Nieuwe-Wercksbrug	-45.138 %	11.703 %	-0.001 %
Rijckerbrug	-	31.485 %	0.003 %
Oude Kinkerbrug	-	-	0.015 %

Table 3: The interdependencies between bridge pairs captured by the average percentage change in total travel time after closure of one bridge of each pair

4.2. Results

This subsection shows the results of the application of the methodology on the case study of Amsterdam. To assess the improvement of the proposed optimisation using the interdependency method, the interdependency method will be compared to the isolation method.

4.2.1. Convergence

In Figure 4 the convergence behaviour can be seen of the objective value for both the isolation and interdependency optimisation methods across different runs, using 20 different seeds. A seed determines the sequence of random numbers generated by the random number generator, which in turn affects various stochastic processes of the genetic algorithm, such as the initialisation of the population, the selection of individuals for crossover, and the mutation process. To assess the general performance of both optimisation methods and avoid the influence of any specific random sequence, it is important to perform multiple runs with different seeds.

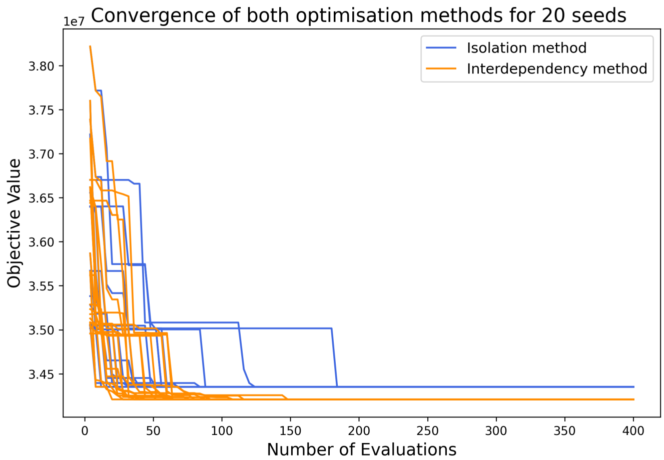


Figure 4: Convergence of the objective value for the two different estimation methods for 20 seeds

The convergence graph shows that across all runs, both methods have reached a stable objective value after 200 evaluations, which suggests that running the genetic algorithm for 100 generations with this population size is sufficient to find an optimal solution. Table 4 shows the optimal objective value and separate scores of the three individual components within the objective value for both methods. The small case study of four bridges allows for checking whether the GA has found the global optimum, by evaluating all possible solutions. With 24 possible starting points, 3 maintenance options and 4 bridges

	Isolation	Interdependency
Additional travel time (h)	155231.21	140916.56
Maintenance cost (€)	32800000	32800000
Penalty for overrun	0	0
Total objective value	34352312.17	34209165.61

Table 4: Results of all optimisation runs for both methods

the total number of possible solutions is $(24 \cdot 3)^4$, which is 26,873,856. The lowest objective value using the interdependency method found among all possible solutions is the same as the objective value of the optimal solution found by the GA, thus the GA has retrieved the global optimum. It should be noted that the time needed for this check increases exponentially with the number of bridges and therefore is only possible within reasonable time for five bridges or less. Evaluating all possible solutions for a similar case with six bridges would already take 22 years[1].

4.2.2. Analysing the optimal maintenance schedules for the isolation and interdependency method

For both methods, the optimal solutions generated with the GA have the same objective value and separate components scores for all 20 different seeds. Looking at these scores in Table 4, three things can be noted: 1) No penalty was given meaning that in all schedules maintenance is executed within the 24 months maintenance period. 2) The maintenance costs for the optimal schedules of both methods are the same, since in all optimal results for every bridge the same duration is chosen. 3) The additional travel time score of the interdependency method is lower than the additional travel time score of the isolation method.

All 20 seeds for both methods lead to the same objective value, however, the configuration within the schedules for these seeds can differ. For four selected seeds, the schedules have been visualised. Figure 5 shows the schedules for the isolation method where simultaneous bridge closures are assessed as if the bridges are isolated from each other. As can be seen, this results in a randomness in distribution of maintenance plans over time, since the additional travel time will always be the sum of all four individual closures. This randomness can coincidentally lead to good or bad combinations of bridge closures. For bridges with negative interdependency values, the travel times on one of the bridges after closing the other will decrease, resulting in lower additional travel times when simultaneously closed. This is the case for the Rijckerbrug and Nieuwe-Wercksbrug simultaneous closed for 2 months in the optimal schedule for seed 18. Positive interdependency values on the other hand will indicate that the combination of the bridges is worse for the additional travel time than their separate closures, as is the case for the Oude Kinkerbrug and Rijckerbrug for seed 5.

[1]For these evaluations, a computer with the following specifications was used: Intel Core i5-6200U CPU @ 2.30GHz 2.40 GHz and installed memory of 8 GB (RAM)

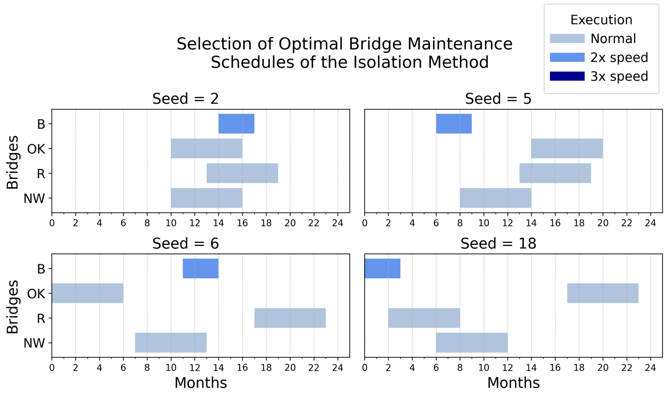


Figure 5: Optimal bridge maintenance schedules of the isolation method for 4 selected seeds (NW = Nieuwe-Werckbrug, R = Rijckerbrug, OK = Oude Kinkerbrug, B = Berlagebrug)

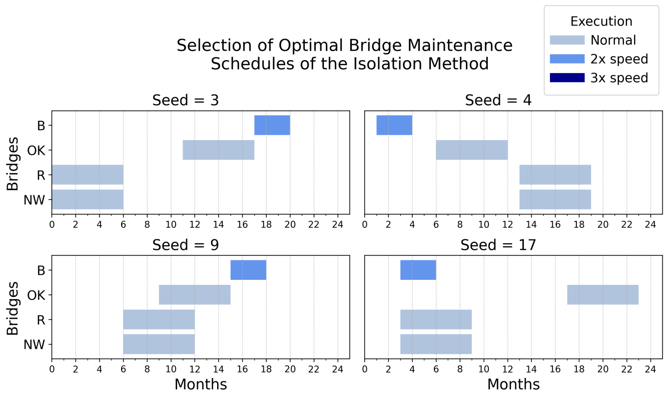


Figure 6: Optimal bridge maintenance schedules of the interdependency method for 4 selected seeds (NW = Nieuwe-Werckbrug, R = Rijckerbrug, OK = Oude Kinkerbrug, B = Berlagebrug)

For the four schedules of the interdependency method shown in Figure 6, it can be seen that the maintenance plans of the Rijckerbrug and Nieuwe-Wercksbrug are always aligned. Those two bridges have the highest absolute interdependency value and are thus always evaluated with their combined simulation, which result in lower additional travel times, see Table 3. However, there is still some randomness in the combination of the pair consisting of the Rijckerbrug and Nieuwe-Wercksbrug and the other two bridges. This can be explained by the evaluation of the interdependency method, which assess simultaneous bridge closures in groups of maximal two bridges. Multiple groups are assessed as if isolated from each other. The schedule for the seed of 9 shows that maintenance plans of the bridge pair with the highest interdependency partly overlaps with the maintenance plan of the Oude Kinkerbrug. This combination should be avoided since both the Nieuwe-Wercksbrug and the Rijckerbrug have a positive interdependency value with the Oude Kinkerbrug, meaning their simultaneous closure would lead to higher additional travel times.

In all optimal solutions of both methods, the maintenance of the Berlagebrug has an execution of two times faster than normal, while the other bridges will be in maintenance for the nor-

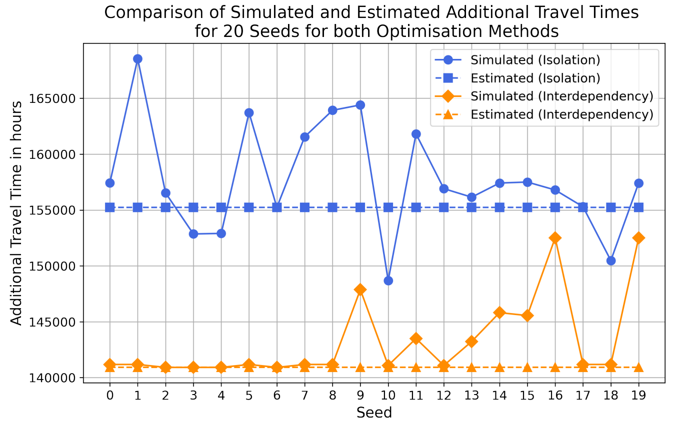


Figure 7: Comparison of the simulated and estimated additional travel times for 20 seeds for both optimisation methods

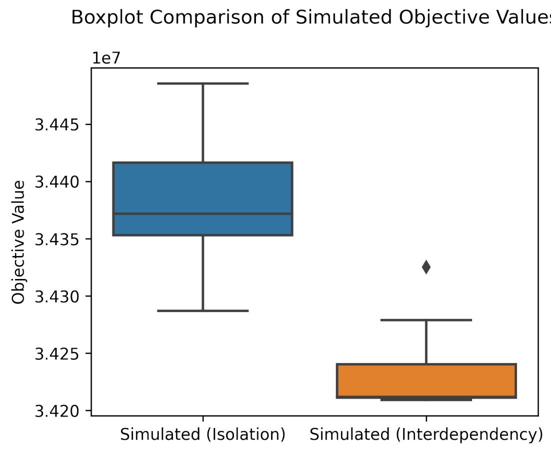


Figure 8: Boxplot of the simulated objective values of the optimal solutions of both methods

mal execution duration of 2 months. This bridge has a larger effect on additional travel time when closed compared to the others. With a weight of 10 for the hindrance component, increasing the execution of the maintenance of the Berlagebrug will result in a better overall solution. The sensitivity of this weight has been investigated by analysing the optimal schedules of both optimisation methods for a consistent seed using different weights. The higher the additional travel times after closure, the lower the weight at which the transition to faster maintenance is made. The analysis also showed that taking interdependencies between bridges into account can avoid using unnecessary expensive and intensive maintenance execution by effectively combining the closures.

4.3. Quality of the optimisation methods

The difference between the optimisation using the interdependency and the optimisation method using the isolation method is in their estimation of additional travel time for simultaneous closures. To evaluate the quality of the estimation methods, first the accuracy of the additional travel time estimation of both methods has been tested by comparing the estimations of the schedules to the fully simulated additional travel time using the UST. Figure 7 shows the UST simulated additional travel time of the optimal maintenance schedules for both methods for 20 seeds compared to the estimations of both methods.

For the isolation method can be seen that the randomness of configurations in the schedule can indeed lead to both better and worse configurations in terms of additional travel time, as the simulated additional travel time of the schedules fluctuates around the estimated additional travel time by the isolation method. Underestimation occurs when there are bad combinations in the schedule, like explained earlier for a seed of 5. Overestimation on the other hand occurs when good combinations are formed in the schedule, like explained for a seed of 18. For the interdependency method, the estimation of additional travel time is always the same or lower than that of the simulated additional travel time. Only when more than 2 bridges closures are combined in the maintenance schedule, the estimation deviates from the full simulation. There is still some randomness when more than 2 bridges are combined, due to assessment in groups of two. Since both the Berlagebrug and the Oude Kinkerbrug have a positive interdependency value with the bridge pair Rijckerbrug and Nieuwe-Wercksbrug, only underestimations due to randomness are possible.

The results also show that low simultaneous closures with low interdependency values between bridges can be assessed as if isolated from each other. The schedules of the isolation method for a seed of 6 and the interdependency method for a seed of 17 both have the Berlagebrug closed together with other bridges. Table 3 showed that the Berlagerug has a low interdependency value with all other bridges. In both methods, the closure effect of the Berlagebrug is assessed as if isolated. As can be seen in Figure 7, the estimations of these seeds are almost similar to the simulated additional travel time.

In terms of additional travel time estimation, the schedules of the interdependency method outperform those of the isolation method. Since the penalty and maintenance cost of all solutions of both methods are the same, the difference in the additional travel time estimation determines the overall difference in quality of the solutions. The quality of both optimisation methods is compared by looking at the simulated objective values in Figure 8. Based on these boxplots, it can be concluded that the optimisation using the interdependency method consistently produces lower objective values and thus is the better optimisation method for creating good maintenance schedules based on the objective function in this research. The isolation method leads to higher and more variable results, making it less effective for the planning of bridge maintenance to reduce hindrance to road users in terms of additional travel time.

5. Conclusions

Infrastructure maintenance has been underfunded for a long time, leading to many structures in the need of repair. The numerous upcoming maintenance projects are expected to cause severe hindrance to road users, especially for bridge maintenance in dense urban road networks like Amsterdam. The complex road network and dependencies between bridges make it difficult to foresee the effect of simultaneous closures on traffic, complicating the efficient planning of bridge maintenance in order to reduce hindrance to road users. Current practices do not consider these interdependencies in maintenance planning, which could lead to more additional travel time for road users. Traffic simulations are a good way of assessing the effect of simultaneous closures, however, evaluating many different maintenance schedules can be time consuming.

To improve the assessment of hindrance to road users in maintenance planning, this research has proposed an optimisation combined with traffic simulation using an improved estimation of simultaneous closures. The developed genetic algorithm selects a starting point and execution duration for all bridges selected for maintenance, minimising both maintenance cost and hindrance to road users in terms of additional travel time for car, freight and bicycle. Using the fast Urban Strategy Tool of TNO, static traffic simulations are used to estimate the effect of simultaneous closures. The improved estimation method uses interdependencies between bridge pairs, determined by the effect on traffic on one bridge after closure of another. That way, the worst combination possible is always avoided and the best possible combination is taken advantage of. The balance between the minimising hindrance and minimising cost is determined by a weight in the objective value, which can be changed according to the decision-makers views. The proposed method is illustrated on a case study of four bridges in the city centre of Amsterdam. The developed interdependency estimation method is compared to the previous isolation method, which does not take network effects into account.

The key findings are summarised as follows: 1) The interdependency estimation method outperforms the isolation method with more accurate additional travel time estimations, resulting in overall better solutions produced by the optimisation. 2) Not all bad combinations can be avoided with the interdependency method. Interdependencies between bridges of different pairs in one simultaneous closure are neglected, since the simultaneous closures are assessed in pairs. 3) Using the interdependency estimation method instead of traffic simulations reduces the computational time of the optimisation while providing accurate estimations of additional travel time, making the optimisation applicable for assisting decision makers in weighing the two goals of minimising hindrance and maintenance costs.

Based on these findings, the proposed fast optimisation method showed to be very useful for planning maintenance activities within dense urban networks, offering a quick and more accurate way to test numerous solutions under different policy strategies, compared to previous methods.

6. Recommendations

The proposed optimisation for bridge maintenance planning improved the assessment of the effect of simultaneous closures, providing good solutions to reduce hindrance to road users. However, there are still some limitations of this study. Therefore, recommendations are made to improve the model in three categories: 1) Improving the additional travel time assessment, 2) expanding the maintenance options considered and 3) improving the evaluation of simultaneous closures.

1) The assessment of additional travel time caused by the planning could be improved by expanding the included modes to public transport and pedestrians. Moreover, by splitting the additional travel time for all modes and including mode-specific weights, more insight can be given in the effect of closures on additional travel time with different policy priorities. In this research static traffic simulations of the morning peak are used. Different simulation periods or other traffic simulations like dynamic traffic models or agent-based models can be used to improve the assessment of additional travel time.

2) The considered maintenance options could by expended by considering bridge specific duration options or different types of closures to improve the bridge maintenance schedules of the optimisation. In this research, only complete closures are considered in the optimisation. The effectiveness and accuracy of the interdependency estimation method after expanding the closure options should be further investigated, since the method is based on complete closure effects.

3) To improve the evaluation of simultaneous closures, other factors can be considered to determine the interdependency value, like the commonality in OD-pairs using the bridges. Further research is also needed to avoid wrong combinations between multiple pairs in simultaneous closures. Testing the proposed optimisation on a larger bridge dataset will explore the accuracy and potential of the estimation further.

In addition to model improvements, several other future research options are suggested to reduce hindrance caused by bridge maintenance. Not only additional travel times play a role in hindrance perception [23]. Therefore, the impact of other factors like the experience of the detour, safety of temporary traffic situations, communication, noise hindrance and repetition of obstructions in time should be further investigated. Other future research could focus on the distribution of hindrance over different groups of road users to improve equity, using for example demographic and social factors or mode distinction. These research directions could contribute to a further reduction of the overall hindrance experienced during maintenance.

Acknowledgements

This paper was written as part of the final thesis to obtain the master's degree of Transport, Infrastructure and Logistics of Delft University of Technology. The thesis was commissioned by the municipality of Amsterdam, specifically the programme Bridges and Quay walls, and in collaboration with TNO. A special thanks to the chair Serge Hoogendoorn and supervisors Gonçalo Correia, Maaike Snelder, Niek Mouter, Laura Hakvoort, Lara Borst, and Marjolein van der Meer of the thesis committee for their critical eye and help during this research.

References

[1] A. Bleijenberg, "Instandhouding civiele infrastructuur proeve van landelijk prognoserapport vervanging en renovatie," tech. rep., TNO, Apr. 2021.

[2] W. Zhang and N. Wang, "Bridge network maintenance prioritization under budget constraint," *Structural Safety*, vol. 67, pp. 96–104, 7 2017.

[3] Programma Bruggen en Kademuren, "Gefundeerd herstellen: Actieplan bruggen en kademuren 2023-2026," tech. rep., Gemeente Amsterdam, Dec. 2022.

[4] F. Swaalf, S. Spruijtenburg, A. M. Colán), R. Shina, H. Weijs, and P. de Heer, "Background literature and decisions for the developed planning-maintenance module for quay walls in amsterdam." Unpublished work, 2023.

[5] M. Alsharqawi, . Saleh, A. Dabous, T. Zayed, F. Asce, and S. Hamdan, "Budget Optimization of Concrete Bridge Decks under Performance-Based Contract Settings," *Journal of Construction Engineering and Management*, vol. 147, 6 2021.

[6] Z. A. Bukhsh, I. Stipanovic, and A. Dorée, "Multi-year maintenance planning framework using multi-attribute utility theory and genetic algorithms," *European Transport Research Review*, vol. 12, 1 2020.

[7] E. M. Abdelkader, O. Moselhi, M. Marzouk, and T. Zayed, "An exponential chaotic differential evolution algorithm for optimizing bridge maintenance plans," *Automation in Construction*, vol. 134, 2 2022.

[8] A. Fani, H. Naseri, A. Golroo, S. A. Mirhassani, and A. H. Gandomi, "A progressive hedging approach for large-scale pavement maintenance scheduling under uncertainty," *International Journal of Pavement Engineering*, vol. 23, no. 7, pp. 2460–2472, 2022.

[9] W. Chen and M. Zheng, "Multi-objective optimization for pavement maintenance and rehabilitation decision-making: A critical review and future directions," 10 2021.

[10] T. Fwa, W. Chan, and K. Hoque, "Multiobjective optimization for pavement maintenance programming," *Journal of Transportation Engineering*, vol. 126, no. 5, pp. 367–374, 2000.

[11] S. Meneses and A. Ferreira, "Pavement maintenance programming considering two objectives: Maintenance costs and user costs," *International Journal of Pavement Engineering*, vol. 14, no. 2, pp. 206–221, 2013.

[12] J. h. Choi, "Strategy for reducing carbon dioxide emissions from maintenance and rehabilitation of highway pavement," *Journal of Cleaner Production*, vol. 209, pp. 88–100, 2 2019.

[13] F. Galatioto, Y. Huang, T. Parry, R. Bird, and M. Bell, "Traffic modelling in system boundary expansion of road pavement life cycle assessment," *Transportation Research Part D: Transport and Environment*, vol. 36, pp. 65–75, 2015.

[14] L. Gao and Z. Zhang, "Management of pavement maintenance, rehabilitation, and reconstruction through network partition," *Transportation Research Record*, no. 2366, pp. 59–63, 2013.

[15] F. Rothlauf, *Optimization Methods*, pp. 45–102. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[16] TU Delft, "Mude textbook," 2024.

[17] G. E. Cantarella and A. Vitetta, "The multi-criteria road network design problem in an urban area," *Transportation*, vol. 33, pp. 567–588, 11 2006.

[18] S. Meyer-Nieberg, N. Leopold, and T. Uhlig, "Introduction to Simulation-Based Optimization," in *Natural Computing for Simulation-Based Optimization and Beyond*, pp. 1–8, Cham: Springer International Publishing, 2020.

[19] C. Liu, A. Hammad, and Y. Itoh, "Maintenance Strategy Optimization of Bridge Decks Using Genetic Algorithm," *Journal of Transportation Engineering*, vol. 123, pp. 91–100, 3 1997.

[20] M. Cloïn, "Met het oog op de tijd," 11 2013.

[21] M. Kouwenhoven, J. Muller, S. Thoen, J. Willigers, and G. d. Jong, "Values of Time, Reliability and Comfort in the Netherlands 2022," tech. rep., KiM Netherlands Institute for Transport Policy Analysis, 11 2023.

[22] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[23] M. Damen, S. Oostlander, W. v. Deursen, and F. Sijbrandi, "Een zure appel op z'n tijd," tech. rep., 2020.