

Multi-objective project portfolio optimization

Application in railway infrastructure networks

S.L. van Ginkel

Multi-objective project portfolio optimization

Application in railway
infrastructure networks

by

S.L. van Ginkel

to obtain the degree of Master of Science in Systems and Control
at the Delft University of Technology,
to be defended publicly on Monday November 27, 2017 at 09:30 AM.

Student number:	4076885	
Project duration:	February 2017 – November 2017	
Thesis committee:	Prof. dr. ir. B. De Schutter	TU Delft, chair
	Dr.ir. A. A. Núñez Vicencio	TU Delft, supervisor
	D. de Poot, MSc.	ORTEC Consulting, supervisor
	Z. Su, PhD(c)	TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



Abstract

Investment projects in railway infrastructure networks are needed to maintain a high condition of the network, maximize the capacity of the network, and to keep the risk low. Railway organizations are aiming to achieve these goals, but also want to minimize the costs of the investments. The desires of stakeholders in railway networks such as the government, passengers, and freight users are often not aligned. Decision makers in this field need to make trade-offs between the conflicting objectives. In the decision-making process regarding investment projects, it is important that the multiple objectives are considered explicitly. A multi-objective optimization problem formulation is proposed for the investment project portfolio optimization problem. Solving a multi-objective optimization problem means to identify the set of Pareto optimal solutions. These Pareto optimal solutions are used by decision makers to gain insight in the effects on the objectives, and to make well-founded trade-offs.

An approach is proposed to include the uncertain nature of investment projects in the problem formulation, without increasing the complexity of the problem too much. This is done by introducing lower and upper bounds for uncertain variables. By optimizing not only the expected values of the objective functions, but also the lower and upper bound, one is able to identify the solutions that give satisfactory results for all the scenarios. Project portfolios that are Pareto optimal for the lower bound, expected value, and upper bound, are considered as robust choices with respect to the uncertainties in the objective values.

Railway infrastructures typically consist of many assets with a long lifespan. The prediction horizon and the level of detail considered in the optimization problem need to be chosen carefully to avoid extremely large computation times. Even for a modest level of detail, the problem size is too large to check all the possible solutions within acceptable time. The computation time is also dependent on the algorithm that is used to search for the Pareto optimal set. Next to the computation time, several performance indicators are selected to measure the quality of the approximation of the Pareto optimal set that is provided by an algorithm.

Two algorithms with different approaches are proposed to approximate the Pareto optimal set, the repeated ε -constraint algorithm and the widely used genetic algorithm NSGA II. Some benchmark problems are introduced to gain insight into the working of the proposed algorithms. A case study is performed for an artificial data set for investment projects in a railway network. The selected performance indicators are used to analyze the results of the case studies for the proposed algorithms with different settings. The repeated ε -constraint algorithm is able to find a close representation of the Pareto optimal set, and has a very stable performance due to the established search method of the CPLEX solver. However, obtaining these results requires an excessive amount of computation time. The genetic algorithm NSGA II is much faster, and for carefully tuned settings, the quality of the approximation set is fine. The coverage of the NSGA II is considered sufficient, so due to the reduced computation time, the NSGA II is recommended to use in the decision support process of selecting investment project portfolios in railway infrastructure networks.

Cover photo by Cameron Kirby.



Acknowledgements

This thesis is the final part of the Systems & Control Master at Delft University of Technology. After my internship at ORTEC Consulting, they offered me to stay for a very interesting graduation project. I was very happy to accept this, and therefore the research is conducted in cooperation with ORTEC Consulting. I want to thank Daphne for all her support, useful feedback from a business perspective, and her belief that my research can contribute to a decision support tool in reality. I want to thank all my colleagues at ORTEC for their interesting stories and relaxing lunch breaks.

Alfredo was my incredibly dedicated supervisor from the TU Delft. I could always expect detailed feedback, and interesting discussions. His positive attitude kept me motivated until the end, and I am very grateful for that! I would like to thank Bart for his interest in my project and his critical but useful notes. I would also like to thank Zhou for being in my graduation committee, and for taking the time to read my thesis.

Next, I would like to thank Marcel from Znapz, for all his efforts to provide me enough information from the rail industry, and the opportunity to join the UIC conference in Paris.

Furthermore, I have some amazing friends who make my life fun and interesting, and without whom I would not have been able to finish my studies. Seth, I have no idea how I can thank you enough for every moment that I needed you and you were there. Maybe I can make up for all my stress with two months of relaxing in South America? Robert, I feel very lucky to have you as my friend and study mate, and I want to thank you for being loyal, helpful, and honest. Thank you all for the welcome distractions, JC Smaak for some great years and the awesome trip to Cambodia, Shanna for all our adventures and hilarious moments, Scheep 20 for truly being the best roommates in party times and in fodmap times, Wocky for all fun and games, and last but not least StuD 16 for giving me shining examples of being ambitious and fun!

To Kasper, you made me realize how important my family is, and I hope that you realize how much I care about you. Last but not least, I want to thank my parents for always supporting me and being proud of me.

Delft, University of Technology
November 2017

S.L. van Ginkel

Contents

Abstract	iii
Acknowledgements	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Problem statement.	3
1.3 Research questions.	3
1.4 Thesis outline.	4
2 Literature overview	5
2.1 Investment project portfolio optimization	5
2.1.1 Objectives.	6
2.1.2 Formulation.	7
2.1.3 Level of detail	7
2.1.4 Uncertainty	9
2.1.5 Dependencies.	9
2.2 Multi-objective optimization methods	10
2.2.1 Multi-objective optimization	11
2.2.2 Repeated ε -constraint algorithm	13
2.2.3 Multi-objective genetic algorithms	13
2.2.4 Uncertainty in multi-objective optimization	15
2.2.5 Performance indicators	16
2.3 Multi-objective project portfolio optimization.	16
2.3.1 Multi-objective optimization methods in project portfolio optimization	16
2.3.2 Visualization and decision support	17
2.4 Summary	18
3 Investment project portfolio optimization method	21
3.1 Optimization problem formulation.	21
3.2 Railway network representation.	25
3.3 Multi-objective optimization methods	32
3.4 Performance indicators	35
3.5 Visualization and decision support	36
3.6 Summary	37
4 Benchmarks & railway network case studies	39
4.1 Benchmark problem 1	39
4.1.1 Repeated ε -constraint algorithm	40
4.1.2 Multi-objective genetic algorithm.	41
4.1.3 Comparison.	42

4.2	Benchmark problem 2	42
4.3	Case studies.	43
4.3.1	Repeated ε -constraint algorithm	45
4.3.2	Multi-objective genetic algorithms	49
4.3.3	Comparison.	53
4.4	Decision support	54
4.5	Summary	54
5	Conclusions and recommendations	55
5.1	Conclusions.	55
5.1.1	Problem formulation.	55
5.1.2	Multi-objective optimization methods	56
5.1.3	Visualization and decision support	56
5.1.4	Research questions	57
5.2	Discussion and further research	58
5.3	Recommendations for ORTEC	60
A	Decision support tool	63
A.1	Insight into the current situation	63
A.2	Creating insight into different scenarios	64
	Bibliography	67

List of Figures

2.1	Different types of dependencies between projects, adapted from Teng et al. (1996), where $\alpha_{a,a'}$ is shown in the dotted and striped areas.	10
2.2	Pareto optimality, adapted from De Schutter et al. (2015).	11
2.3	Visualization of the repeated ε -constraint algorithm for minimizing J_1 and J_2 . The general objective is J_1 , and J_2 is bounded by the ε -bounds.	14
2.4	Visualization methods, as presented in Tušar et al. (2015).	18
3.1	Visualization of how the next parent population is obtained from the combined parent and offspring population, adapted from Deb et al. (2002).	35
4.1	Pareto sets $\mathcal{P}$ and corresponding Pareto optimal fronts for benchmark problem 1.	40
4.2	Comparison of the repeated ε -constraint algorithm and NSGAII for benchmark problem 1.	41
4.3	Pareto sets $\mathcal{P}$ and corresponding Pareto optimal fronts for benchmark problem 2.	42
4.4	Part of the Dutch railway network that is used for the case study.	43
4.5	Scatter plot matrix for the non-dominated reference set $\mathcal{R}$ of the railway network case studies.	44
4.6	Parallel coordinates plot for the non-dominated reference set $\mathcal{R}$ of the railway network case studies.	45
4.7	Solution sets for the repeated ε -constraint algorithm with $\tau = 0\%$, for $g = 3$, $g = 4$, and $g = 5$	47
4.8	Non-dominated solution sets for the repeated ε -constraint algorithm.	48
4.9	Solution sets for the repeated ε -constraint algorithm with 5 constraints per objective, $g = 5$	48
4.10	Non-dominated solution sets for the repeated ε -constraint algorithm with $g = 5$ and different shades of blue for different values of τ	49
4.11	Solution sets for the NSGA II with $ S = 100$ and $\text{gen} = 100$	52
4.12	Solution sets for the NSGA II with $\text{gen} = 100$, and $p_m = 0.01$	52
4.13	Solution sets for the NSGA II with $ S = 200$, and $p_m = 0.02$	53
A.1	Data import page from the decision support tool demo in AIMMS.	63
A.2	Network visualization page from the decision support tool demo in AIMMS.	64
A.3	Projects overview page from the decision support tool demo in AIMMS.	64
A.4	Scenarios overview page from the decision support tool demo in AIMMS.	65
A.5	Optimization settings page from the decision support tool demo in AIMMS.	65
A.6	Results in pairwise comparison plots, included on the results overview page from the decision support tool demo in AIMMS.	65
A.7	Planning of projects of a selected project portfolio, included on the detailed results page from the decision support tool demo in AIMMS.	66
A.8	The core index listed for the projects, included on the decision support page from the decision support tool demo in AIMMS.	66
A.9	Comparison of the planning and objective values for two project portfolios, included on one of the detailed results pages from the decision support tool demo in AIMMS.	66

List of Tables

2.1	Overview of articles on the optimization of maintenance and investment projects planning.	19
4.1	Performance indicators for different settings of the repeated ε -constraint algorithm, for the convex benchmark problem.	41
4.2	Performance indicators for different settings of the NSGA II, for the convex benchmark problem.	41
4.3	Settings and results of the repeated ε -constraint algorithm in the case studies.	46
4.4	Settings and performance indices for the NSGA II in the case studies.	51
4.5	Comparison of performance indices for the repeated ε -constraint algorithm and the NSGA II.	53

Introduction

1.1. Motivation

Railway organizations are under an increasing pressure to improve the performance, safety, and capacity of the railway infrastructure, while budgets are tightened. This is for example due to a continuous competition from other modes of transport and tenders between railway companies. Next to the competition, railway organizations have to deal with high user and governmental expectations regarding the performance and capacity of railway networks. In Europe, this governmental pressure is a result of the European rail transport objectives (UIC, 2017), that are set by the European Union (European-Union, 2017). In these long term goals it is stated that in 30 years, more than half of the medium-distance passenger transport should be via rail, and more than half of the freight transport should be carried out by rail or waterborne transport. In order to achieve these goals, railway organizations are forced to improve their decision making strategy. It is of vital importance to select the investment projects that give the best possible results for the long term performance of the entire network. However, decision makers in this field are faced with a lot of information, and it is not evident what the best investment projects are.

Investment projects in the field of railway infrastructure management are often maintenance and renewal projects, but could also concern for example network expansions, the introduction of a new signalling system, or building new stations. Typically each year, the infrastructure managers are selecting the projects to be executed. This selection of projects, also called the project portfolio, is currently often based on the opinion and experiences of experts, or based on the age of the concerning assets. The assets of the railway infrastructure are deteriorating over time, so to keep the performance and the condition of the railway network at an acceptable level, it is crucial to execute maintenance and renewal projects. Both the assets and the maintenance in the railway infrastructure are very expensive (the maintenance expenditures of ProRail for the Dutch railway network in 2016 were €523 million), and the maintenance budget of railway organizations is often limited. Therefore, not all proposed improvement projects can be executed. To make the right decisions in such project portfolio selection problems, it is crucial to have insight into the problem. In the case of railway infrastructure management, this means insight into both the assets and the proposed projects. To gain the required insight into the assets of the infrastructure, there is a transition in railway organizations from unorganized and decentralized data sheets, to real asset management systems. The ISO 55.000 standard (ISO, 2014) specifies the requirements and applications for asset management systems. Railway organizations across the world are making steps towards such a system. Railway infrastructure managers from different countries are cooperating in the *Asset Management Working Group* of UIC, which has been working on issues like a common interpretation of asset management and sharing good practices. This shows that the railway organizations realize that they should improve the process of project selection.

To improve the process of project selection, not only insight into the assets is needed, but also insight into the proposed projects. It should be clear what the expected costs and benefits are

for each project. The benefits are the positive effects of a project on the strategic goals of the organization. Furthermore, the infrastructure managers should be aware of dependencies between projects, that could imply that the total benefit of a portfolio of projects is not equal to the sum of the individual project benefits (Teng et al., 1996).

The strategic goals of a railway organization are not only based on internal desires, but also on those of external players. Different objectives can for example arise from different stakeholders, or from different strategic goals of an organization. These objectives are often conflicting and not easy to compare due to different measures. The infrastructure manager, who is responsible for the project selection, needs to consider the opinions of the multiple stakeholders of the railway network (Tzanakakis, 2013). Examples of railway network stakeholders are the government, local authorities, passengers, and freight users. Since infrastructure managers would like to optimize the objectives of all the stakeholders, they are actually trying to solve a multi-objective optimization problem. With insight into the effects of projects on the assets and the objectives, it is possible to solve the individual optimization problems of the infrastructure managers. However, due to the conflicting objectives, optimizing one objective at the expense of the others, is not preferable. In an environment with multiple stakeholders and thus multiple objectives, it would be better to select a solution that is a compromise between the objectives. This should be a well-founded decision, based on considerations regarding the costs and benefits of the possible solutions.

In the last decades, project portfolio optimization received more and more attention in the literature. Since organizations in all kinds of industries work with projects, and thus face the problem of project selection, the application of project portfolio optimization can be quite diverse. Examples of applications in literature are in Research & Development types of projects (Stummer et al., 2003; Killen et al., 2007), or in production projects (Walls, 2004), while other researches do not make any distinction in the type of projects (Ghasemzadeh et al., 2000; Carazo et al., 2010). The trend in project portfolio optimization is to consider multiple objectives. Due to these multiple objectives, a single optimal project portfolio might not exist, since there is no unambiguous way to make a compromise between different objectives. This means that also the selection of projects in a project portfolio that performs best according to the multiple objectives, is often not trivial. A simple way to select a compromise, is to give the different objectives weights, and optimize the weighted sum of the different objective functions. This however, could only be done if the objectives are commensurable. As stated in Marler et al. (2004), it is preferable to consider each objective individually, rather than to optimize a specific compromise. The result of such an optimization problem would then consist of multiple solutions corresponding to multiple compromises. This set of solutions, called the *Pareto* optimal set, could be the basis of negotiations, and could be used to support the decision making process. The decision makers could look into the details of different scenarios to gain insight. The final selection of a project portfolio should be the result of well-founded considerations, and insight into the effects on the strategic goals of the different potential portfolios.

There has been a significant amount of research in multi-objective optimization methods to solve such problems. These methods are designed to provide a set of solutions, all corresponding to a compromise of objectives, which are called *Pareto* optimal solutions. Since the selection and planning of projects is a combinatorial optimization problem, most of the optimization methods used in literature are not designed to consider all possible solutions. This because considering all possible solutions would simply take too much time, even on modern computers. In case a large infrastructure network and a large amount of projects is considered, this is definitely a point of attention. A large amount of information needs to be processed, and infrastructure managers are not willing to wait weeks on the results of a multi-objective optimization problem. Therefore, in the selection of an optimization method there is a trade-off between accuracy of the solutions and the computation time.

The information about the projects that is used in the selection process, is often not based on facts but merely on experience or simulation (Stamelos et al., 2001). Since the information provided by experts is mostly an expectation rather than a true value, it can be stated that the information is subject to uncertainty (Ghasemzadeh et al., 1999). Therefore, it would be more realistic to consider the uncertain nature of the information in the optimization problem, rather than working solely with expected values. This means that additionally to the large amount of information already needed for the network, also a lot of information is needed on the uncertain nature of the project information.

The subject of this thesis is the selection process of investment projects in railway infrastructure networks. The main research will be in the multi-objective optimization techniques that could be used to support the decision makers in the selection process. Next to this, different aspects and performance measures of such an optimization tool will be considered. First, the problem statement of this thesis will be presented, whereafter the research questions of the thesis will be discussed.

1.2. Problem statement

In the previous section, the relevance and complexity of the investment project portfolio selection process are delineated. The current approach in most railway infrastructure organizations is sketched, which leads to the following problem statement:

Decision makers in investment project portfolio planning in railway infrastructure networks do not have sufficient insight into the project portfolio optimization problem to make well-founded decisions.

The goal of this thesis is to change this situation. To do this, the investment project portfolio optimization problem should be formulated mathematically and an optimization method should be selected to solve this problem. More general, the goal is formulated as:

Design a decision support tool to provide insight into the investment project portfolio planning problem in infrastructure networks.

This problem statement will be the starting point of this thesis. In the design, the optimization method will play a central role, but also the visualization and user process will be reviewed. State of the art optimization methods will be used to provide the Pareto optimal set. The problem will be formulated mathematically such that it reflects reality as good as possible, while the complexity of the problem is minimized. Furthermore, the process will be designed such that it is usable in the business area of railway organizations.

1.3. Research questions

In order to achieve the goal that is set in the previous section, some research questions are composed. These research questions will be guiding in the thesis. In Chapter 5, the findings regarding these research questions are presented, and it will be reviewed to what extent these questions can be answered with the insights gained from the findings. The main research question of the graduation thesis will be:

How can multi-objective optimization be used efficiently in the decision support for investment project portfolio planning under uncertainty?

In order to answer this main question, some sub-questions are formulated. All the appointed aspects of the project portfolio selection process in railway networks, lead to the following sub-questions:

- *What is the most suitable way to include uncertainty in parameters and future scenarios in the optimization of maintenance and asset investment planning?*
- *In the high-level project portfolio optimization problem, what are the trade-offs between accuracy and computation time when using different time steps, network segments, asset information, and number of project dependencies?*
- *How can we cope with different requirements for the decision support tool in the railway infrastructure business, regarding computation time, solution quality, and the process of the support?*
- *Which optimization methods are best suited for the multi-objective project portfolio problem, regarding solution quality and computation time?*

1.4. Thesis outline

The four remaining chapters in this thesis are used to guide the reader along the track of the conducted research and way of thinking, to the argumentation for the answers on the research questions. Chapter 2 covers the required knowledge about the strengths and weaknesses of the existing literature on project portfolio optimization. The multi-objective project portfolio optimization problem is introduced, as well as some multi-objective optimization methods to solve such a problem. In Chapter 3, the proposed formulation for the investment project portfolio optimization problem is presented, containing an approach to include the uncertain nature of investment projects in the problem formulation. A detailed description of two multi-objective optimization methods that are selected to be suitable for project portfolio optimization is provided. The results that are obtained for the railway network case studies by implementing the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II are discussed in Chapter 4. In Chapter 5, the conclusions of the thesis are provided. Some possible improvements of the proposed methods are presented, along with further research that could be conducted. The thesis ends with some recommendations for ORTEC in their attempt to collaborate with rail infrastructure organizations to realize a decision support tool for project portfolio optimization.

Literature overview

To justify the novelty of the approach proposed in this thesis, a literature survey has been conducted. Many articles can be found that consider similar and related problems to the investment project portfolio optimization problem. The strengths and weaknesses of conventional mathematical representations of the project portfolio selection problem are discussed, to identify the shortcomings of existing approaches. Identified shortcomings are further examined to discover whether and how these subjects can be improved. Important assumptions that are broadly supported by the literature are examined, and either rejected or adopted. For example, the need to include multiple objective functions in the optimization problem is justified, and best practices in this field are reviewed as well. To make sure the proposed approach performs well, state of the art optimization techniques are reviewed. There has been research into suitable methods to measure the performance of multi-objective optimization techniques. Lastly, the visualization and decision support of project portfolio optimization approaches are reviewed. It is argued that this is an essential part of an approach to make it useful for the industry. To learn from the findings and the state of the art methods in literature, in this chapter an overview of the literature is presented.

2.1. Investment project portfolio optimization

Investment project portfolio optimization is a term that is used for a wide variety of optimization problems. This is mainly because the term *project* is very general. In this thesis, a *project* is defined as an undertaking that requires, time, money, and other resources. The term *investment project* is used for projects that contribute to a profitable return, such as interest, income, or an improvement of value. An *investment project portfolio* is a set of investment projects that are selected to be executed. In *investment project portfolio optimization*, one tries to find the set of investment projects that yields the best results regarding the objectives. If the result of an investment project portfolio is time-dependent, next to the selection of projects, also the scheduling of the projects needs to be considered.

Because the generality of the terms, investment project portfolio optimization has many different applications. In the literature, a lot of research can be found on project portfolio optimization. Some articles work with general projects, while others consider specific applications of project portfolio optimization. In this thesis, the application is in maintenance and asset investment of railway infrastructures. Therefore, both articles with a general point of view and those similar to the subject of this thesis, are reviewed.

In this section, different aspects of the investment project portfolio selection problem are reviewed, that are important for this thesis. First, the desires of decision makers are regarded, which are the objectives of the optimization problem. Next, an overview of ways to formulate a project portfolio optimization problem found in the literature is presented. The objectives and formulation of the optimization problem determine what kind of optimization problem is considered. Then some application-dependent aspects are discussed. The uncertain nature of the input data

of project portfolio optimization in infrastructure management gives rise to the question how uncertainty is handled in other optimization problems. Next to uncertainty in data, infrastructure management also has to deal with dependencies between projects.

The goal of investment project portfolio optimization is to support the decision making process in some way. Then, a review of how project portfolio optimization could support the decision making process is presented. Lastly, the optimization methods suggested in the literature are discussed. To summarize, the outline of this section is:

- 2.1.1 Objectives
- 2.1.2 Formulation
- 2.1.3 Level of detail
- 2.1.4 Uncertainty
- 2.1.5 Dependencies

2.1.1. Objectives

The investment projects that are proposed for an organization to execute, serve a certain purpose. This could be creating profit, improving processes, increasing safety, reducing costs, etc. All the executed projects together are supposed to contribute to the objectives derived from the strategic goals of the organization. Due to different stakeholders that are interacting with the organization, the goals might be conflicting to each other. To realize a project portfolio with the best overall result, the project management and selection should be centralized (Killen et al., 2007), since the overall optimal solution is often not equal to the combination of individual optima.

Cooper et al. (1997) suggests three main types of goals in project portfolio management: *value maximization*, *balance*, and *strategic direction*. This means that a project portfolio should maximize the overall values, but also should maintain a balance between individual values. This is to prevent that one part of an organization does very well, at the expense of another part of the organization. Furthermore, a project portfolio should contribute to the strategic direction of an organization, even if this portfolio does not yield the maximum value at this time.

Most articles consider only one objective, and other objectives are formulated as constraints. Morcoux et al. (2005) minimizes the total maintenance costs in infrastructure networks. Budai-Balke (2009) and Nourelfath et al. (2010) minimize the sum of maintenance costs and other costs related to production. Jafarzadeh et al. (2015) and Grimes (1995) focus completely on the financial benefits, and thus uses an objective function of profits and losses. Sriskandarajah et al. (1998) minimizes the costs of performing maintenance jobs too early or too late. So in fact the goal is to create a schedule of maintenance jobs that is closest to the due dates for the jobs. Li et al. (2009) considers an objective function of overall benefits, the sum of individual project benefits. Ghasemzadeh et al. (1999) actually acknowledges the existence of multiple objectives, but optimizes only a single objective: net present value. For multiple objectives, a combined objective function is proposed, obtained by using techniques such as weighted scoring or the analytic hierarchy process. A combined objective function still considers only one objective in the optimization process. However, it is preferable to explicitly consider the multiple objectives that exist in optimization problems (Marler et al., 2004). This is because decision makers usually have no knowledge about the effect on the individual objective functions when a combined objective function is used.

This is the reason that recently more articles identify multiple objectives in the project portfolio optimization problem, and are aiming at a solution that optimizes all objectives. For example, Fwa et al. (1994) is aiming to minimize maintenance costs, maximize the usage of yearly allocated budgets, and to minimize fluctuations of year-to-year maintenance activities. Quan et al. (2007) minimizes two objectives in the scheduling of maintenance: task completion time and the number of workers needed. Rabbani et al. (2010) considers three financial objectives in the project selection problem: minimize the total cost, maximize the total benefit, and minimize the total

risk. Berrichi et al. (2010) combines the objectives of production and maintenance: minimize the makespan for production, and minimize the system unavailability due to maintenance activities. Engelhardt-Funke et al. (2004), Podofillini et al. (2006), Iniestra et al. (2009), and Bai et al. (2015) propose different methods for optimizing investments in railway networks. Objectives in this kind of problem are given by for example minimizing investment costs, minimizing passenger waiting times, minimizing risk, maximizing social impact, maximizing condition of assets, and maximizing safety.

2.1.2. Formulation

The formulation of a project portfolio optimization problem is more than only a selection of objectives. In the formulation, the variables and parameters are used to represent the information that has effect on the objectives. The main goal in formulating the optimization problem is to make sure that the formulation contains all information to create a good representation of reality, while omitting information that is not needed in the optimization process. The decision variables represent the degrees of freedom of the optimization problem, the variables that can be adjusted by the algorithm. Constraints are used to indicate which solutions are acceptable, and which solutions are outside the range of realistic solutions.

Project portfolio optimization problems typically have a binary decision variable for each project, to indicate whether it has been selected or not. In case also the scheduling is included in the optimization problem, the starting period of the project execution is included as decision variable as well. This decision could be represented by a continuous variable per project, or could be given by a binary decision variable per project per time period, that is equal to one only for the time period in which the project starts. Furthermore, some problem-specific decision variables can be present in the formulation.

Currently, in most organizations the maintenance planning and investment project selection is done separately. Each department has their own budget, and there is no centralized project selection process. Investment projects in infrastructure networks usually involve the infrastructure assets, so they might share resources with routine maintenance. Therefore, it would be preferable to merge these processes and benefit from the synergies. A centralized process could contribute to an overview of the effect of projects on the strategic goals, such that projects with the best results for the entire network can be chosen.

The formulation of maintenance optimization problems is quite similar to the formulation of project portfolio optimization problems. In both formulations, a decision variable indicates whether the maintenance job or project is executed. In maintenance planning, the scheduling of the maintenance jobs is always included, while in project selection problems the scheduling could be omitted. In project portfolio optimization, the projects are not limited by repair and replacement activities, as is the case in maintenance optimization. The set of project characteristics could be diverse, and not limited to asset or maintenance related characteristics. To optimize the investment projects in infrastructure networks, maintenance activities could be included, but so could a lot of different types of projects. For example, purchasing new assets with different deterioration characteristics, projects for extension of the network, and organizational changes. But it is also possible to only include maintenance projects in the problem. Thus, maintenance optimization can be seen as a specific application of project portfolio optimization.

2.1.3. Level of detail

The level of detail considered in the project portfolio optimization problem could be influenced by for instance:

- the number of affected assets

- the number of time periods in the prediction horizon
- the length of these time periods.

Infrastructure maintenance and investment project optimization problems typically consider a large number of assets, and a long planning and prediction horizon. It depends on the part of the organization which level of detail in assets and time periods should be considered. In investment decision-making usually a more high-level overview is considered. For example in some articles investments are defined per asset-type, rather than per asset. Assets could also be grouped on geographical location, or on age. Since the long term benefits of investments are considered to investigate whether a project has to be selected, the scheduling is not very detailed or not considered at all. Maintenance optimization problems often focus on the scheduling of a set of maintenance activities, that are all selected for execution. This means that the selection could be omitted from the problem. Even though often not all the aspects of a problem are considered, infrastructure maintenance and investment project optimization problems tend to have a large number of possible solutions because of the combinatorial nature of the problems.

Small-sized optimization problems are possible to solve exactly, for instance by enumerating all possible solutions. However, for large-sized problems this is often not possible within acceptable time. The optimal solutions of large-sized problems can be approximated with methods that give faster and close to optimal results (Marler et al., 2004). In the design of an optimization problem there is a trade-off between the accuracy and the computational complexity.

In the literature, many articles do not consider the scheduling problem of investments (Liesiö et al., 2007; Engelhardt-Funke et al., 2004; Iniestra et al., 2009; Bai et al., 2015); however, a prioritization of the possible investments is provided. The amount of possible solutions is reduced to 2^N , where N is the number of possible investments. When the scheduling is included as well, the number of possible solutions is multiplied by the number of time periods considered. Due to time-dependent results and constraints within time-periods, it would be preferable to consider the selection and scheduling of projects together. The following articles include the selection and scheduling of investment projects together in the problem formulation: (Jafarzadeh et al., 2015; Ghasemzadeh et al., 1999; Doerner et al., 2006). The prediction horizon of the problems found in the literature deviates from 4 days to 20 years, and the number of time periods in the horizon deviates from 3 to 365. For investment project planning, it would be best to consider a long prediction horizon, to include the long-term effects of the investments, such as profits or less repeating maintenance jobs. Maintenance planning typically considers a short planning horizon and detailed asset information, to include urgent and reactive maintenance activities, next to the planned preventive maintenance activities.

When combining high-level decision-making of maintenance planning and investment planning, it is highly relevant that a suitable level of detail is chosen to consider in the optimization problem. For high-level decision-making, a detailed problem formulation might not be necessary. For example, not every detailed maintenance job should be included, but only maintenance activities with a large impact on the objectives and feasibility of the project portfolio should be taken into account. Therefore, the level of detail that is needed is quite problem dependent. As seen in some articles (Archer et al., 1999; Stummer et al., 2003), it is also possible to have some screening or pre-processing done prior to the optimization to reduce the size of the solution space. Though it might still not be possible to use exact methods to optimize the problem, convergence might be faster. It is also possible to have multiple phases in the decision making process, and perform investment project portfolio optimization for different prediction horizons with the same information. In this way, a balance could be created between long-term and short-term objectives.

2.1.4. Uncertainty

Not all information needed in the optimization of maintenance and investment project portfolio is deterministic. Often the information used in optimization problems is based on estimations, judgment-based, uncertain, or unknown information. In the literature, the uncertainties in the variables of the optimization problem are often not considered. The uncertain nature of deterioration is often taken into account in the design of the prescribed maintenance intervals, which are then used as a constraint in the maintenance planning. There are some examples of articles in which the optimization of maintenance planning is based on Markov-chain models for deterioration (Morcoux et al., 2005; Podofilini et al., 2006). Sometimes the occurrence of failures is represented by a probability, for example from a Poisson distribution (Nourelfath et al., 2010), or from the exponential probability distribution (Higgins, 1998; Berrichi et al., 2010). However, others use deterministic values to include deterioration (Fwa et al., 1994; Grimes, 1995; Sriskandarajah et al., 1998; Chikezie et al., 2013).

In project portfolio optimization sometimes the risk of the projects is included in the problem (Rabbani et al., 2010), but often the uncertainty in parameters of the system are not taken into account (Archer et al., 1999; Ghasemzadeh et al., 1999; Stummer et al., 2003; Iniestra et al., 2009; Bai et al., 2015; Doerner et al., 2006; Ahern et al., 2007). Costs and benefits of large investments are often treated as deterministic values, but those values usually involve uncertainty as well. Deviations in costs and duration of large investment projects in infrastructure networks can have significant consequences for the feasibility of a project portfolio. Therefore, it is important to make sure the chosen portfolio of investment projects gives satisfactory results in the case that the estimations of parameters were correct, and still gives acceptable results when the reality is deviant from the estimations. The robustness of a project portfolio is considered in Liesiö et al. (2007), in which intervals are used for incomplete information.

Different ways to include uncertainty in project parameters are presented in literature. Firstly, Monte Carlo simulation with a uniform probability could be used to provide estimates of parameters (de Poot, 2013), when a lower and upper bound is known. In some cases the uncertainty in parameters could be represented by a known probability density function, for example Engelhardt-Funke et al. (2004) used an exponential probability distribution. When there is no information for a probability density function, for example Shakes model could be used (Li et al., 2009). Fuzzy parameter representations are also widely used to deal with uncertainty (Vijayalakshmi Pai, 2016; Mohagheghi et al., 2015). Fuzzy objectives or fuzzy goals can be used in multi-objective optimization as well. The concept of α -Pareto optimality seems promising to include uncertainty in the presentation of the provided solutions. However, including uncertainty will have effect on the computation time of an optimization method. There is a trade-off between a good representation of reality and the computation time that is needed for optimization.

2.1.5. Dependencies

In infrastructure networks, maintenance activities or investment projects can be defined per segment. These undertakings might influence the performance of other nearby segments as well. The condition of infrastructure networks is not merely the sum of the condition of the individual assets, but is also determined by the connectivity of the network. The effect of projects on parts of a network is also affected by the connectivity of the network. Executing combination of projects might give different results than the sum of the individual results, due to project dependencies. The dependencies between projects and maintenance activities are often not considered in the literature (Jafarzadeh et al., 2015; Archer et al., 1999; Ghasemzadeh et al., 1999; Liesiö et al., 2007; Engelhardt-Funke et al., 2004; Bai et al., 2015). This issue already received some attention in the literature (Carazo et al., 2010), and there are a number of articles that take the network and project dependencies into account.

In Teng et al. (1996), different types of dependencies of transportation investment alternatives are defined. The relevance of a structured judgment process is stressed, for which two stages are proposed. First a classification should be made on the type of dependency, and the second step is to define the degree of this dependency.

Independent investment alternatives do not influence the performance of any other investment alternatives on any of the objectives, and vice versa. A visualization of the results of two investments is shown in Figure 2.1a. Alternative a does not affect a' and a' does not affect a , so the result of selecting both alternatives is given by the sum of the individual results of a and a' .

Two investment alternatives are called *complementary investment alternatives* if the result of selecting both alternatives is higher than the sum of the individual results, for at least one objective. In Figure 2.1b, the increased result of the objective is shown in the middle, which will be noted by $\alpha_{a,a'}$. The increased result is given by $a + a' + \alpha_{a,a'}$, with a positive $\alpha_{a,a'}$, if a and a' are complementary investment alternatives.

If selecting two investment alternatives implies that they can not achieve their maximal individual performance for at least one objective, they are called *substitutive investment alternatives*. The performance of one alternative is so to say (partly) replaced by the performance of another. This can be seen in Figure 2.1c. The result of selecting two substitutive investment alternatives a and a' is also given by $a + a' + \alpha_{a,a'}$, but now $\alpha_{a,a'}$ is negative.

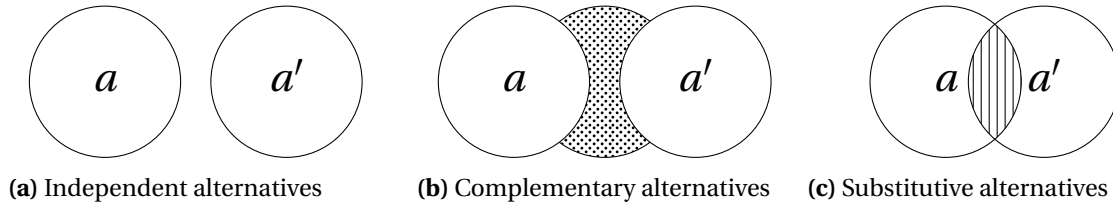


Figure 2.1: Different types of dependencies between projects, adapted from Teng et al. (1996), where $\alpha_{a,a'}$ is shown in the dotted and striped areas.

Often simple, straightforward dependencies such as complementary alternatives and substitutive alternatives are considered for combinations of two projects (Teng et al., 1996; Doerner et al., 2006; Rabbani et al., 2010). Only Stummer et al. (2003) and Iniestra et al. (2009) include interdependencies between different types of sets of projects. Since in reality the dependencies can be very complex, it is interesting to take a closer look at the added value of including more complex dependencies in large-sized problems. Whether the improved quality of the results compensate for the increased complexity of the problem, requires further research. However, simple dependencies can easily be implemented for a better representation of reality.

2.2. Multi-objective optimization methods

Since an investment project portfolio optimization problem often includes multiple objectives, there is a need to look further than the classical single objective optimization methods, and explore the possibilities in multi-objective optimization methods. There exist many optimization methods to solve multi-objective optimization problems, all with different benefits and disadvantages.

What the best multi-objective optimization method would be for the project portfolio optimization problem is discussed in this section. First, the concept of multi-objective optimization is explained in Section 2.2.1. The definition of a *Pareto* optimal solution is presented, and an overview of different approaches to find the Pareto optimal solutions is provided. Two methods that are considered for further research are explained more thoroughly, repeated ε -constraint algorithms and multi-objective optimization based on genetic algorithms. In Section 2.2.4, the possibilities of including uncertainty in the parameters of a multi-objective optimization problem are

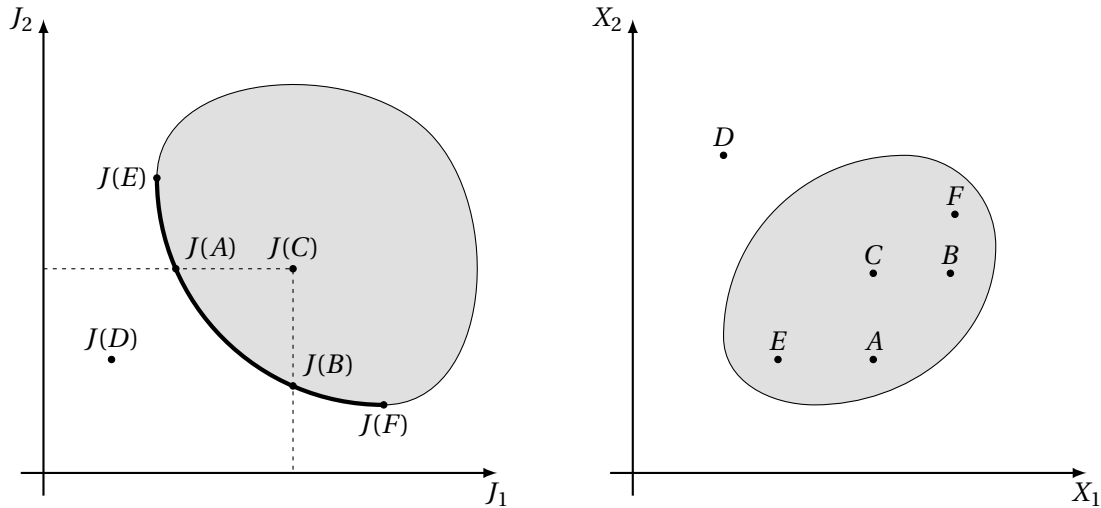
discussed. Lastly, some performance measures are discussed, to compare different multi-objective optimization methods.

2.2.1. Multi-objective optimization

In multi-objective optimization, a vector of m objective functions $J(x) = [J_1(x), \dots, J_m(x)]^T$ needs to be minimized, where $x \in X$ is the parameter vector or vector of *decision variables*. The feasible set $X \subseteq \mathbb{R}^n$ consists of all the parameter vectors x which satisfy the constraints $g(x) \leq 0$ and $h(x) = 0$. The corresponding set of attainable values of the objective functions is called the *objective space*, given by $\mathcal{J}$. The optimal solution that minimizes objective function $J_p(x)$ independently, is given by x_p^* . The optimal value of that objective function is then given by $J_p(x_p^*)$. Since a solution that minimizes all objective functions at the same time might not exist, the concept of Pareto optimality and efficient solutions are introduced.

Definition 1 (Pareto optimal solution). *For a Pareto optimal solution $x^* \in X$, it holds that there is no other feasible solution $x \in X$ that performs better on one or more objectives and does not perform worse on any other objective. So if for objective p it holds that $J_p(x) < J_p(x^*)$, then there must be at least one other objective p' for which it holds that $J_{p'}(x) > J_{p'}(x^*)$, presumed that x^* is Pareto optimal. In the literature, Pareto optimal solutions are sometimes called efficient solutions.*

The *Pareto optimal set* is the set of all Pareto optimal solutions, notated by $\mathcal{P}$. The set of corresponding objective values $\{J(x) \in \mathcal{J} | x \in \mathcal{P}\}$, is referred to as the *Pareto optimal front*. In Figure 2.2a, the objective space $\mathcal{J}$ of two objective functions J_1 and J_2 (gray) is presented, and the Pareto front is shown as a thick line on the border of the objective space, between $J(E)$ and $J(F)$. The feasible solution $C \in X$ is not Pareto optimal since $J_1(A) < J_1(C)$ and $J_2(A) = J_2(C)$. As can be seen in Figure 2.2b, where the corresponding solutions are visualized in the solution space, and the gray area is the feasible solution set X , D is not a feasible solution, so it does not matter that for example $J_1(D) < J_1(E)$ and $J_2(D) < J_2(E)$ for the Pareto optimality of E , as D cannot be a Pareto optimal solution. From these six solutions, $A, B, P, Q \in \mathcal{P}$ are Pareto optimal solutions. Which Pareto optimal solution is finally selected, depends on the preference of the decision maker.



(a) The objective space $\mathcal{J}$ with the objective values and the Pareto optimal front.

(b) The solution space, where the feasible solution set X is the gray area.

Figure 2.2: Pareto optimality, adapted from De Schutter et al. (2015).

In the case that the objective values are not known for all solutions in the feasible set, Pareto optimality can not be claimed by definition. Since this could happen in reality, an alternative for Pareto

optimality is introduced, *non-dominance*.

Definition 2 (Non-dominated solution). *Let $X_{\mathcal{R}} \subseteq X$ be the reference solution set, such that for $x \in X_{\mathcal{R}}$, the objective values J are known.*

For a non-dominated solution $x_{\mathcal{R}}^ \in X_{\mathcal{R}}$, it holds that there is no other feasible solution $x \in X_{\mathcal{R}}$ that performs better on one or more objectives and does not perform worse on any other objective. So if for objective p it holds that $J_p(x) < J_p(x_{\mathcal{R}}^*)$, then there must be at least one other objective p' for which it holds that $J_{p'}(x) > J_{p'}(x_{\mathcal{R}}^*)$, presumed that $x_{\mathcal{R}}^*$ is non-dominated.*

Since the concept of Pareto optimality has been introduced by Pareto (in 1906), a large number of multi-objective optimization methods has been presented in the literature. A distinction can be made between *a priori* methods, and *a posteriori* methods.

In the case that the relative preference between different objectives can be determined on forehand, methods with an *a priori* preference structure can be used. Methods with an *a priori* preference structure are capable of finding a single optimum, rather than the Pareto optimal set. In methods with an *a priori* preference structure a *general objective function* is defined, which is a combination of all the objective functions in the problem. Often this general objective function involves mutual preferences or utilities. If the mathematical formulation of a multi-objective optimization problem involves only one general objective function, this problem can be solved with single-objective optimization methods. Examples of *a priori* methods with a combined global objective function are the weighted sum method (Zadeh, 1963), the weighted exponential sum method (Athanasopoulos et al., 1996), and the weighted product method (Marler et al., 2004). In goal programming methods (Charnes et al., 1977) the decision maker can specify the desired *goals*, that are handled sequentially. In bounded objective function methods (Haimes et al., 1971), one objective function is optimized, while the other objectives are bounded by posed constraints.

Often it is not possible or desirable to determine the relative preference between different objectives on forehand. One approach would be to repeat the *a priori* optimization method with different weights on the objective functions, in order to represent the Pareto optimal set (Marler et al., 2004). Alternatively, there have been formulated several multi-objective optimization methods with an *a posteriori* preference structure, where the preferences can be determined with insight into the resulting solutions.

Repeatedly solving weighted multi-objective optimization problems with varying weights can result in a set of Pareto optimal solutions, but it may be difficult to select the right weights to achieve a well-distributed set of Pareto optimal solutions. Therefore, some multi-objective optimization methods are designed specifically to produce a set of Pareto optimal solutions that closely represents the complete Pareto optimal set. An example of a *a posteriori* method is physical programming (Messac et al., 2002), a method in which pseudo-preference values are selected such that they span the objective space, to find all the combinations of preferences that result in Pareto optimal solutions. The idea of the normal boundary intersection method (Das et al., 1998) is that the intersection of the boundary of the objective space and the normal on the convex hull of individual minima provides the part of the boundary that contains the efficient solutions. However, the performance of this method is highly dependent on the convexity of the objective space.

Classical multi-objective optimization methods generate a set of Pareto optimal solutions by modifying the problem such that the solution is obtained by solving multiple single-optimization problems, such as the repeated ε -constraint method. For large-sized multi-objective optimization problems these exact methods will take a lot of time. In contrast to exact methods, heuristics do not evaluate the whole search space. A specific search strategy is used to evaluate possible solutions, and with some assumptions they try to identify the most optimal solutions. In a heuristic approach one tries to create a balance between computational effort and the quality of the solutions. Single-objective heuristics are often modified to be used for the multi-objective case. Often

this is done by considering vectors of scalars.

Heuristics are often designed for a specific problem, and consequently are only useful for that specific problem. A *metaheuristic* can be seen as a solution concept, which is applicable to different problems. An example of a metaheuristic that can be used for multi-objective optimization problems is Pareto ant colony optimization (Doerner et al., 2001), which is inspired on the behavior of ants and their pheromones when they are searching for food. Other examples are genetic algorithms, tabu search, and simulated annealing.

For multi-objective optimization in a decision making process, it is preferred that the output is the set of Pareto optimal solutions. With the set of Pareto optimal solutions the decision makers are able to gain insight in the effects of certain solutions, and are supported to make well-founded decisions. Two methods are explained in more detail. First, a more classical approach is discussed: the repeated ε -constraint algorithm. Next, the concept of genetic algorithms is explained, and how such algorithms could be used for multi-objective optimization problems.

2.2.2. Repeated ε -constraint algorithm

In the *bounded objective function method*, only the most important objective function $J_s(x)$ is included in the general objective function. The other objective functions are added in the formulation as constraints, with a lower bound l_p and an upper bound ε_p . The optimization problem can be formulated as follows:

$$\begin{aligned} \min_{x \in X} J_s(x) \\ \text{s.t. } l_p \leq J_p(x) \leq \varepsilon_p, \quad p = 1, \dots, m, p \neq s. \end{aligned} \quad (2.1)$$

Haimes et al. (1971) introduces a strongly related approach, the *ε -constraint method*. In this approach the lower limit is omitted. Varying properly selected bounds ε_p will provide Pareto optimal solutions (Hwang et al., 1979). A standard for the selection of the bound ε_p is given in Carmichael (1980):

$$J_p(x_p^*) \leq \varepsilon_p \leq J_s(x_p^*). \quad (2.2)$$

Using the Lagrange multipliers it can be shown that a constraint is active. Active ε -constraints in an optimization problem indicate that the solution is Pareto optimal, since the solution can not be improved in the direction of J_s without worsening one or more of the other objective values. However, for integer problems this method will be less useful, since the bound ε_p will often not be equal to a feasible objective value of an objective J_p .

In Figure 2.3, the working of the repeated ε -constraint algorithm is visualized. Both objectives J_1 and J_2 are to be minimized. Minimizing J_1 without constraints results in $x_{1,0}^*$. The three single-objective optimization problems with J_1 as general objective, and the altering constraint $J_2 \leq \varepsilon_2$ results in the three solutions $x_{1,1}^*$, $x_{1,2}^*$, and $x_{1,3}^*$ respectively. Since in the ε -constraint algorithm there is no lower bound posed for the remaining objectives, it might happen that two or more values for the ε -bounds result in the same solution. This indicates that the concerning area of J_2 values does not contain solutions with lower values for J_1 .

2.2.3. Multi-objective genetic algorithms

The concept of *genetic algorithms* is introduced in Holland (1992), and widely used in literature since. The method owes its name to the similarity to Darwin's evolution theory. The *population* consists of potential solutions. The decision variables that represent these potential solutions can be seen as the genes of that individual solution. The general idea is that there are *parents* selected

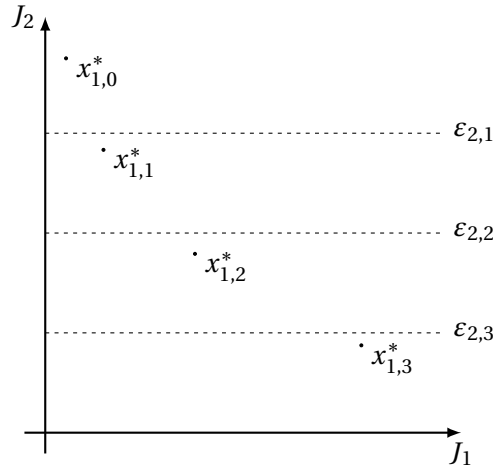


Figure 2.3: Visualization of the repeated ε -constraint algorithm for minimizing J_1 and J_2 . The general objective is J_1 , and J_2 is bounded by the ε -bounds.

from the population to create a new *generation* of *off-spring*, and due to genetic operations the fittest will survive over generations. An advantage of a genetic algorithm is that it can handle all kinds of optimization problems. There are no requirements for the type of objective functions, constraints or objective space. The method does not use gradients but function values only. It is a metaheuristic, providing good solutions within reasonable computation time, but (Pareto-) optimality can not always be guaranteed. Next, some important definitions are described.

Encoding Before the algorithm can start, an encoding scheme has to be selected. This encoding scheme is used to represent a vector of decision variables as a genetic string. Such a genetic string is also referred to as a *chromosome*. Binary strings (as used in Holland (1992)) are widely used, but alternatives such as real number or integer encoding are also found in literature.

Generating generations To start the algorithm, an initial population should be generated. For example a random selection of possible solutions. The population of N solutions is represented by the set of (binary) strings: $\{b_1, b_2, \dots, b_N\}$. A new generation is formed with two types of genetic operations: *mutation*, in which a new individual arises from a single individual with some changes in the genes, and *crossover*, in which the genes of two individuals are used to create a new individual. The combination of these operations makes that genetic algorithms use both random and local search. In creating new individuals, it is important that these individuals are still feasible solutions, or is even a solution to the given problem at all. These issues could arise in the mapping between the encoding space and the (feasible) solution space. To handle the issue of infeasibility, a problem that arises often in constrained optimization problems, penalty methods can be used. For *illegality*, repair techniques can be used to transform the individual to a solution. Gen et al. (1997) discuss a third type of technique to handle constraints in genetic algorithms: rejecting techniques. With such a technique, infeasible solutions are not selected in for the next iteration. An iteration in the algorithm is associated with a generation, which is denoted by t . The population of a generation is then denoted by $P(t)$.

Selection The strength of genetic algorithms lies in the selection method. A selection method that performs well, leads the search to promising new generations. Gen et al. (1997) discusses the following type of selection methods: roulette wheel selection, $(\mu + \lambda)$ -selection, tournament selection, steady-state reproduction, ranking and scaling, and sharing.

The *roulette wheel selection method* is a proportional stochastic sampling method. For each individual solution in the population the *fitness value* $f(b_i)$ is determined. The roulette wheel, with a bin for each individual in the population proportional to its selection probability (relative fitness value), is spinned N times. These N (not necessarily different) individuals are selected as parent.

In ranking methods, also referred to as *elitist selection methods*, the best individuals in a population are selected as a parent. In (μ, λ) -selection, the μ best off-spring individuals are selected as a parent. $(\mu + \lambda)$ -selection differs in the sense that the best μ individuals are selected from the off-spring and old parents. With an original population of size λ , these methods reduce the size of populations over generations by selecting $\mu \leq \lambda$. These ranking methods prevent individuals to be selected multiple times. For combinatorial optimization problems this is very useful.

Tournament selection uses earlier mentioned selection methods on randomly chosen subsets of the population. From this set of solutions the best is selected (ranking), or roulette wheel selection could be applied, to continue to the new generation. N tournaments are held to fill the new population.

Crossover After the selection of the parents, the off-spring is created. To introduce variation in the new population and to stimulate random search, *crossover* is conducted. A new genetic string is composed from the genetic strings of the parent solutions. There exist many different ways to do this. A widely used method is *single-point crossover*, in which a single point in the string is selected, and the parts after that position are interchanged. Other crossover methods are often proposed for problems with more complex characteristics.

Mutation In the crossover process the genes of the parents are interchanged, but no new genes are introduced. So if a specific gene is not present in a population, it will not appear in new generations unless *mutation* occurs. Mutation is the random change of individual genes. When a binary encoding is used, mutation is usually interpreted by randomly flipping the bits of the offspring. Each gene in the offspring is mutated with a chosen probability p_m . When another encoding is used, random values are selected to replace a mutated gene.

Multi-objective genetic algorithms Genetic algorithms can also be applied to multi-objective optimization problems, which is then called *evolutionary multi-objective optimization* or *genetic multi-objective optimization*. The main challenge in this application is determining the fitness value of individuals, regarding multiple objectives. Gen et al. (1997) classifies different methods for fitness assignment. Pareto-based approaches provide a Pareto ranking for a population, which is based on non-dominance. The fitness value of an individual depends on the Pareto ranking. A notable feature of this approach is that non-dominated individuals receive the same fitness value, which is not the case in all other fitness assignment methods.

2.2.4. Uncertainty in multi-objective optimization

In classical optimization approaches, the values of parameters that are subject to uncertainty are usually fixed to their average or a typical value. In real-life, it is reasonable to assume that these values might deviate from the fixed parameters. In case that influential decisions are based on the results of optimization with these parameters, it would be more realistic to take the uncertain nature of the parameters into account.

One approach is to simulate realizations of the parameters, and include those in the optimization problem (Gabriel et al., 2006). In this approach, that is called Monte Carlo simulation, one needs to select the probability distributions from which the realizations are drawn. In practice, usually there are no probability distributions available for the parameters with uncertainty.

Another approach, mostly used in financial optimization, is to minimize both the expected return and the variance (risk) (Ghaoui et al., 2003). However, for this approach you also need probability distributions. Sakawa et al. (1987) states that it would be more appropriate to interpret the values of judgment-based parameters as fuzzy numbers. These could then be used in fuzzy multi-objective optimization problems, to find the α -Pareto optimal solutions.

Liesiö et al. (2007) uses intervals (a lower and an upper bound) for incomplete information in optimization problems, which leads to an interval of overall objective values. With some adjustments to the classical definition of dominance, it is possible to represent the dominance relations of objective intervals in multi-objective optimization.

2.2.5. Performance indicators

To compare the performance of different methods, performance indicators need to be chosen that are able to represent the quality of the approximation set of a certain method. Zitzler et al. (2003) deals with the question how the performance of multi-objective optimization methods should be compared. Decision makers, just like the majority of the people, want the solutions to their problems as fast as possible. Since the computation time of different optimization algorithms or even different settings of an algorithm could be far apart, it is important to consider this in the selection of a method. The computation time is easy to measure and compare, and therefore is widely used as performance indicator. Next to the computation time, the quality of the results from the multi-objective optimization method should be taken into account. Okabe et al. (2003) compares several performance indicators that can be found in the literature. Performance indicators are divided in cardinality-based indicators, accuracy indicators, and distribution indicators. For better understanding of the performance of an algorithm, it is wise to consult different types of performance indicators, for different test runs of the algorithm. The approximation of a Pareto-optimal set is considered suitable for comparing the performance in a quantitative manner. For example the distance to the Pareto-optimal set, the hypervolume measure, or the diversity could be used as quality indicators. Statements can be made on dominance relations with respect to Pareto-optimality. However, these statements do not cover the entire optimization method, solely the dominance of approximation sets.

2.3. Multi-objective project portfolio optimization

2.3.1. Multi-objective optimization methods in project portfolio optimization

In the literature, the multiple objectives in project portfolio optimization are handled in different ways. In some cases the different objectives are weighted prior to the project portfolio selection process. In this way the multiple objectives can be written as one global objective function, enabling the use of simpler and more explored optimization methods. Examples of articles that use the weighted sum method to combine multiple objective functions are Fwa et al. (1994), Teng et al. (1996), Higgins (1998), Liesiö et al. (2007), Ahern et al. (2007), and Ghapanchi et al. (2012).

Using a weighted sum of objectives implies a preference structure prior to optimization. Decision makers are not able to gain insight in the effects of adjusting the weights they have given, which is not desirable (Ghasemzadeh et al., 1999). Therefore, the use of multi-objective optimization methods with an a posteriori preference structure increased during the past decades. Such multi-objective optimization methods search for *Pareto* optimal solutions. These can be evaluated after the optimization process. Examples of articles that use a posteriori preference methods in project portfolio optimization are Stummer et al. (2003), Engelhardt-Funke et al. (2004), Dörner et al. (2006), Podofilini et al. (2006), Quan et al. (2007), Iniestra et al. (2009), Rabbani et al. (2010), Berrichi et al. (2010), Chikezie et al. (2013), Bai et al. (2015), and Chen et al. (2015).

Due to the computational effort, that is needed to solve the combinatorial multi-objective optimization problems, often (meta-) heuristics are used to obtain satisfactory results within accept-

able time (Higgins, 1998; Budai et al., 2006). Genetic algorithms are widely accepted as an efficient solution technique for multi-objective optimization problems (Fwa et al., 1994; Grimes, 1995; Sriskandarajah et al., 1998; Morcous et al., 2005; Quan et al., 2007; Podofillini et al., 2006; Noureldath et al., 2010; Chikezie et al., 2013; Engelhardt-Funke et al., 2004; Iniestra et al., 2009; Bai et al., 2015). Although some articles provided solution techniques with even faster computation times, in general genetic algorithms with reasonable tuned parameters are praised for their computation time.

One of the techniques which is in some articles presented to provide better results than genetic algorithms is Pareto ant colony optimization (Berrichi et al., 2010; Doerner et al., 2006). This solution technique is less explored in literature, but might be interesting for further research. Another technique to find close to optimal solutions with short computation time is to solve the LP-relaxation of a mixed-integer formulated problem (Budai et al., 2006; de Poot, 2013). The formulation of the problem has more restrictions than for genetic algorithms, which can handle all kinds of objective functions, constraints and objective spaces. Some other solution techniques used in the literature are dynamic programming (Liesiö et al., 2007) and particle swarm optimization (Rabbani et al., 2010).

2.3.2. Visualization and decision support

For the use of multi-objective optimization in decision support tools, an a posteriori method is preferred, since the results provide insight into the effects of certain preferences on all the objectives. These effects might not be known or evident, which makes it difficult for a decision maker to provide preferences a priori.

To support the decision-making process, the set of Pareto optimal solutions that is provided by an a posteriori method needs to be presented in a clear way. Since in multi-objective optimization a trade-off has to be made between the objectives, a well-designed visualization of Pareto optimal solutions can be very helpful. For example the shape and range of the Pareto front can provide insights for decision makers to select their preferred solution. Especially in the comparison between different approximations of the Pareto front, it is important that the Pareto dominance relations are clear in the visualization. Furthermore, the visualization method should be simple to understand for decision makers, to really provide good insights.

Tušar et al. (2015) presents a review of visualization methods for Pareto fronts. Properties of visualization methods like robustness, handling of large sets, simplicity, preservation of the dominance relation, and preservation of the front shape are considered for different methods.

Simplicity is a very important aspect for decision support tools, since users are often not familiar with reading complex plots. Two examples of relatively simple visualization methods are shown in Figure 2.4. In a scatter plot matrix, the classic 2D representation of a Pareto front for 2 objectives is presented for all combinations of objectives. A scatter plot matrix is an interesting method, since the front shape and distribution of solutions are preserved. However, it will be difficult to get an overview of the solutions, as it is not clear which of the dots belong to the same solution. In a parallel coordinates plot, the different objectives are represented by scaled vertical axis. Each solution is represented by a connection line between the different function values for that solution. A parallel coordinates plot is good in terms of scalability, it is easy to extend the visualization for many objectives. However, the front shape is not clear in the parallel coordinates plot, and the overview will be lost if too many solutions are plotted.

Even though the visualization methods described are relatively simple, it is still not easy to understand how the results should be interpreted. Therefore, it might not be sufficient to present one figure with the results of a complex multi-objective optimization problem. One of the methods described could be used for an overview, after which other methods could be used to gain insight into the details of the different solutions.

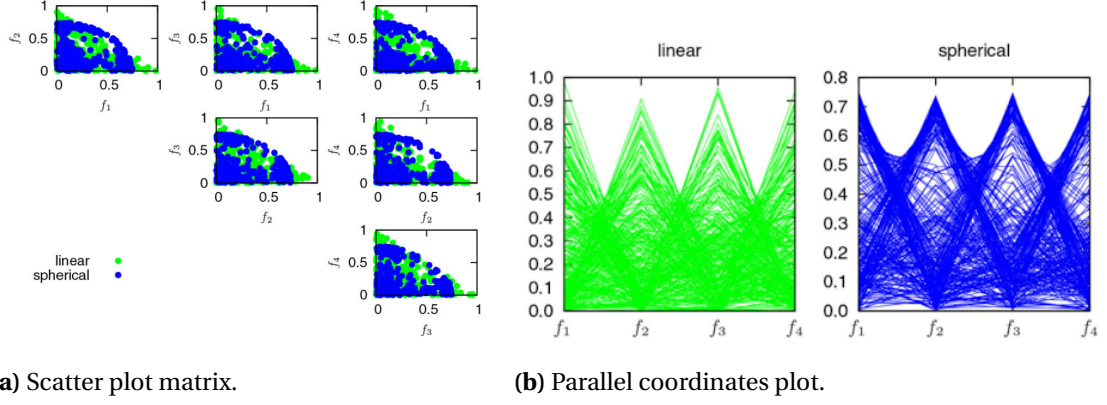


Figure 2.4: Visualization methods, as presented in Tušar et al. (2015).

Next to the visualization of the Pareto front, additional information could be provided for decision making. A good example of this is the *core index* of projects in project portfolio selection. Liesiö et al. (2007) introduced this term, defined as follows:

Definition 3 (Core index). *The core index of project a with regard to the reference set $\mathcal{R}$ of non-dominated project portfolios $x_{\mathcal{R}}$ is given by:*

$$CI(a, \mathcal{R}) = \frac{|\{x_{\mathcal{R}} \in \mathcal{R} | a \in x_{\mathcal{R}}\}|}{|\mathcal{R}|}$$

The core index of a project reflects how likely it is that the project is selected in a non-dominated project portfolio. If the core index is equal to 1, the project is selected in all non-dominated solutions. A project with a high core index reflects a project that scores well on the combination of considered objectives. If a decision maker wants to force a project to be in the final project portfolio, the core index indicates the fraction of the non-dominated set that is still feasible.

2.4. Summary

The literature on project portfolio optimization is used to identify the best practices and open issues in this field. For project portfolio optimization in infrastructure networks, maintenance projects can be included in the problem, from a high-level point of view. It can be concluded that it is preferable to include multiple objectives in the optimization problem, such that the Pareto optimal set of solutions could be used in the decision-making process. In the mathematical formulation of the optimization problem it is important to include possible dependencies between projects and assets. Furthermore, one must be aware of the uncertain nature of some of the variables in the problem. If possible, it would be wise to include information on these uncertainties, such that a project portfolio can be selected that performs well, even if the parameters or variables deviate from their expected values. Different multi-objective optimization methods are discussed, and what performance indicators might be useful in the selection of a suitable algorithm. Lastly, the visualization and decision support of project portfolio optimization problems is discussed. Simple visualizations of the results are desirable, but difficult to realize for problems that regard three or more objectives.

In Table 2.1, an overview is presented of articles on the optimization of maintenance and investment projects planning. This could be used for quick insights into the tendency of including uncertainty, scheduling, and dependencies in the problem formulation, as well as the number of objectives considered and the optimization techniques used to optimize these.

Article	Uncertainty	Scheduling	Dependencies	Objectives	Decision criteria	Optimization technique
Chen et al. (2015)	Not included	10 years, per year	Independent	2	A posteriori	Dichotomic approach
Fwa et al. (1994)	Not included	20 years, per year	Independent	3	Weighted sum	Genetic algorithm
Grimes (1995)	Not included	Not included	Independent	1	-	Genetic algorithm
Sriskandarajah et al. (1998)	Not included	1 year, per day	Independent	1	-	Genetic algorithm
Morcous et al. (2005)	Markov-chain	15 years, per year	Independent	1	-	Genetic algorithm
Budai-Balke (2009)	Not included	2 years, per week	Independent	1	-	Genetic algorithm, tabu search, heuristics, simulated annealing
Nourelfath et al. (2010)	Poisson process	5 months, per month	Independent	1	-	Genetic algorithm, exhaustive evaluation
Quan et al. (2007)	Not included	Minimized horizon	Independent	2	Search strategy	Genetic algorithm
Podofilini et al. (2006)	Markov-chain	Not included	Independent	2	A posteriori	Genetic algorithm, crude search
Chikezie et al. (2013)	Not included	20 years, per year	Independent	2	A posteriori	Genetic algorithm
Higgins (1998)	Exponential distribution	4 days, per hour	Independent	2	Weighted sum	Tabu search
Berrichi et al. (2010)	Exponential distribution	Minimized horizon	Independent	2	A posteriori	Pareto ant colony optimization, NSGA II, SPEA II
Jafarzadeh et al. (2015)	Not included	Flexible horizon	Independent	1	-	Branch & bound
Li et al. (2009)	Shackle's model	Not included	Independent	1	-	LaGrangian relaxation
Teng et al. (1996)	Not included	Not included	2 projects	4	Weighted sum	Heuristic
Ghasemzadeh et al. (1999)	Estimated risk values	10 periods	Sets of projects	1	-	Branch & bound
Liesiö et al. (2007)	Intervals	Not included	Independent	4	Weighted sum	Dynamic programming
Stummer et al. (2003)	Not included	3 periods	Sets of projects	5	A posteriori	Brute force algorithm
Engelhardt-Funke et al. (2004)	Probability density function	Not included	Independent	2	A posteriori	Genetic algorithm
Iniestra et al. (2009)	Not included	Not included	Sets of projects	5	A posteriori	Genetic algorithm
Bai et al. (2015)	Not included	Not included	Independent	5	A posteriori	Genetic algorithm
Doerner et al. (2006)	Not included	3 periods	Set of projects	2	A posteriori	Pareto ant colony optimization, NSGA II, Simulated annealing
Ahern et al. (2007)	Not included	Not included	Independent	9	Weighted sum	Goal programming
Rabbani et al. (2010)	Estimated risk values	3 periods	2 projects	3	A posteriori	Particle swarm optimization, SPEA II

Table 2.1: Overview of articles on the optimization of maintenance and investment projects planning.

Investment project portfolio optimization method

With the findings and lessons from the literature presented in the previous section, some choices can be made on the formulation and the optimization method for the investment project portfolio optimization problem. The choices that are made are presented in this chapter. First, the general investment project portfolio optimization problem is formulated mathematically in Section 3.1, and the railway infrastructure networks specific functions are presented in Section 3.2. The repeated ε -constraint algorithm and the genetic algorithm NSGA II are selected to solve the investment project portfolio optimization problem. The details of these approaches are discussed in Section 3.3, as well as the performance indicators that will be used in the comparison between the algorithms.

3.1. Optimization problem formulation

In this section, the mathematical formulation of the general investment project portfolio optimization problem for infrastructure networks will be presented. The notation of Su et al. (2017) will be used as a guideline for corresponding variables and parameters. The same notation could be used for other investment project portfolio optimization problems that consider the states of assets or components, and a set of possible projects to influence these states. An overview of the notation is presented at the end of Section 3.2.

The optimization problem considered in this thesis concerns the selection and planning of investment projects in an infrastructure network. The *planning horizon*, also called the *control horizon*, consists of N_c time periods. To be able to make well-founded decisions, the state of the infrastructure network will be predicted for the *prediction horizon*, consisting of N_p time periods ($N_p > N_c$). The investment projects can also be scheduled in time periods after the planning horizon. However, only the investment projects within the planning horizon will be used for the detailed planning and execution of the projects. The optimization problem will be reconsidered after the planning horizon, with updated information and again a prediction horizon of N_p time periods. This concept is called a *rolling horizon*.

An infrastructure network is considered, consisting of n assets or components (referred to as assets), that do not necessarily have to be of the same asset type. This set of assets C may also include some assets that are not yet build or in use. Some static information on the asset could cover for example the asset type, the installation date of the asset, and the location of the asset. Time-dependent asset information is considered as the state of the asset. The condition of the j^{th} asset in time period k is given by $x_j^{\text{con}}(k) \in [0, 1]$. A condition of 1 means that the concerning asset is as new, so in full state. An asset with a low condition is expected to fail more often, and or to deliver lower performance. The condition of an asset may deteriorate over time, but the deterioration could also depend on for example the environment and the usage. Together with some auxiliary variables $x_j^{\text{aux}}(k)$, the state of the j^{th} asset in time period k is given by:

$$x_j(k) = \begin{bmatrix} x_j^{\text{con}}(k) \\ x_j^{\text{aux}}(k) \end{bmatrix} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p\}. \quad (3.1)$$

In the auxiliary variables one could store information on the asset that is relevant for the selection of investment projects, or information that is needed to compute other variables. Examples of auxiliary variables are the usage of the asset in a time period, or the operational expenses needed for that asset in a time period.

Each asset is linked to a set of possible investment alternatives $\{a_0, \dots, a_{N_j}\} = A^j \subseteq A$, where N_j is the number of maintenance or investment alternatives for the j^{th} asset, and A is the set of all N possible alternatives. Each alternative could represent a maintenance activity $a \in A^M \subseteq A$, that could be executed multiple times in the prediction period, or an investment project $a \in A^I \subseteq A$, that could be executed only once. The alternative $a_0 \in A^0$ represents when no maintenance activity or investment project is selected. There is no limit for how many times a_0 could be 'selected' in a portfolio. Let v_a for $a \in A$ be the maximum number of times that the alternative a may be executed in the prediction period. Maintenance activities could be repairs or renewals. Investment projects could be a replacement, building an asset, breaking down an asset, or other reorganization projects.

In a time period k , only one alternative could be executed. For a_0 it is by definition that no maintenance or investment project could be executed, but also for other alternatives it holds that a project can not start while another project is being executed. Let $u_j(k)$ represent the alternative $a \in A^j$ that starts for the j^{th} asset in time period k . The binary representation of the start periods of the alternatives are given by:

$$\gamma_a(k) = \begin{cases} 1 & \text{if alternative } a \text{ starts in time period } k \\ 0 & \text{if alternative } a \text{ does not start in time period } k \end{cases} \quad (3.2)$$

$$\forall a \in A, \forall k \in \{1, \dots, N_p\}.$$

So $\gamma_a(k) = 1$ if and only if $u_j(k) = a$, for $a \in A^j$. The duration of alternative $a \in A$ in time periods is given by d_a . This means that the alternative is executed in d_a periods, starting from time period k for which $u_j(k) = a$. Let $\eta_a(k)$ be the binary representation of the execution periods of the alternatives:

$$\eta_a(k) = \begin{cases} 1 & \text{if alternative } a \text{ is being executed in time period } k \\ 0 & \text{if alternative } a \text{ is not executed in time period } k \end{cases} \quad (3.3)$$

$$\forall a \in A, \forall k \in \{1, \dots, N_p\},$$

for which $\eta_a(k) = \sum_{l=0}^{d_a} \gamma_a(k-l)$. For each asset, only one alternative could be executed in a time period. This gives rise to the constraint:

$$\sum_{a \in A^j} \eta_a(k) \leq 1 \quad \forall j \in \{1, \dots, n\}, k \in \{1, \dots, N_p\}. \quad (3.4)$$

This constraint and the definition of $\eta_a(k)$ make that it is also impossible that an alternative starts for the j^{th} asset starts while another alternative is still being executed. A maintenance or investment project has effect on the state of the assets from the moment the execution is finished. Therefore, the binary variable that represents whether the execution of alternative $a \in A$ is finished in time period k is introduced:

$$\lambda_a(k) = \begin{cases} 1 & \text{if the execution of alternative } a \text{ ends in time period } k \\ 0 & \text{if the execution of alternative } a \text{ does not end in time period } k \end{cases} \quad (3.5)$$

$$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p\},$$

for which $\lambda_a(k) = \gamma_a(k - d_a)$. Similar to $u_j(k)$, let $w_j(k)$ represent the alternative that ends in time period k , so for $a \in A^j$ it holds that $\lambda_a(k) = 1$ if and only if $w_j(k) = a$. An alternative ends $d_{u_j(k)}$ periods after it started: $u_j(k) = w_j(k + d_{u_j(k)})$.

Some alternatives might affect multiple assets. In that case, the alternative needs to be executed in the same time periods for all affected assets. So if assets j and j' have a mutual alternative $a \in A^j \cap A^{j'}$, then $u_j(k) = u_{j'}(k)$, and $w_j(k) = w_{j'}(k)$. This makes sure that an alternative can not be executed if for one of the concerning assets, another alternative is being executed.

Due to deterioration, the condition of the assets decreases over time, and some auxiliary variables may also change. Maintenance or investment alternatives may improve the condition of the assets, and can also have effect on the auxiliary variables. Both the effect of the deterioration process and the effect of maintenance and investment alternatives are uncertain, so predictions and expectations are used in the optimization problem. For a realistic representation, one could include the uncertain nature of these predictions and expectations in the formulation. Let $\theta_j(k) \in \Theta_j$ be the realization of the uncertainties related to the condition of the j^{th} asset in time period k , and $\theta(k) = [\theta_1(k), \dots, \theta_n(k)]^T$. In general it is assumed that if θ is added up in the objective function, the expected value of θ is 0, or if θ is a multiplier, that the expected value of $\theta=1$. The generic model for the state of the assets is then given by:

$$x_j(k+1) = f_j(x_j(k), w_j(k), \theta_j(k)) \quad (3.6)$$

$$= \begin{cases} f_j^0(x_j(k), \theta_j(k)) & \text{if } w_j(k) = a_0 \in A^0 \text{ (no maintenance)} \\ f_j^m(x_j(k), \theta_j(k)) & \text{if } w_j(k) = a \in A^M \text{ (} a \text{ is a maintenance project)} \\ f_j^i(\theta_j(k)) & \text{if } w_j(k) = a \in A^I \text{ (} a \text{ is an investment project)} \end{cases} \quad (3.7)$$

$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p - 1\}.$

In the functions f_j , some parameters are included for the effect that an alternative has on the state of the concerning assets. If no maintenance or investment projects are executed, the change in the state of assets is only due to deterioration. So the state of an asset in the next time period depends on the current state and some uncertainties due to for example measurement errors and model inaccuracies. When a maintenance project is finished, the state of the asset in the next time period will depend on the expected effect of the alternative as well. Lastly, if a renewal or replacement project is executed, the condition and auxiliary variables of the asset in the next time period are given by the "as good as new" condition and corresponding auxiliary variables, and depending on uncertainties in the exact condition and expected auxiliary variables.

Next to the constraint for the execution of alternatives, also constraints on the states of the assets must be considered, as well as global constraints for the whole infrastructure network. An example of a logical local constraint is that the condition of an asset should be in the interval $[0, 1]$, so deterioration stops if the condition is 0, and maintenance or renewal projects can not result in a condition higher than 1. Global constraints could arise for example from limited available resources for the whole infrastructure network. All the constraints that must be met for the whole system are given by:

$$g(x(k), \gamma(k), \eta(k), \lambda(k), u(k), w(k), \theta(k)) \leq 0 \quad \forall k \in \{1, \dots, N_p\}, \quad (3.8)$$

where $x(k) = [x_1(k), \dots, x_n(k)]^T$, $\gamma(k) = [\gamma_{a_0}(k), \dots, \gamma_{a_N}(k)]^T$, $\eta(k) = [\eta_{a_0}(k), \dots, \eta_{a_N}(k)]^T$, $\lambda(k) = [\lambda_{a_0}(k), \dots, \lambda_{a_N}(k)]^T$, $u(k) = [u_1(k), \dots, u_n(k)]^T$, and $w(k) = [w_1(k), \dots, w_n(k)]^T$.

Now that the formulation of the states and constraints is presented, the m objectives in the optimization problem can be denoted by:

$$J(x, \gamma, \eta, \lambda, u, w, \theta) = [J_1(x, \gamma, \eta, \lambda, u, w, \theta), \dots, J_m(x, \gamma, \eta, \lambda, u, w, \theta)]^T, \quad (3.9)$$

where $x = [x(1), \dots, x(N_p)]$, $\gamma = [\gamma(1), \dots, \gamma(N_p)]$, $\eta = [\eta(1), \dots, \eta(N_p)]$, $\lambda = [\lambda(1), \dots, \lambda(N_p)]$, $u = [u(1), \dots, u(N_p)]$, $w = [w(1), \dots, w(N_p)]$, and $\theta = [\theta(1), \dots, \theta(N_p)]$. In the investment project portfolio optimization problem, the long term behaviour is considered. Therefore, the objective functions often consider the states in the entire prediction horizon. This could be by summation over states in the time periods in the prediction horizon, but also by taking the average of the states.

The set of all possible realizations of the uncertain information, Θ , can be very large. Since for each time period, all those possible realizations might be different, the set of possible realizations for the entire prediction period Θ^{N_p} will be huge. It might, however, not be necessary to consider all possible realizations. In the end, the goal of the optimization is to support the decision making process by considering a good variety of possible cases. Although it is useful to include more information on parameters than just the expected value, to gain more insight, for some applications it might be sufficient to consider the expected values and the extreme values (lower and upper bounds of the uncertain variables). To reduce the additional effort of taking into account the uncertainties, not all sources of uncertainty will be considered in this problem formulation. The available information about the investment alternatives is often based on the judgment and expectations of experts. The costs and benefits could deviate from these expected values, due to for example errors in calculation, non-fulfilled agreements with external parties, and unexpected delays. Uncertainties in these variables have a great impact on the results, and are therefore selected to be included in the problem formulation. Since deterioration models and measurements become more and more accurate, the deviations in predicted conditions will not be taken into account, but only deviations in the conditions that are the result of investment alternatives. This means that when no investment alternative is executed for an asset, the lower bound, expected value, and upper bound for the state of that asset are equal.

Let $\underline{\theta}_j(k)$ be the lower bound of the possible realizations, and $\overline{\theta}_j(k)$ the upper bound of the possible realizations. Then the lower and upper bounds of the state of the assets is given by:

$$\underline{x}_j(k+1) = \begin{cases} f_j^0(\underline{x}_j(k)) & \text{if } w_j(k) = a_0 \in A^0 \text{ (no maintenance)} \\ f_j^m(\underline{x}_j(k), \underline{\theta}_j(k)) & \text{if } w_j(k) = a \in A^M \text{ (} a \text{ is a maintenance project)} \\ f_j^i(\underline{\theta}_j(k)) & \text{if } w_j(k) = a \in A^I \text{ (} a \text{ is an investment project)} \end{cases} \quad (3.10)$$

$$\overline{x}_j(k+1) = \begin{cases} f_j^0(\overline{x}_j(k)) & \text{if } w_j(k) = a_0 \in A^0 \text{ (no maintenance)} \\ f_j^m(\overline{x}_j(k), \overline{\theta}_j(k)) & \text{if } w_j(k) = a \in A^M \text{ (} a \text{ is a maintenance project)} \\ f_j^i(\overline{\theta}_j(k)) & \text{if } w_j(k) = a \in A^I \text{ (} a \text{ is an investment project)} \end{cases} \quad (3.11)$$

$$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p - 1\}.$$

Now the lower and upper bound of the objective functions are given by:

$$\underline{J}(\underline{x}, \gamma, \eta, \lambda, u, w, \underline{\theta}) = [\underline{J}_1(\underline{x}, \gamma, \eta, \lambda, u, w, \underline{\theta}), \dots, \underline{J}_m(\underline{x}, \gamma, \eta, \lambda, u, w, \underline{\theta})]^T \quad (3.12)$$

$$\overline{J}(\overline{x}, \gamma, \eta, \lambda, u, w, \overline{\theta}) = [\overline{J}_1(\overline{x}, \gamma, \eta, \lambda, u, w, \overline{\theta}), \dots, \overline{J}_m(\overline{x}, \gamma, \eta, \lambda, u, w, \overline{\theta})]^T. \quad (3.13)$$

Furthermore, additional constraints can be posed for the lower and upper bounds of the states:

$$g(x(k), \underline{x}(k), \overline{x}(k), \gamma(k), \eta(k), \lambda(k), u(k), w(k), \theta(k), \underline{\theta}(k), \overline{\theta}(k)) \leq 0. \quad (3.14)$$

The defined lower and upper bounds of the objectives and constraints can be used in different ways. The most simple way of including these realizations is to optimize only the expected values of the objective functions, and use the lower and upper bounds in the evaluation of the Pareto

optimal or non-dominated solutions. This will provide quick and simple insight into the uncertain nature of the investment projects. A computationally more expensive way is to optimize three separate optimization problems. Solutions that are Pareto optimal or non-dominated for each of the optimization problems is considered a robust choice with respect to the uncertain information of the investment alternatives. It is possible that the intersection of the non-dominated solution sets is empty. In that case, some additional performance indicators are needed to compare the effects of a certain project portfolio on the different scenarios for the objective values.

3.2. Railway network representation

The application of investment project portfolio optimization that is considered in this thesis is the selection and planning of investment projects in railway infrastructure networks. The optimization problem for project portfolio optimization in railway infrastructure networks will be presented in this section, following the general formulation as described in the previous section. An overview of the notation of all sets, indices, parameters, and variables is presented at the end of this section.

Prediction horizon

The investment project portfolio optimization problem in railway infrastructure networks can be considered for different levels. Due to the long lifespan of assets in the railway infrastructure, for a high-level point of view typically a prediction horizon of 20 years will be considered, and the planning horizon could be for example 1 year. For a detailed planning, time periods could be for example 1 hour. Then a prediction horizon should be only a couple of days.

Assets and states

Railway organizations store information on all assets and components in the railway infrastructure network. Some static information on the assets is stored, which is needed in the investment project portfolio optimization problem. Examples of the static data used in railway networks are the asset type t_j , the railway segment in which the asset is located, the start and end location of the asset, the length of the asset l_j , the neighbour assets C_j^x , and the installation date (if installed). The dynamic information on the assets are given by the states:

$$x_j(k) = \begin{bmatrix} x_j^{\text{con}}(k) \\ x_j^e(k) \\ x_j^o(k) \\ x_j^r(k) \\ x_j^s(k) \\ x_j^p(k) \\ x_j^\phi(k) \end{bmatrix} \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p\}, \quad (3.15)$$

where $x_j^e(k)$ is the amount of operational expenses (OPEX), $x_j^o(k) \in [0, 100]$ the percentage of outage, and $x_j^r(k) \in [0, 1000]$ the risk score, all for the j^{th} asset in time period k . For assets with the asset type 'track', some additional states are defined. The maximum speed at a track segment $x_j^s(k)$ (km/h) and the expected number of passengers per time period $x_j^p(k)$ are used to compute the capacity of a railway infrastructure network for passenger transport. The capacity of a track segment is given by $x_j^\phi(k)$. For assets of other types than asset type *track*, the speed, passengers, and capacity are 0 at all times.

In this formulation, $x_j(k)$ could also represent the state of a group of assets. These groups could

for example be based on asset type or age. This enables a high-level view, and could be used if the set of assets is very large, which is often the case in railway infrastructure networks.

For assets in railway networks there exist many advanced deterioration models (Sadeghi et al., 2010) that can be used to predict the condition. It is highly important for the results of the optimization problem that the predictions are as accurate as possible. However, research into the best deterioration models is out of scope for this thesis. For sake of simplicity, the deterioration of the assets is considered linear. For each asset type, a deterioration rate δ_t is provided. The condition of the j^{th} asset, with asset type t_j , in time period $k + 1$ is then given by:

$$x_j^{\text{con}}(k+1) = \max(x_j^{\text{con}}(k) - \delta_{t_j}, 0) \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p - 1\}. \quad (3.16)$$

Furthermore, the assumption is made that the operational expenses, the percentage of outage, the risk score, and the maximum speed have an affine relationship with the condition of the asset. The expected number of passengers is linearly depending on the speed of the asset, so indirectly also on the condition of the asset. For each asset j and each characteristic $h \in \{e, o, r, s, p\}$, a basic characteristic value $b_{j,h}$ is defined as the characteristic value when the asset is in full condition. The relations for the auxiliary variables e, o, r, s, p are defined as follows:

$$x_j^h(k) = b_{j,h} \cdot (\beta_{t_j,h} + \alpha_{t_j,h} \cdot x_j^{\text{con}}(k)) \quad h \in \{e, o, r, s\}, \quad (3.17)$$

$$x_j^p(k) = b_{j,p} \cdot (\alpha_{t_j,p} \cdot x_j^s(k)), \quad \forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p\}, \quad (3.18)$$

where the parameters $\alpha_{t,h}, \beta_{t,h}$ are the slope and the intercept of the relations respectively, defined for each asset type t and each characteristic $h \in \{e, o, r, s, p\}$.

Investment alternatives

The set of all investment alternatives is given by A . Each alternative $a \in A$, is either an alternative without (maintenance) actions $a \in A^0$, a maintenance action $a \in A^M$, or an investment project or replacement action $a \in A^I$. Alternatives in the set $A_j \subseteq A, A_j = \{a_0, \dots, a_{N_j}\}$ have effect on the j^{th} asset. The cardinality of this set is $N_j + 1$.

For each alternative a , the costs c_a , duration d_a , and man hours needed m_a are specified. The effect of an alternative on the concerning assets is given by $e_{a,\text{con}}$ and $e_{a,h}$ for $h \in \{e, o, r, s\}$. The effect starts as soon as the execution of the alternative ends. The definition of u, w, γ, η , and λ is the same as in Section 3.1. The states of the assets are given by:

$$x_j^{\text{con}}(k+1) = \begin{cases} \max(x_j^{\text{con}}(k) - \delta_{t_j}, 0) & w_j(k) = a_0 \in A^0 \\ \min(\max(x_j^{\text{con}}(k) - \delta_{t_j}, 0) + e_{a,\text{con}}, 1) & w_j(k) = a \in A^M \\ e_{a,\text{con}} & w_j(k) = a \in A^I \end{cases} \quad (3.19)$$

$$x_j^h(k+1) = \begin{cases} b_{j,h} \cdot (\beta_{t_j,h} + \alpha_{t_j,h} \cdot \max(x_j^{\text{con}}(k) - \delta_{t_j}, 0)) & w_j(k) = a_0 \in A^0 \\ b_{j,h} \cdot (\beta_{t_j,h} + \alpha_{t_j,h} \cdot (\min(\max(x_j^{\text{con}}(k) - \delta_{t_j}, 0) + e_{a,\text{con}}, 1))) & w_j(k) = a \in A^M \\ e_{a,h} \cdot (\beta_{t_j,h} + \alpha_{t_j,h} \cdot e_{a,\text{con}}) & w_j(k) = a \in A^I \end{cases} \quad (3.20)$$

$$x_j^p(k+1) = \begin{cases} b_{j,p} \cdot (\alpha_{t_j,p} \cdot x_j^s(k)) & w_j(k) = a_0 \in A^0 \\ b_{j,p} \cdot (\alpha_{t_j,p} \cdot (x_j^s(k) + e_{a,s})) & w_j(k) = a \in A^M \\ e_{a,p} \cdot (\alpha_{t_j,p} \cdot e_{a,s}) & w_j(k) = a \in A^I \end{cases} \quad (3.21)$$

$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p - 1\}, \forall h \in \{e, o, r, s\}.$

Constraints

For the investment project portfolio optimization problem, many constraints can be posed, that are mostly organization specific. In this general railway network representation, only a few of those are taken into account.

First, a constraint is posed on the amount of man hours, since there are a limited amount of workers to execute the projects. The maximum amount of man hours that can be scheduled per time period is given by $m^{\max}$. For each time period, the amount of scheduled man hours should be less than, or equal to $m^{\max}$:

$$g_1(x(k), \gamma(k), \eta(k), \lambda(k), u(k), w(k), \theta(k)) = \sum_{a \in A} \eta_a(k) \cdot m_a - m^{\max} \leq 0 \quad (3.22)$$

$$\forall k \in \{1, \dots, N_p\}.$$

Next, a local constraint could be considered for the states of the assets, for example to meet the safety requirements of the railway organization. The condition of each asset should be at least $x^{\text{con}, \min}$, in each time period:

$$g_{1+j}(x(k), \gamma(k), \eta(k), \lambda(k), u(k), w(k), \theta(k)) = x^{\text{con}, \min} - x_j^{\text{con}}(k) \leq 0 \quad (3.23)$$

$$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p\}.$$

Lastly, a constraint on the scheduling of the alternatives is given by a minimal time period in which an alternative could start. For all alternatives $a_q \in A \setminus a_0$, the period in which it starts should not be before the first possible start period $k_{a, \min}$.

$$g_{1+n+q}(x(k), \gamma(k), \eta(k), \lambda(k), u(k), w(k), \theta(k)) = \gamma_{a_q}(k) \cdot (k - k_{a, \min}) \leq 0 \quad (3.24)$$

$$\forall k \in \{1, \dots, N_p\}.$$

Objectives

The set of feasible solutions is bounded by the constraints. Now the goal is to find the best possible solutions in this set. In this goal, the *best possible solutions* are those that score best according to the multiple objectives, so the Pareto optimal solutions. For the general railway network representation, 4 objectives are defined to be most suitable: costs, condition, capacity, and risk.

The costs of the investment project portfolio for the railway network consist of two parts. The costs for the executed maintenance and investment projects, and the operational costs needed for all the assets in the network. The first objective is thus to minimize:

$$J_1(x, \gamma, \eta, \lambda, u, w, \theta) = \sum_{k \in \{1, \dots, N_p\}} \left(\sum_{a \in A} (c_a \cdot \gamma_a(k)) + \sum_{j \in C^\mu} (x_j^e(k)) \right), \quad (3.25)$$

where C^μ is the set of assets that are currently in use, so for which the installation date is in the past. Next to minimizing the costs of the investment project portfolio, an important goal is to maximize the condition of the railway network. For sake of simplicity, all the objectives in the optimization problem are to be minimized. Therefore, the objective function for the capacity of the network is included with a minus sign. The condition of the railway network is considered as the average condition of an asset in one time period. For the investment project portfolio, the negative version of the average condition is given by:

$$J_2(x, \gamma, \eta, \lambda, u, w, \theta) = - \frac{\sum_{k \in \{1, \dots, N_p\}} \sum_{j \in C^\mu} \left(x_j^{\text{con}}(k) \right)}{n \cdot N_p}. \quad (3.26)$$

A railway organization can make more profit if the capacity of the railway network is higher. Therefore, another goal of the organization is to maximize the capacity of the network. This could be achieved for example by expanding the network, or by maintaining a good quality network. The capacity of a railway network can be fully utilized if the condition is high. In this railway network representation, the capacity is defined for passenger transport. The capacity is measured in *passenger kilometers*. Because passengers are more likely to travel at a certain railway segment if that segment has good connections, the capacity of a track type asset j in time period k is given by:

$$\tilde{x}_j^p(k) = \left(x_j^p(k) + \sum_{j' \in C_j^x} \left(x_{j'}^p(k) \cdot l_{j'} \right) \right) \quad (3.27)$$

$$x_j^\phi(k) = \tilde{x}_j^p(k) \cdot l_j, \quad \forall j \in C^\mu \cap C^t, \forall k \in \{1, \dots, N_p\}, \quad (3.28)$$

where $\tilde{x}_j^p$ represents a shadow variable to prevent a loop in the definition of the expected number of passengers of neighbouring assets. The negative average amount of kilometers traveled by passengers in one time period is given by:

$$J_3(x, \gamma, \eta, \lambda, u, w, \theta) = - \frac{\sum_{k \in \{1, \dots, N_p\}} \left(x_j^\phi(k) \right)}{N_p}, \quad (3.29)$$

where the minus sign is inserted to invert the objective function to one that should be minimized. Lastly, an objective is defined to minimize the risk profile of the assets. The average risk score of an asset in a time period is to be minimized:

$$J_4(x, \gamma, \eta, \lambda, u, w, \theta) = \frac{\sum_{k \in \{1, \dots, N_p\}} \sum_{j \in C^\mu} \left(x_j^r(k) \right)}{n \cdot N_p}. \quad (3.30)$$

Lower and upper bounds

The optimization problem for the railway network representation defined so far, is the optimization problem with expected values. For the effect of the investment alternatives on the assets, lower bounds $\underline{e}_a$ and upper bounds $\overline{e}_a$ are considered. Also lower and upper bounds for the costs, duration, and man hours needed for the investment alternatives are included. The lower and upper bounds for the man hours can be used to pose additional constraints that could be seen as the maximum amount of man hours that can be deployed, including over hours.

All the lower and upper bounds of the asset states are defined as explained in the previous section. With the lower and upper bounds for the effects of the investment alternatives, the lower and upper bounds for the states of the assets are given by:

$$\underline{x}_j^{\text{con}}(k+1) = \begin{cases} \max\left(\underline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) & w_j(k) = a_0 \in A^0 \\ \min\left(\max\left(\underline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) + \underline{e}_{a,\text{con}}, 1\right) & w_j(k) = a \in A^M \\ \underline{e}_{a,\text{con}} & w_j(k) = a \in A^I \end{cases} \quad (3.31)$$

$$\underline{x}_j^h(k+1) = \begin{cases} b_{j,h} \cdot \left(\beta_{t,h} + \alpha_{t,h} \cdot \max\left(\underline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) \right) & w_j(k) = a_0 \in A^0 \\ b_{j,h} \cdot \left(\beta_{t,h} + \alpha_{t,h} \cdot \left(\min\left(\max\left(\underline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) + \underline{e}_{a,\text{con}}, 1\right) \right) \right) & w_j(k) = a \in A^M \\ \underline{e}_{a,h} \cdot \left(\beta_{t,h} + \alpha_{t,h} \cdot \underline{e}_{a,\text{con}} \right) & w_j(k) = a \in A^I \end{cases} \quad (3.32)$$

$$\underline{x}_j^p(k+1) = \begin{cases} b_{j,p} \cdot \left(\alpha_{t,p} \cdot \underline{x}_j^s(k) \right) & w_j(k) = a_0 \in A^0 \\ b_{j,p} \cdot \left(\alpha_{t,p} \cdot \left(\underline{x}_j^s(k) + \underline{e}_{a,s} \right) \right) & w_j(k) = a \in A^M \\ \underline{e}_{a,p} \cdot \left(\alpha_{t,p} \cdot \underline{e}_{a,s} \right) & w_j(k) = a \in A^I \end{cases} \quad (3.33)$$

$$\overline{x}_j^{\text{con}}(k+1) = \begin{cases} \max\left(\overline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) & w_j(k) = a_0 \in A^0 \\ \min\left(\max\left(\overline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) + \overline{e}_{a,\text{con}}, 1\right) & w_j(k) = a \in A^M \\ \overline{e}_{a,\text{con}} & w_j(k) = a \in A^I \end{cases} \quad (3.34)$$

$$\overline{x}_j^h(k+1) = \begin{cases} b_{j,h} \cdot \left(\beta_{t,h} + \alpha_{t,h} \cdot \max\left(\overline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) \right) & w_j(k) = a_0 \in A^0 \\ b_{j,h} \cdot \left(\beta_{t,h} + \alpha_{t,h} \cdot \left(\min\left(\max\left(\overline{x}_j^{\text{con}}(k) - \delta_{t_j}, 0\right) + \overline{e}_{a,\text{con}}, 1\right) \right) \right) & w_j(k) = a \in A^M \\ \overline{e}_{a,h} \cdot \left(\beta_{t,h} + \alpha_{t,h} \cdot \overline{e}_{a,\text{con}} \right) & w_j(k) = a \in A^I \end{cases} \quad (3.35)$$

$$\overline{x}_j^p(k+1) = \begin{cases} b_{j,p} \cdot \left(\alpha_{t,p} \cdot \overline{x}_j^s(k) \right) & w_j(k) = a_0 \in A^0 \\ b_{j,p} \cdot \left(\alpha_{t,p} \cdot \left(\overline{x}_j^s(k) + \overline{e}_{a,s} \right) \right) & w_j(k) = a \in A^M \\ \overline{e}_{a,p} \cdot \left(\alpha_{t,p} \cdot \overline{e}_{a,s} \right) & w_j(k) = a \in A^I \end{cases} \quad (3.36)$$

$$\forall j \in \{1, \dots, n\}, \forall k \in \{1, \dots, N_p - 1\}, \forall h \in \{e, o, r, s\}.$$

The lower and upper bounds for the costs $\underline{c}_a$ and $\overline{c}_a$ are used to compute the upper and lower bound of the costs objective. Furthermore, the lower and upper bounds for the states are used to compute the lower and upper bounds for the objective values:

$$\underline{J}_1(\underline{x}, \gamma, \eta, \lambda, u, w, \underline{\theta}) = \sum_{k \in \{1, \dots, N_p\}} \left(\sum_{a \in A} \left(\underline{c}_a \cdot \gamma_a(k) \right) + \sum_{j \in C^\mu} \left(\underline{x}_j^e(k) \right) \right) \quad (3.37)$$

$$\overline{J}_1(\overline{x}, \gamma, \eta, \lambda, u, w, \overline{\theta}) = \sum_{k \in \{1, \dots, N_p\}} \left(\sum_{a \in A} \left(\overline{c}_a \cdot \gamma_a(k) \right) + \sum_{j \in C^\mu} \left(\overline{x}_j^e(k) \right) \right) \quad (3.38)$$

$$\underline{J}_2(\underline{x}, \gamma, \eta, \lambda, u, w, \underline{\theta}) = - \frac{\sum_{k \in \{1, \dots, N_p\}} \sum_{j \in C^\mu} \left(\underline{x}_j^{\text{con}}(k) \right)}{n \cdot N_p} \quad (3.39)$$

$$\overline{J}_2(\overline{x}, \gamma, \eta, \lambda, u, w, \overline{\theta}) = - \frac{\sum_{k \in \{1, \dots, N_p\}} \sum_{j \in C^\mu} \left(\overline{x}_j^{\text{con}}(k) \right)}{n \cdot N_p} \quad (3.40)$$

$$\underline{J}_3(\underline{x}, \gamma, \eta, \lambda, u, w, \underline{\theta}) = - \sum_{k \in \{1, \dots, N_p\}} \left(\underline{x}_j^\phi(k) \right) \quad (3.41)$$

$$\bar{J}_3(\bar{x}, \gamma, \eta, \lambda, u, w, \bar{\theta}) = - \sum_{k \in \{1, \dots, N_p\}} \left(\bar{x}_j^\phi(k) \right) \quad (3.42)$$

$$\underline{J}_4(\underline{x}, \gamma, \eta, \lambda, u, w, \underline{\theta}) = \frac{\sum_{k \in \{1, \dots, N_p\}} \sum_{j \in C^\mu} \left(\underline{x}_j^r(k) \right)}{n \cdot N_p} \quad (3.43)$$

$$\bar{J}_4(\bar{x}, \gamma, \eta, \lambda, u, w, \bar{\theta}) = \frac{\sum_{k \in \{1, \dots, N_p\}} \sum_{j \in C^\mu} \left(\bar{x}_j^r(k) \right)}{n \cdot N_p} \quad (3.44)$$

The lower and upper bounds for the duration of the investment alternatives can be used in the multi-objective optimization problems with the lower and upper bounds of the objective functions respectively. In this way, the lower bound four the objectives is really the best case scenario, and the upper bound the worst case scenario. Projects that are analyzed to be non-dominated in all three scenarios are considered as robust choices with respect to the uncertainty in the effects of the investment alternatives.

Sets and indices

	k	Time periods, $\{1, \dots, N_p\}$
C	j	Assets, $\{1, \dots, n\}$
$C^t \subseteq C$		Assets with asset type track
$C_j^x \subseteq C$		Neighbouring track type assets of j^{th} asset, $j \in C^t$
$C^\mu \subseteq C$		Assets that are currently in use
T	t	Asset types, $\{\text{track}, \text{switch}, \text{signalling}\}$
H	h	Asset characteristics, $\{\text{OPEX}, \text{outage}, \text{risk}, \text{speed}\}$
S	s	Railway segments
A	a	Investment alternatives, $\{a_0, \dots, a_N\}$
$A_j \subseteq A$		Investment alternatives for j^{th} asset
$A^0 \subseteq A$		No (maintenance) actions
$A^M \subseteq A$		Maintenance (repair) projects
$A^I \subseteq A$		Replacement or renewal investment projects

Parameters

N_c	Control horizon
N_p	Prediction horizon
c_a	Costs of alternative a
$\underline{c}_a$	Lower bound for costs of alternative a
$\bar{c}_a$	Upper bound for costs of alternative a
d_a	Duration of alternative a in periods
$\underline{d}_a$	Lower bound for duration of alternative a in periods
$\bar{d}_a$	Upper bound for duration of alternative a in periods
$e_{a,h}$	Effect of alternative a on asset characteristic $h \in \{e, o, r, s\}$
$\underline{e}_{a,h}$	Lower bound for effect of alternative a on asset characteristic $h \in \{e, o, r, s\}$
$\bar{e}_{a,h}$	Upper bound for effect of alternative a on asset characteristic $h \in \{e, o, r, s\}$
m_a	Man hours needed for alternative a per period
$\underline{m}_a$	Lower bound for man hours needed for alternative a per period
$\bar{m}_a$	Upper bound for man hours needed for alternative a per period
$m^{\max}$	Maximum amount of man hours per period
$\bar{m}^{\max}$	Maximum amount of man hours per period, including overtime

l_j	Length of asset j , in km
t_j	Type of asset j
s_j	Railway segment of asset j
μ_j	Binary parameter, indicating whether asset j is currently in use
$b_{j,h}$	Basic value of characteristic $h \in \{e, o, r, s, p\}$ for asset j
δ_t	Deterioration rate for asset type t
$\alpha_{t,h}$	Slope for condition in multiplier definition for asset type t and characteristic $h \in \{e, o, r, s, p\}$
$\beta_{t,h}$	Intercept multiplier definition for asset type t and characteristic $h \in \{e, o, r, s\}$

Variables

$x_j^{\text{con}}(k)$	The condition of asset j in time period k
$\underline{x}_j^{\text{con}}(k)$	Lower bound for the condition of asset j in time period k
$\overline{x}_j^{\text{con}}(k)$	Upper bound for the condition of asset j in time period k
$x_j^e(k)$	The amount of operational expenses for asset j in time period k
$\underline{x}_j^e(k)$	Lower bound for the amount of operational expenses for asset j in time period k
$\overline{x}_j^e(k)$	Upper bound for the amount of operational expenses for asset j in time period k
$x_j^o(k)$	The percentage of outage of asset j in time period k
$\underline{x}_j^o(k)$	Lower bound for the percentage of outage of asset j in time period k
$\overline{x}_j^o(k)$	Upper bound for the percentage of outage of asset j in time period k
$x_j^r(k)$	The risk score of asset j in time period k
$\underline{x}_j^r(k)$	Lower bound for the risk score of asset j in time period k
$\overline{x}_j^r(k)$	Upper bound for the risk score of asset j in time period k
$x_j^s(k)$	The maximum speed for asset $j \in C^t$ in time period k
$\underline{x}_j^s(k)$	Lower bound for the maximum speed for asset $j \in C^t$ in time period k
$\overline{x}_j^s(k)$	Upper bound for the maximum speed for asset $j \in C^t$ in time period k
$x_j^p(k)$	The number of passengers for asset $j \in C^t$ in time period k
$\underline{x}_j^p(k)$	Lower bound for the number of passengers for asset $j \in C^t$ in time period k
$\overline{x}_j^p(k)$	Upper bound for the number of passengers for asset $j \in C^t$ in time period k
$\tilde{x}_j^p(k)$	The number of passengers for asset $j \in C^t$ in time period k , without additional passengers due to connectivity advantages
$\underline{\tilde{x}}_j^p(k)$	Lower bound for the number of passengers for asset $j \in C^t$ in time period k , without additional passengers due to connectivity advantages
$\overline{\tilde{x}}_j^p(k)$	Upper bound for the number of passengers for asset $j \in C^t$ in time period k , without additional passengers due to connectivity advantages
$x_j^\phi(k)$	The capacity asset $j \in C^t$ in time period k
$\underline{x}_j^\phi(k)$	Lower bound for the capacity asset $j \in C^t$ in time period k
$\overline{x}_j^\phi(k)$	Upper bound for the capacity asset $j \in C^t$ in time period k
$u_j(k)$	The alternative a that starts for asset j in time period k
$w_j(k)$	The alternative a that ends for asset j in time period k
$\gamma_a(k)$	Binary variable that indicates whether alternative a starts in time period k
$\eta_a(k)$	Binary variable that indicates whether alternative a is being executed in time period k
$\lambda_a(k)$	Binary variable that indicates whether alternative a ends in time period k

3.3. Multi-objective optimization methods

Many multi-objective optimization methods are presented in the literature. The selected optimization method should be able to solve problem formulation, but other than that, the selection of an optimization method is mostly based on preferences. In project portfolio optimization, the output of the optimization is used in the decision-making process. Therefore, an a posteriori method is preferred. Decision makers are often interested in optimization methods that are fast, such that they are able to run the optimization for different scenarios. Because of the high impact of the solutions that are provided by the optimization method, decision makers want to be sure that the selected solution is optimal. Because the investment project portfolio problem in infrastructure networks is a combinatorial optimization problem that often considers many possible projects, there will be a trade-off between the computation time and the solution quality. Two multi-objective optimization methods are selected to solve the investment project portfolio problem as defined in the previous sections. Genetic algorithms are known to be fast, but because it is a metaheuristic search technique, no information can be provided whether the obtained solutions are optimal. The repeated ε -constraint algorithm is a method in which multiple single-objective optimization methods are solved. For these single-objective problems, optimality can often be guaranteed. The general idea of the algorithms is explained in more detail, specified for the investment project portfolio optimization problem. The process of the algorithms is clarified with some pseudocode.

Repeated ε -constraint algorithm

The main idea of the repeated ε -constraint algorithm is to solve many single-objective problems. One of the objectives in the optimization problem is optimized, while for the others, the objective value is bounded by a constraint. Each of the objectives can be chosen to be the main objective. To determine appropriate boundaries for the other objectives, first the so called *extreme points* are generated by optimizing the individual objectives without ε -bounds. Now that the ranges for each of the objective functions are determined, these ranges are divided into g intervals. For each objective function region, a different amount of intervals could be used. For sake of simplicity, but without loss of generality, in the proposed algorithm the same amount of intervals is used for all bounded objectives. All the ε boundaries start at the maximum objective values, and move in different for-loops to the minimum objective values. The pseudocode for the repeated ε -constraint algorithm applied to the case study is presented here:

```

for all objectives do
    Minimize objective  $p$ 
    Save all objective values
end for
Determine minimum and maximum objective values
for all remaining objectives  $p$  do
    Set  $\varepsilon_p = \max(J_p)$ 
    for index- $p = 1:g$  do
        for all remaining objectives  $p'$  do
            Set  $\varepsilon_{p'} = \max(J_{p'})$ 
            for index- $p' = 1:g$  do
                 $\vdots$ 
                Minimize objective  $J_1$ 
                s.t.  $J_p \leq \varepsilon_p, \forall p \in \{2, \dots, m\}$ 
                 $g \leq 0$ .
            end for
        end for
        Save solution
    end for
end for

```

```

        Set  $\varepsilon_p = \max(J_p) - \text{index} - p \cdot \frac{(\max(J_p) - \min(J_p))}{g}$ 
        :
    end for
end for
end for
end for

```

In this algorithm, the single-objective optimization problems are mixed integer linear programming problems (MILP), or mixed integer non-linear programming problems (MINLP). In case that a combination of boundaries ε_p creates an infeasible problem, the 'solution' is not used for further analysis. This also indicates that other problems in which one or more of the boundaries is even more tight, will also be infeasible. To save computation time, one could use this in an exit strategy, as proposed in Mavrotas (2009). In this way, these subsequent infeasible problems will be skipped.

For optimization problems with a very large solution set, it might not be sufficient to use an exit strategy for subsequent infeasible problems in terms of computation time. Stopping criteria could be used to terminate an optimization statement. For high-level problems, the distance to the Pareto front could be less important. In that case, for a MILP, one could use a relative stopping criterion that terminates the optimization if the current best solution is within a specified region from the current best Linear Programming (LP) bound. Another option is to set a maximum optimization time, and let the optimization terminate in case the solving takes longer than the specified time. In both cases, the real optimal solution might not have been found. This means that using stopping criteria has effect on the performance criteria regarding the distance to the real Pareto front. This will thus only be used if reduced computation time is more important than the distance to the Pareto set. An overview of performance indicators is provided in Section 3.4.

Multi-objective genetic algorithm

The widely used genetic algorithm, NSGA II, for multi-objective optimization problems is used to search for the Pareto front with an entirely different approach in comparison with the repeated ε -constraint algorithm. The search strategy as presented in Section 2.2.3 has some settings to tune the genetic algorithm for the specific optimization problem. The NSGA II is a non-sorted genetic algorithm, proposed in Deb et al. (2002).

The improved non-dominated sorting algorithm (NSGA II) owes its name to the fitness assignment. The fitness of each of the solutions in the solution set is based on a non-dominated sorting. This non-dominated sorting means that Pareto optimal (non-dominated) solutions get a better rank than solutions that are dominated by other solutions. The NSGA II can be used with different types of encoding schemes. Since the investment project portfolio optimization problem is a combinatorial optimization problem, a binary string is used for the chromosomes of the individuals in the population.

The pseudocode of the main loop of the implemented algorithm is presented here:

```

Initialize random population
Compute the objective values of individuals in the initial population
Compute the error of individuals in the initial population
Compute non-dominated sorting
for each generation do
    Use tournament selection to select the parent population
    Use crossover and mutation to create the offspring population

```

Compute the objective values and the error of individuals in the offspring population
 Compute non-dominated sorting of combined parent and offspring population
 Select the best individuals for the next generation
end for

The tournament selection is implemented with a tournament size of two. For crossover, a *two-point crossover* is used, for which two random points in the binary string are selected. The two children will exist of the binary strings of the parents, with the part between the two points exchanged. The non-dominated sorting approach is adopted from (Deb et al., 2002). The approach makes use of the partial order $<_n$ for non-domination.

Definition 4 (Partial order for non-domination). *Let b_{rank} , and b'_{rank} be the non-domination rank of individuals b and b' , and $b_{\text{cdistance}}$, $b'_{\text{cdistance}}$ their crowding distance indicator. The partial order $<_n$ for non-domination of individual b with respect to individual b' is defined by:*

$$b <_n b' \text{ if } (b_{\text{rank}} < b'_{\text{rank}}) \text{ or } (b_{\text{rank}} = b'_{\text{rank}} \text{ and } b_{\text{cdistance}} > b'_{\text{cdistance}}) \quad (3.45)$$

Now the pseudocode for the non-dominated sorting approach is presented:

```

for all feasible solutions in the population,  $b \in P(t)$  do
  Initialize set of solutions  $S_b$  that is dominated by  $b$ 
  Initialize the number of solutions that dominate  $b$ ,  $n(b)$ 
  for each other feasible solution in the population,  $b' \in P(t)$  do
    if  $b <_n b'$ ,  $b$  dominates  $b'$  then
       $S_b = S_b \cup b'$ , add the solution  $b'$  to the set of solutions dominated by  $b$ 
    else if  $b' <_n b$ ,  $b'$  dominates  $b$  then
       $n(b) = n(b) + 1$ , increase the number of solutions that dominate  $b$ 
    end if
  end for
  rank = 1
  Add all individuals  $b$  with  $n(b) = 0$  to the first front fr(1)
  while front fr(rank) is not empty do
    for all  $b \in \text{fr}(\text{rank})$  do
       $b_{\text{rank}} = \text{rank}$ 
      for all  $b'$  that are dominated by  $b$  do
         $n(b) = n(b) - 1$ 
      end for
      rank = rank + 1
    end for
    Add all individuals  $b$  with  $n(b) = 0$  to the front fr(rank)
  end while
end for
for all infeasible solutions  $b \in P(t)$  do
  Add all individuals to the front fr(rank)
end for
Sort population on rank

```

To clarify the process of how generations in NSGA II evolve, process of the algorithm to obtain the next parent population from the current parent population is visualized in Figure 3.1.

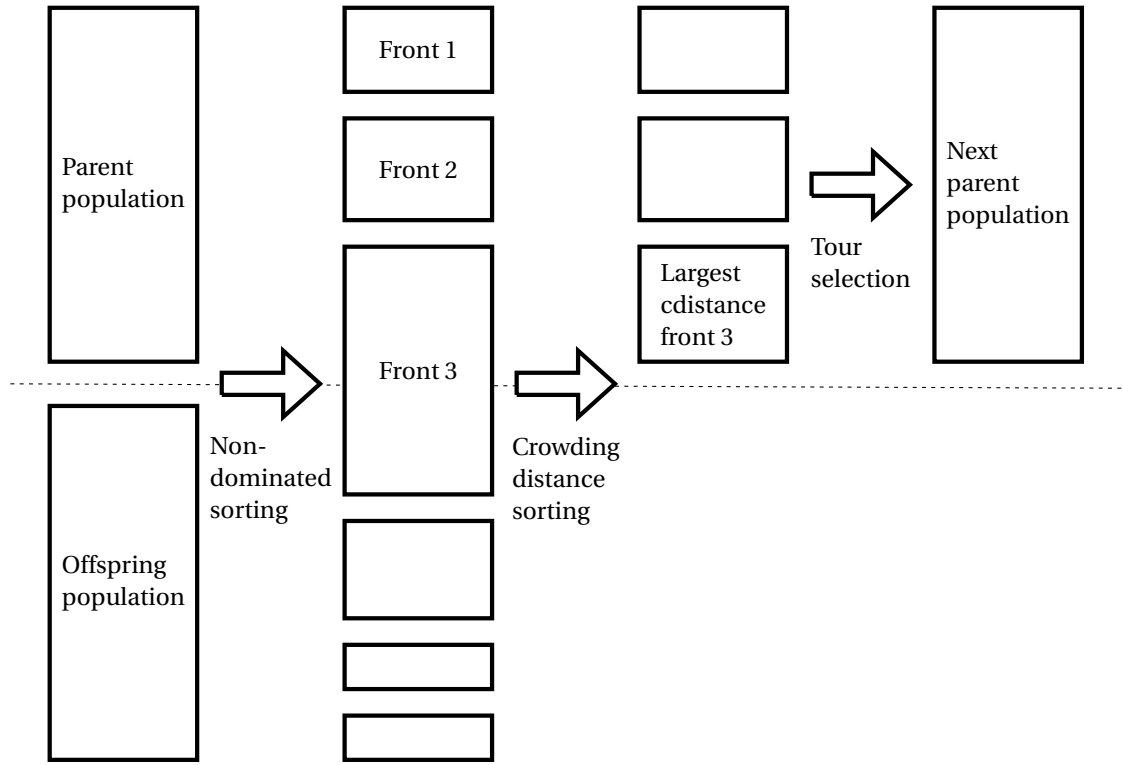


Figure 3.1: Visualization of how the next parent population is obtained from the combined parent and offspring population, adapted from Deb et al. (2002).

3.4. Performance indicators

To answer the research question 'Which solution techniques are best suited for the multi-objective investment project portfolio problem, regarding solution quality and computation time?', there is a need for a definition of *solution quality*. A number of possible performance indicators are discussed in Section 2.2.5. The selected performance indicators to compare the performance of the repeated ε -constraint algorithms and the NSGA II algorithms will be presented here. Each algorithm with unique settings will be tested multiple times to make sure the outcomes are reliable. The performance indicators will be presented either as the average of the runs, or the average and the standard deviation of the runs.

First, let S be the set of solutions obtained by a certain algorithm, which can be seen as an approximation set of the Pareto set $\mathcal{P}$. The real Pareto optimal solutions in $\mathcal{P}$ are not known in any problem. Therefore, for the case studies, a reference Pareto set $\mathcal{R}$ is used. This set $\mathcal{R}$ consists of all non-dominated solutions that exists in the union of all approximation sets S . This could be equal to the Pareto optimal set, $\mathcal{R} = \mathcal{P}$, but it could just as well be the case that real Pareto optimal solutions are not found by any of the algorithms.

The average computation time that is needed to obtain the approximation set S is given by t_{run} . The number of (not necessarily unique) solutions in the solution set S is given by the cardinality $|S|$. Due to constraints in the problem, infeasible solutions could be present in the solution set. The number of feasible solutions in the solution set is given by $|S_{\mathcal{F}}|$. If the real Pareto set $\mathcal{P}$ is known, the average number of Pareto optimal solutions in the approximation set is noted by $|S_{\mathcal{P}}|$. In the case that $\mathcal{P}$ is not known, the average number of solutions that are in the union of the approximation set and the reference set $S \cap \mathcal{R}$ are noted by $|S_{\mathcal{R}}|$. The number of real Pareto optimal solutions, $|\mathcal{P}|$, or the number of non-dominated solutions, $|\mathcal{R}|$, is used to compute the average percentage of the reference set that is covered by a certain algorithm. This percentage is noted by

$cp_{\mathcal{P}}$ or $cp_{\mathcal{R}}$. For the coverage, the standard deviation (s.d.) is included for comparison as well. This indicates how constant the algorithm performs. Lastly, the *generational distance* as defined in Van Veldhuizen et al. (2000), is used to compute the average distance from the approximation sets S to the reference set $\mathcal{R}$. The generational distance is the average of the distance from each solution $s \in S$ to the nearest solution in the reference set, $r \in \mathcal{R}$, based on Euclidean distance. The generational distance of S to $\mathcal{R}$ is defined by:

$$G = \frac{\left(\sum_{i=1}^{|S|} \min_{r \in \mathcal{R}} \left(\sqrt{\sum_{p=1}^m (s_{i,p} - r_{i,p})^2} \right) \right)^{\frac{1}{2}}}{|S|}. \quad (3.46)$$

Since the different objective functions all could have a very different scale, the values for all objective functions p are normalized with respect to the minimum and maximum value of objective p that are present in the union of all solution sets. For the generational distance, the standard deviation is considered as well.

The selected performance indicators can be used when selecting an appropriate algorithm to solve a combinatorial multi-objective optimization method. It would be preferable for an algorithm to have a short computation time, a large number of solutions in the Pareto set or reference set, a high percentage of $\mathcal{P}$ or $\mathcal{R}$ coverage, and a small generational distance to the reference set or Pareto set. Since there might not exist an algorithm that gives optimal results for all performance indicators, the problem of selecting an algorithm is a multi-objective problem itself. A trade-off has to be made between the different performance indicators, based on the preferences of the user.

3.5. Visualization and decision support

To make sure that there is a connection between the investment project portfolio optimization approach and the user interface of a decision support tool, and that the combination really meets the needs of the users, a global design is proposed for the user interface of a decision support tool. The scatter plot matrix and the parallel coordinates plot as described in Section 2.3.2 are used to visualize the solutions of the investment project portfolio optimization problem. For 2 or 3 objectives, it would be sufficient to consider the scatter plot matrix of the solutions. In the case that 4 or more objectives are considered, both a scatter plot matrix and a parallel coordinates plot could be provided. A parallel coordinate plot is better to link the different objective values that belong to one solution, whereas the scatter plot matrix is better to get an overview of the shape of the non-dominated front. It is recommended that the selection of visualization methods for a decision support tool is done in cooperation with the final users.

For both visualization methods, it is difficult to distinguish the individual solutions. Therefore, in a decision support tool, it is strongly recommended to include filtering features. Decision makers are able to gain insight in the interval of objective values, after which they can select an objective for which a part of the interval contains objective values that are unacceptable or undesirable. Then, a filter could be used to filter out the unacceptable or undesirable solutions. In this way, the amount of visualized non-dominated solutions can be reduced, to obtain a clearer overview of the interesting results. Another option to reduce the amount of visualized solutions, is to filter for project portfolios that contain a specific solution that is desired in the portfolio. If the project that is used for this filter option is not selected in any of the non-dominated project portfolios, so with a core index of 0, the problem should be re-optimized while that project is forced to be in the project portfolio. This leads to a completely new non-dominated front, since the problem is adapted. The new non-dominated front will only contain solutions that were dominated in the

previous non-dominated set of solutions. However, it still might be a good solution for the organization due to for example social values of a certain project, while the social value is not explicitly taken into account as objective in the optimization problem.

Another feature that is recommended to include in a decision support tool for investment project portfolio optimization, is that decision makers should be able to examine the details of the non-dominated project portfolios. The details of a portfolio would include an overview of which projects are selected in the selected portfolio, with the planning. Also the objective values and their lower and upper bounds should be obtainable. It could happen that a decision maker detects a combination or planning of projects that is inconvenient or even not possible at all. This could for example occur if the constraints that the decision maker handles, are not considered in the problem formulation, or if the first possible start period of a project is not entered correctly. After concluding that a certain project portfolio is not feasible, either the input data should be adapted, or a constraint should be added. A scenario page is included to add or adjust constraint, or to force certain projects to be included in the solutions.

Since it is already difficult to distinguish individual solutions in a scatter plot matrix or parallel coordinates plot, it is not recommended that the lower and upper bounds of the objective values are visualized in a combined plot with the expected values. The lower and upper bounds should be visualized for a few selected solutions, or three separate plots should be provided for the lower bounds, the expected values, and the upper bounds respectively.

For obtaining detailed insight into the non-dominated solutions, interaction with the decision support tool is needed. Multiple pages, containing different information and visualizations, are used to do this. In the comparison of the results in this thesis, no interaction is possible. Therefore, most of the visualizations will be in a parallel coordinates plot. Distinctive appearances of the solutions are used to distinguish different algorithms, different settings, or different domination information. Solutions in a solution set that are Pareto optimal or selected in the non-dominated reference set $\mathcal{R}$, are visualized with solid markers and solid lines. Solutions that are dominated by other solutions are visualized with unfilled markers and dashed lines.

3.6. Summary

The mathematical formulation of the investment project portfolio optimization problem is presented for both general purposes, as for the railway infrastructure network representation. An extensive number of sets, indices, parameters, variables, and objective functions is introduced, along with some general relations.

The available information about the investment alternatives is often uncertain. For different realizations of the uncertainty, the set of non-dominated solutions might be completely different. Decision makers need to have insight in the possible realizations to be able to select a project portfolio that is also satisfactory for other realizations than the expected values. For the costs and benefits of investment alternatives, lower and upper bounds are introduced. The lower and upper bounds of the states of the assets and the final objective values are defined as well. Solving the multi-objective optimization problem for the three scenarios makes it possible for decision makers to make well-founded decisions, and gain insight into which project portfolios are robust choices.

The repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II are selected to solve the investment project portfolio optimization problem. The general approach of the different algorithms is explained, and some pseudocode is presented to get an idea of the structure of the algorithms.

A number of performance indicators are selected for comparison of the proposed algorithms. Examples are the computation time, the number of non-dominated solutions, the coverage of the reference set, and the generational distance.

Benchmarks & railway network case studies

In the previous chapters, the theory behind project portfolio optimization is discussed. In this chapter, the proposed formulation, optimization methods, and visualizations are put into practice. The multi-objective optimization methods as described in Chapter 3, the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II, are used to solve some simple benchmark problems, and finally also to solve the investment project portfolio optimization problem for railway infrastructure networks. The benchmark problems are used for quick insight into the differences of the proposed methods. Since the multi-objective benchmark problems are very simple, the real Pareto front is known, which makes the comparison of the performance of the methods more reliable. For some in-depth examination of the performance of the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II to solve the project portfolio optimization problem in practice, a railway network case for investment project portfolio optimization is introduced. The results obtained by the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II for the benchmark problems and the railway network case studies are presented and discussed in this chapter. The repeated ε -constraint algorithm is implemented in the optimization software AIMMS¹. Each of the single-objective problems is solved with CPLEX² in the MIP cases, and with the AIMMS Outer Approximation algorithm in combination with CPLEX in the MINLP case. The multi-objective genetic algorithm NSGA II is implemented in the mathematical software MATLAB³. All tests are executed on the same computer⁴.

4.1. Benchmark problem 1

The first benchmark problem is a convex problem, consisting of two objective functions of two integer variables $x_1 \in \{1, \dots, 32\}, x_2 \in \{1, \dots, 32\}$. The variable θ , indicates the uncertain nature of J_2 . The expected value of θ is 1. The two objective functions J_1, J_2 are given by:

$$J_1(x_1, x_2) = (x_1 - 10)^2 + (x_2 - 13)^2 \quad (4.1)$$

$$J_2(x_1, x_2, \theta) = \theta \cdot (2x_1 + 3x_2). \quad (4.2)$$

There are no additional constraints posed in the optimization problem, besides $1 \leq x_1 \leq 32, 1 \leq x_2 \leq 32$. The feasible set X consists of 1024 possible solutions (x_1, x_2) , of which 47 are Pareto optimal solutions in the Pareto optimal set $\mathcal{P}$. These Pareto optimal solutions are not equally spread in the objective space.

¹AIMMS, Version 4.37.4.1024 64 bit

²IBM ILOG CPLEX Optimization Studio, Version 12.6.3.0

³MathWorks MATLAB R2017a (9.2.0.556344) 64-bit

⁴Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz, 8.00GB RAM, 64 bit operating system.

One of the objective functions is considered to be subject to uncertainty, that is represented by θ . Let the lower and upper bounds of θ be given by $\underline{\theta} = 0.95$ and $\bar{\theta} = 1.05$. The lower and upper bounds for the objective function J_2 are then given by:

$$\underline{J}_2(x_1, x_2, \underline{\theta}) = 0.95 \cdot (2x_1 + 3x_2) \quad (4.3)$$

$$\bar{J}_2(x_1, x_2, \bar{\theta}) = 1.05 \cdot (2x_1 + 3x_2). \quad (4.4)$$

The Pareto optimal set $\mathcal{P}$ for the objective function J_2 with expected value $\theta = 1$ is shown in Figure 4.1b, in blue. The Pareto optimal sets for the lower bounds of J_2 are in green, and the upper bounds in red. It can be seen that for this problem, the Pareto optimal sets are equal. For $\underline{J}_2$, J_2 , and $\bar{J}_2$, the corresponding Pareto optimal front is shown in Figure 4.1a.

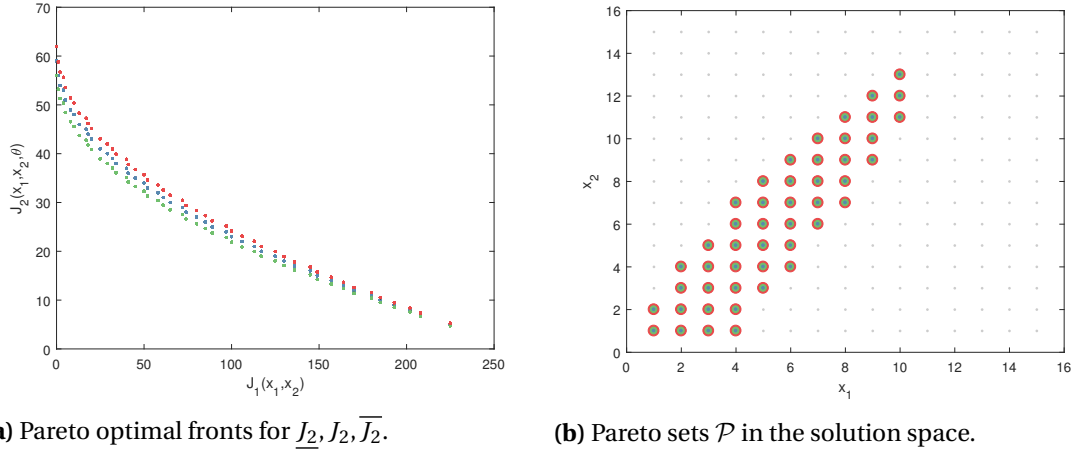


Figure 4.1: Pareto sets $\mathcal{P}$ and corresponding Pareto optimal fronts for benchmark problem 1.

4.1.1. Repeated ε -constraint algorithm

The first objective function J_1 is chosen as the main objective function, that is minimized in the single-objective optimization problems. The second objective function, J_2 , is bounded by ε_2 . The maximum value is $J_2 = 2 \cdot 32 + 3 \cdot 32 = 160$, and the minimum value is $J_2 = 2 \cdot 1 + 3 \cdot 1 = 5$. The objective value will always be integer due to the two integer variables. The most thorough way to search for the Pareto optimal set is to consider 155 single-objective optimization problems, in which ε_2 takes all integer values between 160 and 5. The output of the repeated ε -constraint algorithm, the solution set S , consists of 155 solutions. In this case, with only one ε -bound, all the single-objective optimization problems will be feasible, by definition of the ε_2 . For the objective values of J_2 the minimum and maximum are given by $\underline{J}_2 = 0.95 \cdot 5$ and $\underline{J}_2 = 0.95 \cdot 160$. Equally for $\bar{J}_2$, the minimum and maximum are given by $\bar{J}_2 = 0.95 \cdot 5$ and $\bar{J}_2 = 0.95 \cdot 160$. The number of possible values for the objective does not change in this problem for the lower or upper bound in comparison with the expected value. Therefore, using 155 bounds is still sufficient to find the entire Pareto optimal set, albeit with different values.

The number of bounds, and thus cardinality of the solution set $|S|$, could also be lower, but then there is no guarantee that all the Pareto optimal solutions will be included in the solution set. It does however save computation time. For this simple benchmark problem, this might not be necessary because the computation time for all 155 problems is only 0.84 seconds on average. Although the computation time of the repeated ε -constraint algorithm with 100 bounds is only 0.58 seconds on average, the performance in terms of percentage of the Pareto front that is found

is much lower. On average, only 42.55% of the Pareto front has been found with the 100 bounds, in contrast to 100% that has been found with 155 bounds. An overview of the performance measures can be found in Table 4.1.

$ S $	Algorithm	t_{run}	$ S_{\mathcal{F}} $	$ S_{\mathcal{P}} $	$\text{cp}_{\mathcal{P}}$	s.d. $\text{cp}_{\mathcal{P}}$	G	s.d. G
100	BM1- ε -1	0.58 s	100	98.0	42.55%	0.00	0.000262	0.000000
155	BM1- ε -2	0.84 s	155	152.0	100.00%	0.00	0.000164	0.000000

Table 4.1: Performance indicators for different settings of the repeated ε -constraint algorithm, for the convex benchmark problem.

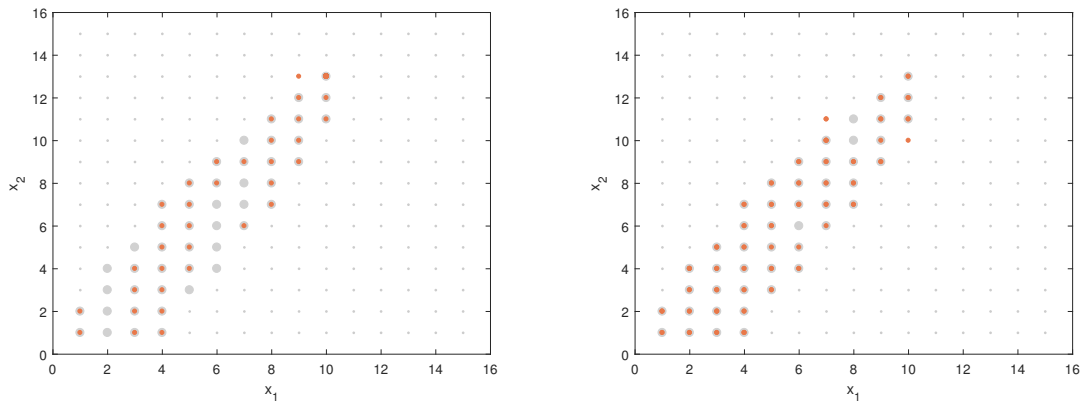
The standard deviations of the coverage and the generational distance are 0 for the repeated ε -constraint algorithm, because the same search strategy is used in every run. Only the computation time differs for the runs.

4.1.2. Multi-objective genetic algorithm

The NSGA II has been used to approximate the Pareto optimal set of the convex benchmark problem. For this simple benchmark function with a relatively small feasible set, only a small number of generations is needed to provide good approximations. Experiments have been conducted for the same size of solution sets as for the repeated ε -constraint algorithm, so a population size of 100 and 155. The cardinality of the solution set S is equal to the population size. With only 10 generations the algorithm managed to give some satisfactory results. Increasing the number of generations to 20 even improves the results. For a population size of 155 and 20 generations, in every run the final population consisted only of Pareto optimal solutions, and the coverage was 99.36% on average.

$ S $	Gen	p_m	Algorithm	t_{run}	$ S_{\mathcal{F}} $	$ S_{\mathcal{P}} $	$\text{cp}_{\mathcal{P}}$	s.d. $\text{cp}_{\mathcal{P}}$	G	s.d. G
100	10	0.1	BM1-GA-1	0.28 s	100	99.1	96.17%	3.59	0.000084	0.000126
	20	0.1	BM1-GA-2	0.50 s	100	99.4	98.51%	2.02	0.000063	0.000136
155	10	0.1	BM1-GA-3	0.42 s	155	154.2	98.09%	2.73	0.000045	0.000086
	20	0.1	BM1-GA-4	0.70 s	155	155.0	99.36%	1.03	0.000000	0.000000

Table 4.2: Performance indicators for different settings of the NSGA II, for the convex benchmark problem.



(a) Solution set of the BM1- ε -1 algorithm.

(b) One solution set of the BM1-GA-1 algorithm.

Figure 4.2: Comparison of the repeated ε -constraint algorithm and NSGAII for benchmark problem 1.

4.1.3. Comparison

The performance indicators as discussed in Section 3.4, are presented for the repeated ε -constraint algorithm and the NSGA II for 10 runs, and presented in Table 4.1 and Table 4.2.

In Figure 4.2, two solution sets with $|S| = 100$ from the repeated ε -constraint algorithm and from the NSGA II are plotted. In these plots, the Pareto optimal set is represented in large gray dots, and the orange dots indicate the solutions in the solution set. For the NSGA II, the run of BM1-GA-1 with the lowest number of Pareto optimal solutions is selected. Still the coverage of this run is much better than that of the BM1- ε -1 algorithm, while the computation is twice as fast.

4.2. Benchmark problem 2

For the second benchmark function, only the uncertainty of the second objective function will change with respect to the first benchmark problem. This benchmark is presented so we can analyze the differences in uncertainty. In this case, the uncertainty is not indicated with a changing θ , but as some functions that are the result of $\underline{\theta}$ and $\bar{\theta}$.

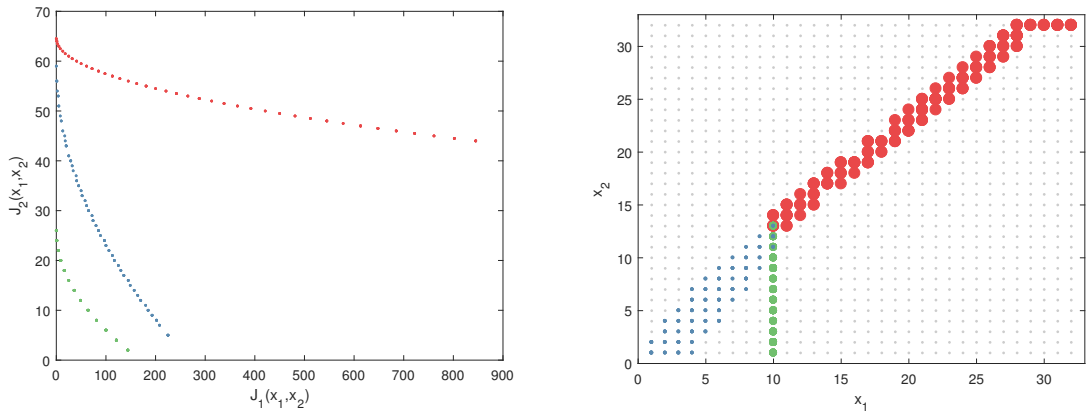
$$J_1(x_1, x_2) = (x_1 - 10)^2 + (x_2 - 13)^2 \quad (4.5)$$

$$J_2(x_1, x_2, \theta) = (2x_1 + 3x_2) \quad (4.6)$$

$$\underline{J}_2(x_1, x_2, \underline{\theta}) = 2x_2 \quad (4.7)$$

$$\overline{J}_2(x_1, x_2, \bar{\theta}) = 76 - \frac{(x_1 + x_2)}{2}. \quad (4.8)$$

The lower and upper bounds of the objective function J_2 are quite different. In the Pareto fronts, shown in Figure 4.3a, it seems like the objective values are only shifted, just as in the previous benchmark function. However, in Figure 4.3b, it can be seen that the Pareto sets for these upper and lower bounds are completely different from the Pareto set of the expected value of J_2 . In this case, there is exactly one solution that is Pareto optimal for all three optimization problems of J_2 , $\underline{J}_2$, and $\overline{J}_2$. This would reflect the most robust choice if the upper and lower bounds resemble the upper and lower bounds of all different possible scenarios. In the case that this solution that is Pareto optimal for each of the three scenarios would not be Pareto optimal for one of the objectives, the generational distance and other performance indices could show that the region around $x_1 = 10, x_2 = 13$ is a suitable range of solutions with respect to the lower bound, the expected value, and the upper bound of J_2 .



(a) Pareto optimal fronts for $\underline{J}_2, J_2, \overline{J}_2$.

(b) Pareto sets $\mathcal{P}$ in the solution space.

Figure 4.3: Pareto sets $\mathcal{P}$ and corresponding Pareto optimal fronts for benchmark problem 2.

4.3. Case studies

In this section, the investment project portfolio optimization approach will be tested on a railway network case study. For this case study, a part of the Dutch railway network is considered, that is visualized in Figure 4.4. The assets in this network are not considered individually. Twenty groups of assets are defined, based on asset type or location. Three asset types are considered, with different deterioration rates: track, switches, and railway signalling. A medium-level point of view is chosen. Time periods consist of two weeks (14 days), and the prediction horizon is set as 2 years. The computational effort that is needed to compute the states for all $N_p = 52$ time periods of 14 days, is hardly influenced by the length of the time periods, but mostly by the number of considered time periods. Therefore, the performance of the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II for the chosen prediction horizon are representative for a prediction horizon of 52 time periods of for example 1 hour or 1 year. For this network, 31 investment alternatives are defined. Some of the alternatives affect multiple groups of assets. Examples of these investment alternatives are tamping, track renewals, switch renewals, and the implementation of ERTMS⁵. For each of the investment alternatives, all the specified information is presented with a lower bound, an expected value, and an upper bound. With this information, three different optimization problems can be solved, and it can be analyzed with performance indices for all three optimization problems, whether project portfolios are expected to give satisfactory results in all scenarios.

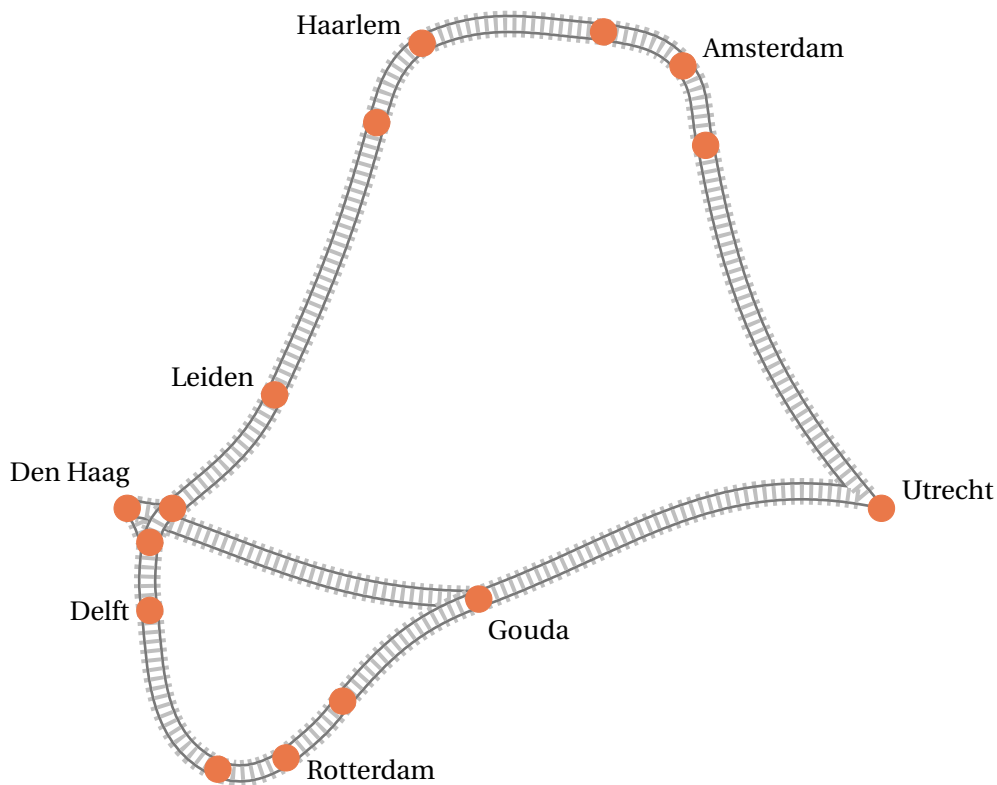


Figure 4.4: Part of the Dutch railway network that is used for the case study.

For the case study, an artificial data set is used that is based on information from the IBM Maximo asset management tool⁶ that is available for this thesis. The multi-objective investment

⁵The European Rail Traffic Management System, <http://www.ertms.net/>.

⁶<https://www.ibm.com/nl-en/marketplace/maximo>

project portfolio optimization problem for railway infrastructure networks, as formulated in Section 3.2, is implemented in both AIMMS and MATLAB.

For both the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II, different settings are tested for their performance. It is expected that even better results can be obtained by further tuning of the different settings. The obtained results are considered sufficient for comparison. The reliability of the algorithms is tested by considering the consistency of the algorithm over 10 runs. The results for the selected performance indices for these runs are presented in tables, where the average computation time t_{run} that is needed to obtain the approximation set, is in minutes. The general performance of the algorithms for the case studies are discussed. More details are provided for some algorithms with remarkable results. The visualizations of the obtained solution sets that are used for comparison are presented in parallel coordinate plots, such that individual solutions can be distinguished. The objective values are all normalized with respect to their minimum and maximum values that are present in the set of all feasible solutions that are obtained by the different algorithms. Solid markers and lines represent non-dominated solutions, and empty markers with dashed lines represent dominated solutions. For both the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II, the different settings are evaluated, after which a comparison of the two approaches is presented.

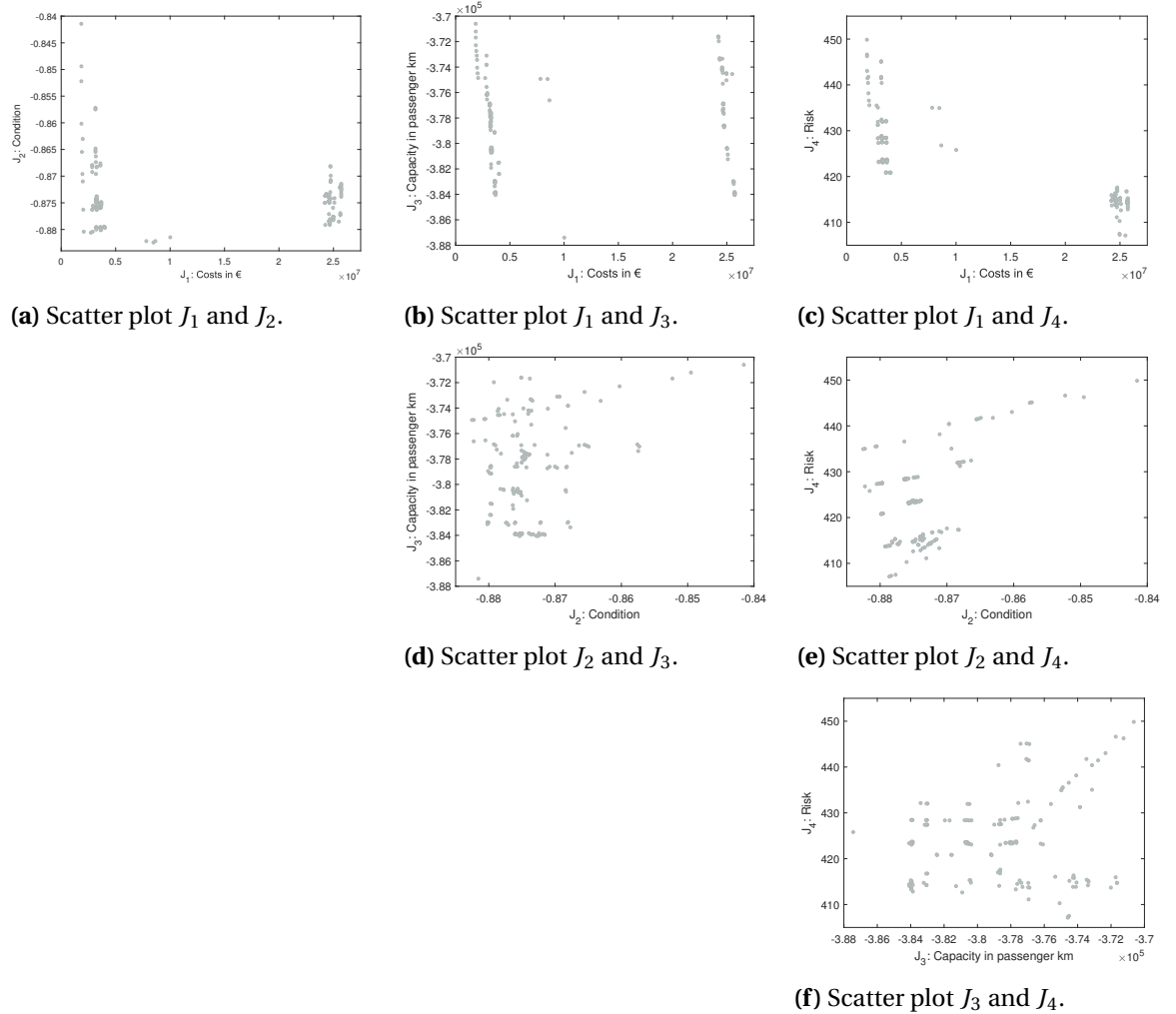


Figure 4.5: Scatter plot matrix for the non-dominated reference set $\mathcal{R}$ of the railway network case studies.

Since the real Pareto optimal set for the investment project portfolio optimization problem is not

known or easy to obtain, a reference set $\mathcal{R}$ is used. All the feasible solutions that are obtained by the executed algorithms are stored in the solution set $X_{\mathcal{R}}$. The non-dominated solutions in $X_{\mathcal{R}}$ are selected to be in $\mathcal{R}$. This set is used as a reference for all the algorithm tests to compute the performance indicators that regard the quality of the obtained solution sets. The reference set $\mathcal{R}$ is visualized in a scatter plot matrix in Figure 4.5, and in parallel coordinates plot in Figure 4.6. In both visualizations, it is clearly visible that for the non-dominated set $\mathcal{R}$ is divided into separate parts. This is due to two investment alternatives with exceptionally high costs. Project portfolios that include those costly investment alternatives can be non-dominated because of their good results for other objectives. A decision maker needs to judge whether the benefits for the other objectives outweigh the high costs.

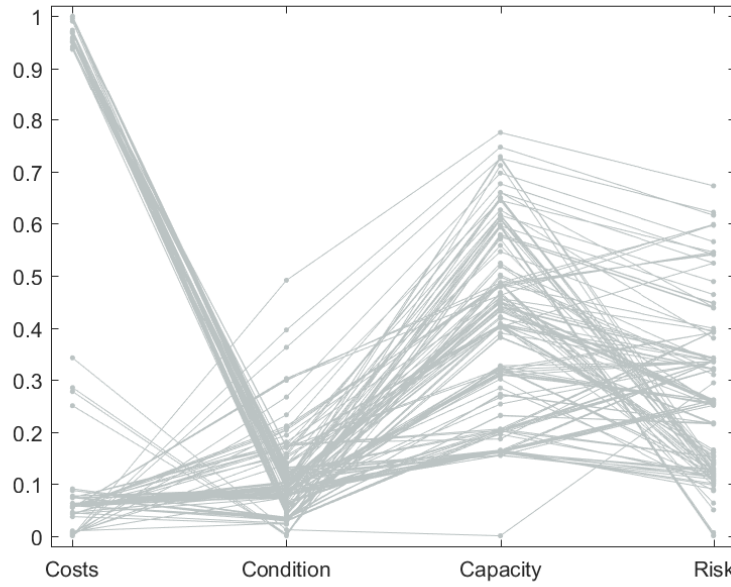


Figure 4.6: Parallel coordinates plot for the non-dominated reference set $\mathcal{R}$ of the railway network case studies.

4.3.1. Repeated ε -constraint algorithm

In the investment project portfolio optimization problem in railway infrastructure networks, 4 objectives are considered. For the case studies, the costs are chosen as main objective. This means that the size of the solution set S is dependent on the number of ε -bounds for the three bounded objectives. If for each objective, the same number g , of ε -bounds is posed, the size of the solution set is given by $|S| = 4 + g^3$. The first 4 solutions are obtained by the single-objective optimization of each of the objectives, without ε -bounds. These 4 solutions, the extreme points of the Pareto optimal set, are used for the minimum and maximum values of the ε -bounds. The combination of the three ε -bounds does not necessarily result in a feasible solution. This means that the number of resulting feasible solutions could be lower than $|S|$.

Solving many large-sized optimization problems could require high computational effort. Stopping criteria are introduced to bound the total running time of the algorithm. Next to full optimization, the algorithm will also be tested with a relative optimality tolerance stopping criterion. This means that the solution procedure stops as soon as the solver can guarantee that the best solution is within a percentage τ of the MIP relative optimality tolerance from the global optimum. The MIP relative optimality tolerance from the global optimum can be computed by comparison

of the best linear programming (LP) bound, and the best integer solution of the MIP problem. A drawback of such a method, is that the single-objective optimization problem is less likely to produce a Pareto optimal, or non-dominated solution. However, the generational distance to the reference set could be very small, while the computation time often reduces significantly if the solver does not have to guarantee a full optimum.

Furthermore, some combinations of ε -bounds result in a solution set with a feasible region that is difficult to reach for the CPLEX solver. This means that these combinations could require an excessive amount of computation time. This phenomenon can already be witnessed in the individual optimization problems for the four objectives. It depends on which objective function is optimized, how fast the solver can find the optimal solution. Full optimization of the second objective function J_2 , regarding the condition of the railway network, takes over 50 minutes, while for the other objectives it takes about 1 minute to find the optimal solution. Since an important performance indicator is the computation time that is needed to obtain the solution set, it is not desirable that the solver searches hours for one particular solution. Therefore, an additional stopping criterion could be used. This stopping criterion is time-based. The solver stops searching for the optimal solution when it exceeds the maximum computation time, that is set to 3600 seconds (one hour) for the investment project portfolio optimization problem. Once it is known that a particular combination of ε -bounds results in an infeasible solution, or exceeds the maximum computation time, one could choose to skip that optimization problem if the algorithm is repeated with the same settings. This also holds for ε -bounds that are all smaller or equal to the bounds that did not result in a satisfactory solution. The exit strategy as presented in Mavrotas (2009) is implemented to reduce computation time of the algorithm, not only for infeasible problems but also for the problems that were not solved within one hour. The computation time is reduced in comparison with the case that each of the computationally expensive ε combinations would have been addressed, but the computation is still time-consuming. Since it is not clear on forehand which combination of ε -bounds results in a computationally expensive optimization problem, it takes as long as the maximum computation time to identify such combinations. If the problem is adapted, the results regarding computation time might be totally different, so one can not assume that the combination of ε -bounds for two similar optimization problems are equally computationally expensive.

$ S $	τ	Algorithm	t_{run}	$ S_{\mathcal{F}} $	$ S_{\mathcal{R}} $	$\text{cp}_{\mathcal{R}}$	s.d. $\text{cp}_{\mathcal{R}}$	G	s.d. G
31	0%	CS- ε -1	142.22	22	12.0	3.29%	0.00	0.005401	0.000000
	1%	CS- ε -2	19.46	31	4.0	2.63%	0.00	0.014171	0.000000
	2%	CS- ε -3	14.46	31	2.0	1.32%	0.00	0.014579	0.000000
	5%	CS- ε -4	9.88	31	5.0	3.29%	0.00	0.014811	0.000000
68	0%	CS- ε -5	229.24	68	48.0	11.84%	0.00	0.002491	0.000000
	1%	CS- ε -6	55.90	68	13.0	8.55%	0.00	0.006778	0.000000
	2%	CS- ε -7	38.63	68	6.0	3.95%	0.00	0.006901	0.000000
	5%	CS- ε -8	31.65	68	11.0	7.24%	0.00	0.007232	0.000000
129	0%	CS- ε -9	427.33	128	99.0	21.05%	0.00	0.001898	0.000000
	1%	CS- ε -10	148.37	129	13.0	8.55%	0.00	0.003741	0.000000
	2%	CS- ε -11	75.77	129	29.0	18.42%	0.00	0.003883	0.000000
	5%	CS- ε -12	63.68	129	28.0	17.76%	0.00	0.004052	0.000000

Table 4.3: Settings and results of the repeated ε -constraint algorithm in the case studies.

All the settings that are used for the repeated ε -constraint algorithm in the case studies are presented in Table 4.3, along with the results of the 10 tests that have been performed in the last 7 columns. The selected performance indicators can be reviewed in Section 3.4. Most remarkable

about the performance indicators for the repeated ε -constraint algorithm is that the standard deviation of the reference set coverage and the standard deviation of the generational distance are 0.00. Due to the fixed search method of the CPLEX solver, the repeated ε -constraint algorithm provides very constant results.

The computation time of the full optimization algorithms, in which $\tau = 0\%$, is three to seven times as high as the computation time of the algorithms with $\tau = 1\%$. In return for the high computation time, the number of non-dominated solutions that are found is also higher. For the algorithm CS- ε -1, the solver has some difficulties finding the optimal solutions for the lowest value of ε_3 , the constraint for the risk objective. This results in only 22 feasible solutions, and therefore the number of non-dominated solutions is not that high. The result is that the computation time is disproportionately high, over seven times as high as for the algorithm with $\tau = 1$, while only producing three times as many non-dominated solutions. This is due to the computationally expensive combinations of ε -bounds. However, the generational distance is much lower than that of algorithms with higher values for τ . This means that the solutions in the solution set of CS- ε -1 are closer to the reference set than the solutions provided by other algorithms with $g = 3$. The generational distance G to the reference set $\mathcal{R}$ is also lower for the algorithms with $\tau = 0$ and $g = 4$ or $g = 5$, than the algorithms with higher values for τ . Especially the generational distance of CS- ε -9 is very low. This is because of algorithm found relatively more non-dominated solutions. The computation time of this algorithm is just so high, that repeating the optimization takes a lot of time. Especially when one wants to consider multiple scenarios or re-optimize after certain projects are selected without waiting for hours, this algorithm is not suitable.

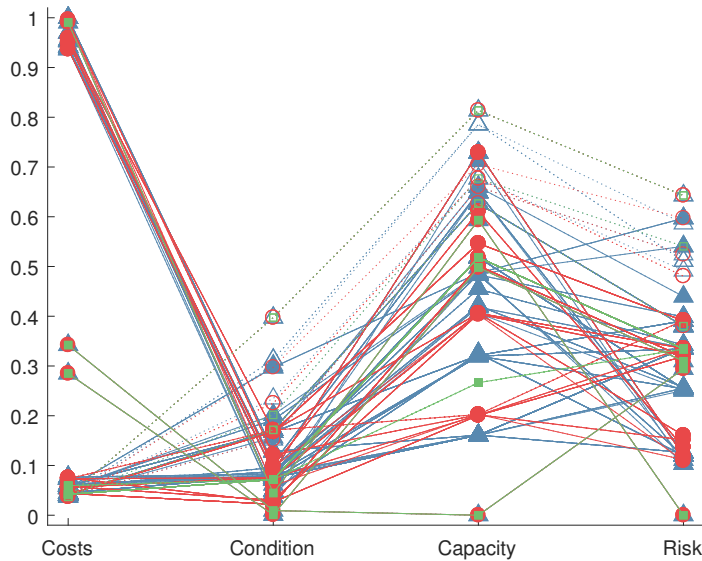


Figure 4.7: Solution sets for the repeated ε -constraint algorithm with $\tau = 0\%$, for $g = 3$, $g = 4$, and $g = 5$.

As can be seen in Figure 4.7 and Figure 4.8, the solution sets of the algorithms with different numbers of ε -bounds really provide different solutions. This means that using 5 ε -bounds per objective is not enough to capture the entire Pareto set. It seems therefore interesting to increase the number of bounds per objective. For $g = 6$ the solution set will consist of $|S| = 220$ solutions, which will increase the computation time even further. And unfortunately, for $g = 6$, the same problems as for $g = 3$ will appear since some ε -bounds will be equal. It is possible to increase the number of ε -bounds until the number of non-dominated solutions does not change anymore, but the extremely high computation time makes that such an approach is not recommended to use

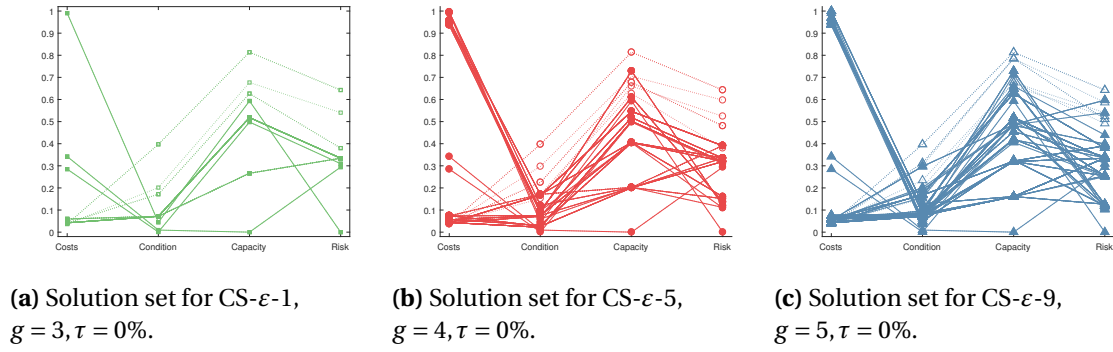


Figure 4.8: Non-dominated solution sets for the repeated ε -constraint algorithm.

in an interactive decision support tool, even though the results for other performance indicators will be very good. If the decision maker is not interested in re-optimizing repeatedly, or in quick results, but prefers to obtain a solution set that reflects the non-dominated reference set as well as possible, the repeated ε -constraint algorithm can be used with a higher number of ε -bounds. This could be the last step before the final decision-making. It is recommended to use an algorithm with less intensive settings to check the results for possible modelling errors, and to determine whether all constraints are implemented correctly.

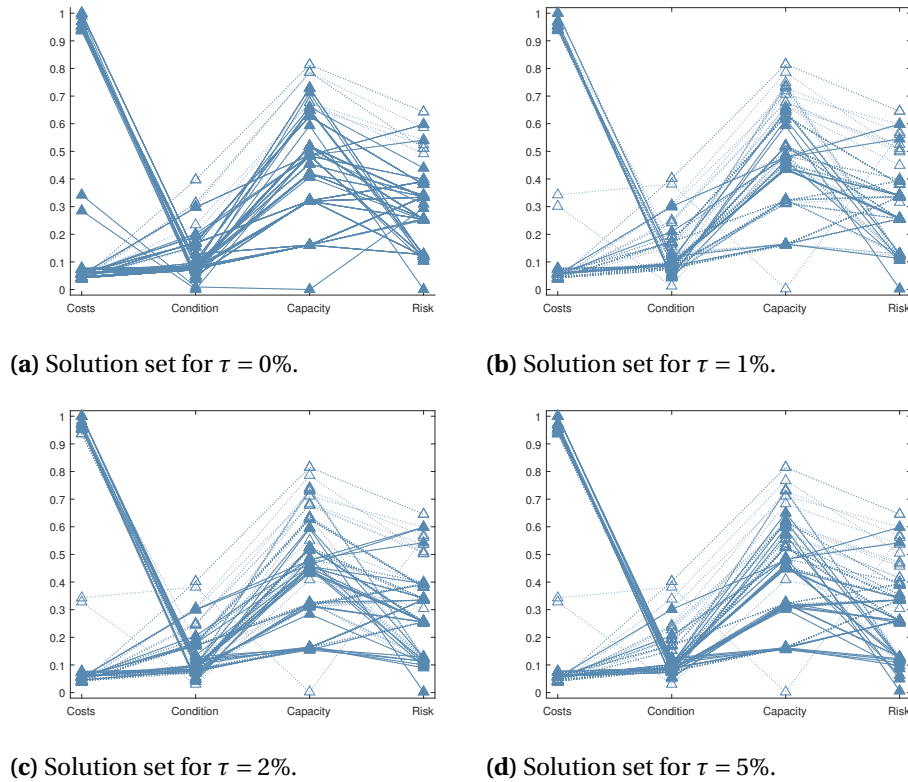


Figure 4.9: Solution sets for the repeated ε -constraint algorithm with 5 constraints per objective, $g = 5$.

Another option to obtain more unique non-dominated solutions, is to subsequently execute algorithms with different values for τ . Although one would expect that an algorithm with a stopping criterion that causes the algorithm to stop before full optimization is reached, would not result in non-dominated solutions in the reference set, it is possible. It should be noted however,

that there is no guarantee that the solutions obtained with positive values of τ are in fact non-dominated or Pareto optimal. Adjusting the MIP relative tolerance to the global optimum τ makes that the algorithm stops for a value that might not be optimal for the single-objective optimization problem that the solver is trying to solve, but the solution could result in a non-dominated solution in the union of all obtained solution sets. The solution sets of the algorithms CS- ϵ -9 till CS- ϵ -12 are shown individually in Figure 4.9. The solution sets are quite similar, but some differences can be spotted for the solution with the lowest value for the capacity objective. The solutions for the capacity optimization problem with positive values for τ are dominated by the solution for the same problem with $\tau = 0$. A curious observation is that CS- ϵ -10 with $\tau = 1$ obtained only 13 solutions that are included in the non-dominated set $\mathcal{R}$, while the two algorithms with $\tau = 2$ and $\tau = 5$ obtain 29 and 28 non-dominated solutions respectively. The algorithms CS- ϵ -3 and CS- ϵ -7 also obtain less non-dominated solutions in comparison with algorithms with higher values for τ .

The non-dominated solutions obtained by the algorithms with $g = 5$, CS- ϵ -9 till CS- ϵ -12 are shown in different shades of blue, together in Figure 4.10. Solutions for $\tau = 0\%$ are in light blue, $\tau = 1\%$ in mid blue, $\tau = 2\%$ in mid-dark blue, and for $\tau = 5\%$ in dark blue. Different sized square markers make it is possible to see which solutions are equal, and which are different. Although it is still difficult to really indicate which solutions belong to which algorithm, it can be distinguished that the algorithms with positive values for τ provide additional non-dominated solutions to the ones obtained with CS- ϵ -9. For example the solutions with a normalized risk value around 0.1 are not obtained with $\tau = 0$. Data analysis shows that CS- ϵ -9 obtains 32 unique non-dominated solutions, so a reference set coverage of 21.05%, and that the combination of the algorithms CS- ϵ -9 till CS- ϵ -12 provide a respectable number of 93 unique non-dominated solutions, which is 61.18% of the reference set. It takes a significant amount of time to run the four algorithms, almost 12 hours, but the performance in terms of reference set coverage is very high.

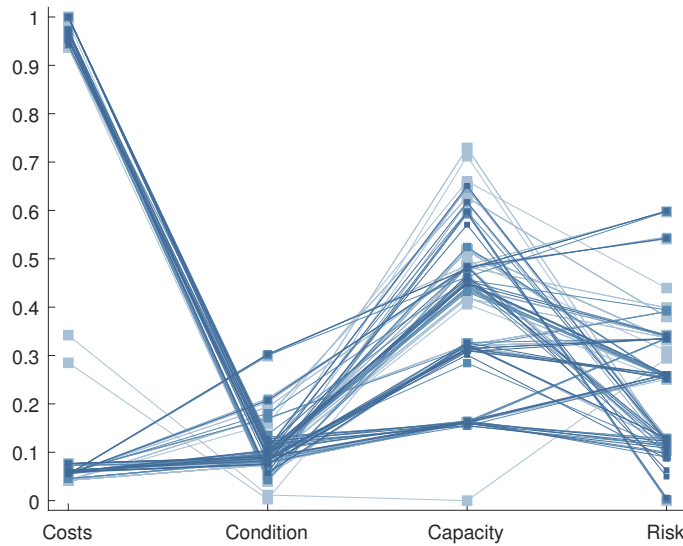


Figure 4.10: Non-dominated solution sets for the repeated ϵ -constraint algorithm with $g = 5$ and different shades of blue for different values of τ .

4.3.2. Multi-objective genetic algorithms

Next to the repeated ϵ -constraint algorithm, genetic algorithms are selected to approximate the Pareto optimal set of the investment project portfolio problem. Genetic algorithms are often cho-

sen because of their ability to handle non-convex problems, but definitely also because genetic algorithms are known to be fast and produce satisfactory results.

For the case studies, the binary strings for the chromosomes consists of two parts. The first part is a binary representation of which alternative is selected for each of the investments. Then for each investment, a binary representation is included of the time period in which the selected alternative will start. Since the amount of alternatives and time periods for each investment might not be equal to the number of possibilities in a binary representation, there might exist some infeasible solutions in the set of binary strings. To minimize the number of infeasible solutions, the binary representation for alternative selection consists of a minimal number of bits. The constraints posed in the formulation can also lead to a smaller number of feasible solutions. It is also possible to map the remainder of the binary representations to a feasible solution, but this will create a sort of preference for these solutions in the genetic algorithm. So instead of a repair technique, a rejecting technique is used. This means that for all solutions (binary strings) that appear in the algorithm, the error is computed. In the non-dominated sorting, the solutions with an error receive a lower rank than feasible solutions. Due to the low ranking, infeasible solutions are not likely to be selected for the next generation. Extinction of infeasible solutions follows. The total number of bits in the chromosomes is 100.

As discussed in Section 3.3, the multi-objective genetic algorithm NSGA II is used to search for non-dominated solutions. The algorithms are tested for different settings. For the population size, $|S| = 100$, $|S| = 200$, and $|S| = 400$ are tested. For each of these population sizes, tests are conducted with 100 generations and with 200 generations. The effect of the mutation probability p_m is tested for each of the combinations of population size and number of generations, with the proposed values of $p_m = 0.005$, $p_m = 0.01$, $p_m = 0.02$, and $p_m = 0.05$. An overview of the settings for NSGA II that are used to solve the investment project portfolio optimization problem can be seen in the first three columns of Table 4.4. The results for the performance indicators after testing these settings 10 times are presented in the last seven columns of this table.

In the column of $|S_F|$ it can be noticed that each of the tested multi-objective genetic algorithms provides as many feasible solutions as there are solutions in the population. This means that each of the algorithms is capable of rejecting the infeasible solutions.

The fast computation that was expected, can be confirmed. The computation time of the genetic algorithms is mostly affected by the population size and the number of generations. The computation time of the most computationally expensive tested genetic algorithms are for the settings with a population size of $|S| = 400$ and 200 generations, with a maximum of an average of 5.05 minutes.

Remarkable is that all the tests with a mutation probability of $p_m = 0.05$ failed to find any non-dominated solution. This indicates that the value is so high, that even when good solutions are selected as parent, the process of creating the offspring population is not able to maintain and improve the quality of the selected parents. Because the length of the binary string is 100, the proposed value for the mutation probability is $\frac{1}{100}$ (Deb et al., 2002), so with this information in mind it, the results for $p_m = 0.05$ are not striking. Algorithms with the setting $p_m = 0.05$ score by far lowest on the number of non-dominated solutions $|S_{\mathcal{R}}|$ and the generational distance G of all the tested genetic algorithms. Since the computation time of these algorithms are not lower, there is no performance indicator that favours this setting. Therefore, the mutation probability $p_m = 0.05$ can be rejected.

In figure 4.11, the solution sets of the 10 runs are visualized in the parallel coordinates plot together, for CS-GA-1 to CS-GA-4. The solution sets for the algorithms with $p_m = 0.005$, $p_m = 0.01$, and $p_m = 0.02$ contain some non-dominated solutions, shown by the solid lines and markers, in contrast to the setting $p_m = 0.05$. The algorithms with lower values for the mutation probability provide a more dense solution set, so less unique solutions. In the visualization all the 10 runs

$ S $	Gen	p_m	Algorithm	t_{run}	$ S_{\mathcal{F}} $	$ S_{\mathcal{R}} $	$\text{cp}_{\mathcal{R}}$	s.d. $\text{cp}_{\mathcal{R}}$	G	s.d. G
100	100	0.005	CS-GA-1	0.55	100	35.3	2.24%	3.03	0.000436	0.000542
		0.01	CS-GA-2	0.56	100	51.0	3.16%	2.72	0.000439	0.000420
		0.02	CS-GA-3	0.55	100	5.9	0.72%	1.53	0.003116	0.001651
		0.05	CS-GA-4	0.56	100	0	0.00%	0.00	0.006841	0.001884
	200	0.005	CS-GA-5	1.10	100	36.1	2.30%	3.03	0.000165	0.000158
		0.01	CS-GA-6	1.11	100	51.5	3.49%	3.07	0.000208	0.000267
		0.02	CS-GA-7	1.10	100	91.0	6.05%	0.81	0.000095	0.000105
		0.05	CS-GA-8	1.11	100	0	0.00%	0.00	0.005088	0.001062
200	100	0.005	CS-GA-9	1.23	200	192.5	6.18%	0.99	0.000078	0.000139
		0.01	CS-GA-10	1.12	200	147.5	5.07%	1.99	0.000305	0.000526
		0.02	CS-GA-11	1.23	200	7.7	1.12%	1.58	0.002343	0.001389
		0.05	CS-GA-12	1.12	200	0	0.00%	0.00	0.004669	0.001051
	200	0.005	CS-GA-13	2.39	200	176.6	5.92%	2.21	0.000022	0.000030
		0.01	CS-GA-14	2.37	200	138.2	4.87%	3.15	0.000026	0.000055
		0.02	CS-GA-15	2.42	200	192.9	6.97%	0.55	0.000033	0.000098
		0.05	CS-GA-16	2.40	200	0	0.00%	0.00	0.003649	0.001189
400	100	0.005	CS-GA-17	2.48	400	390.1	6.38%	0.82	0.000014	0.000018
		0.01	CS-GA-18	2.47	400	356.9	6.12%	2.22	0.000012	0.000023
		0.02	CS-GA-19	2.48	400	68.7	4.67%	2.25	0.001420	0.000489
		0.05	CS-GA-20	2.48	400	0	0.00%	0.00	0.002741	0.000410
	200	0.005	CS-GA-21	5.05	400	396.2	6.71%	0.60	0.000009	0.000015
		0.01	CS-GA-22	4.49	400	397.7	7.17%	0.65	0.000012	0.000035
		0.02	CS-GA-23	4.48	400	397.5	7.30%	0.58	0.000113	0.000357
		0.05	CS-GA-24	4.41	400	0	0.00%	0.00	0.002446	0.000380

Table 4.4: Settings and performance indices for the NSGA II in the case studies.

are combined, so it is not clear from the figure whether the results are the same in the different runs. However, in Table 4.2 it can be noted that the standard deviation of the coverage is quite high. Actually, some of the runs do not contain non-dominated solutions at all. For CS-GA-1, the maximum number of non-dominated solutions in the solution set is 11, and in 6 of the 10 runs no non-dominated solution was found. The algorithm CS-GA-2 has a maximum of 10, and 3 times no non-dominated solutions, and lastly CS-GA-3 has a maximum of 6, but 8 times no non-dominated solutions. Based on these results, one would select a mutation probability of $p_m = 0.01$, since this results in the highest average coverage.

The cause of the large differences in the results of an algorithm with exactly the same settings, is the random initial population. If all the 100 solutions in the initial solutions are non-dominated, perfect results would be obtained. But if all the 100 solutions in the initial population are far from the non-dominated set, the algorithm depends on the crossover and mutation operators to obtain solutions with an improved non-domination rank. The total performance of the genetic algorithms are dependent on random variables selected for the initial population, and on random variables used for tournament selection, crossover, and mutation. To increase the probability that there are non-dominated solutions in the initial population, the population size could be increased. In Figure 4.12, the results for three individual runs of the NSGA II are presented, with different population sizes. For CS-GA-2, CS-GA-10, and CS-GA-18, the best runs are selected. It can be seen that the number of non-dominated solutions is not growing very fast. For $|S| = 100$, 10 unique non-dominated solutions are found, while for $|S| = 200$, and $|S| = 400$, 11 unique non-dominated solutions are found. This improvement is quite low. However, the algorithm becomes

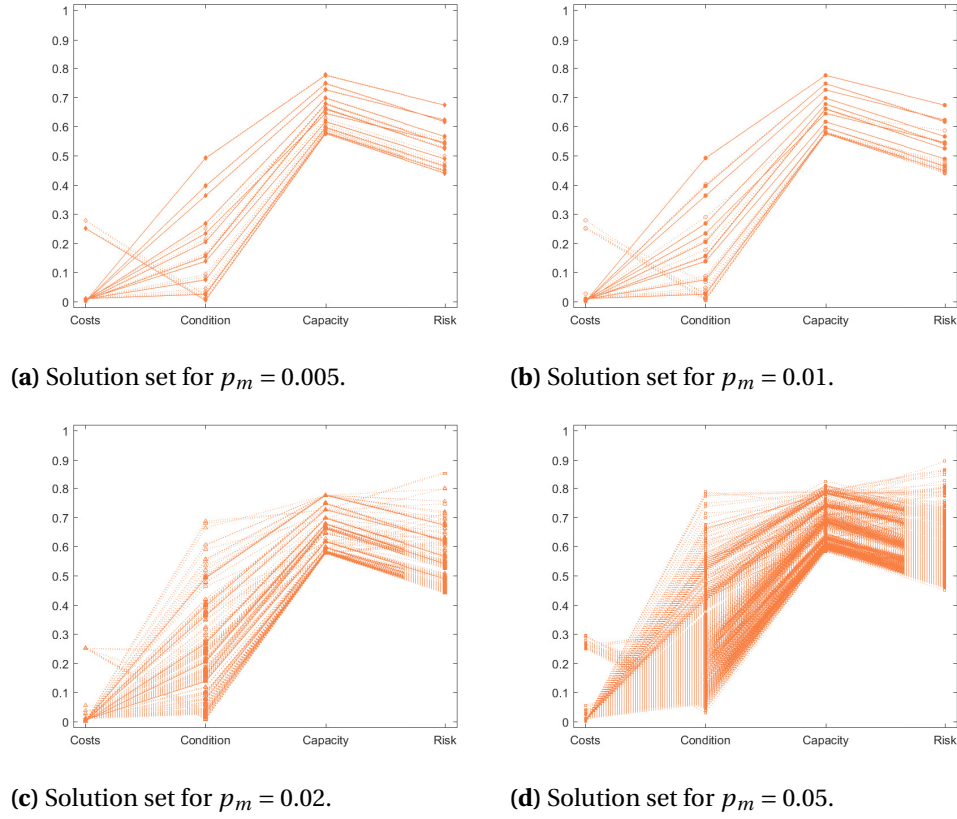


Figure 4.11: Solution sets for the NSGA II with $|S| = 100$ and $\text{gen} = 100$.

more consistent if the population size is larger. For CS-GA-2, three of the ten runs do not provide any non-dominated solutions, while for the other two algorithms, this only happens once. If increasing the computation time a bit more does not create any problems, one could test even larger population sizes.

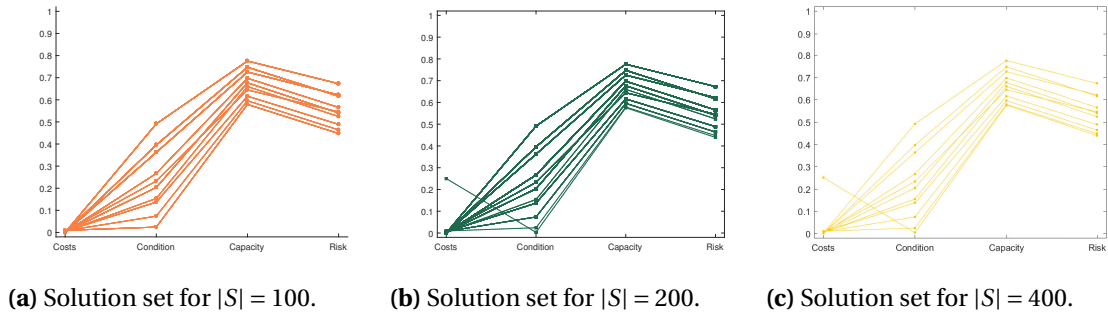
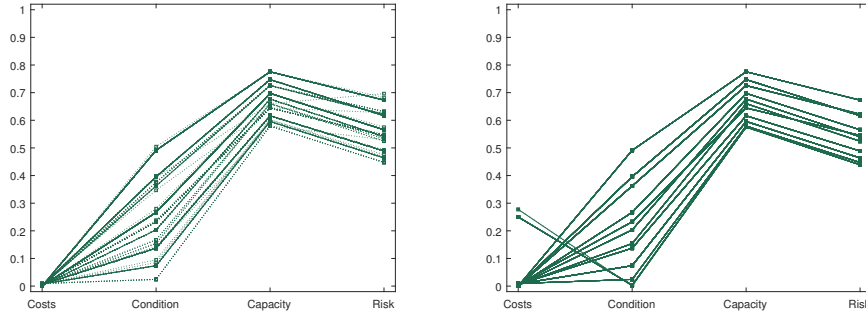


Figure 4.12: Solution sets for the NSGA II with $\text{gen} = 100$, and $p_m = 0.01$.

Lastly, the influence of the number of generations is considered. The algorithms are tested with 100 generations and with 200 generations. Interesting is that overall, the number of non-dominated solutions provided by the algorithm is not increasing significantly for doubling the number of generations. Only for $p_m = 0.02$, we see that the performance improves a lot if the number of generations is doubled. So if 200 generations are considered, $p_m = 0.02$ would be preferred. In Figure 4.13, it can be seen that the best individual runs of CS-GA-11 and CS-GA-15 are quite different. CS-GA-15 provides much more unique non-dominated solutions, and is also more

consistent, since the number of unique non-dominated solutions varies between 9 and 12, while for CS-GA-11, this varies between 0 and 7, with multiple times 0.



(a) Solution set for Gen=100.

(b) Solution set for Gen=200.

Figure 4.13: Solution sets for the NSGA II with $|S| = 200$, and $p_m = 0.02$.

4.3.3. Comparison

The performance of the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II are quite different. The repeated ε -constraint algorithm requires an excessive amount of computation time, but provides constant results. A higher number of ε -bounds indicates that more computation time is needed to find the optimal value for the main objective. The repeated ε -constraint algorithm provides satisfactory results for the performance indices of CS- ε -9, except for the computation time.

The highest coverage of the NSGA II case studies are for CS-GA-23, but still the highest coverage is only about a third of that of the repeated ε -constraint algorithm. The good thing about the NSGA II is the computation time. As can be seen in Table 4.5, the computation time of some algorithms with similar coverage, CS- ε -8, CS-GA-15, and CS-GA-23, the computation time of CS- ε -8 is seven times higher, while the coverage is lower than that of CS-GA-23. The four algorithms presented in Table 4.5, are all good options for optimizing the multi-objective project portfolio optimization problem, depending on the preferences of the decision maker for the performance indices.

Algorithm	t_{run}	$ S_{\mathcal{F}} $	$ S_{\mathcal{R}} $	$\text{cp}_{\mathcal{R}}$	s.d. $\text{cp}_{\mathcal{R}}$	G	s.d. G
CS- ε -8	31.65	68	11.0	7.24%	0.00	0.007232	0.000000
CS- ε -9	427.33	128	99.0	21.05%	0.00	0.001898	0.000000
CS-GA-15	2.42	200	192.9	6.97%	0.55	0.000033	0.000098
CS-GA-23	4.48	400	397.5	7.30%	0.58	0.000113	0.000357

Table 4.5: Comparison of performance indices for the repeated ε -constraint algorithm and the NSGA II.

What is not captured in the current performance indices, is the *spread* of the approximation set. None of the visualizations of the solutions sets of the NSGA II algorithms includes a project portfolio that minimizes the risk or maximizes the capacity (minimize J_3). As was already discussed for the repeated ε -constraint method, it seems that a solutions minimizing the risk score is very hard to obtain, and probably only a few solutions that give similar good results for the risk score. Since the extreme points of the non-dominated front are used in the initialization of the repeated ε -constraint algorithm, this method will always score very good on spread. For the NSGA II algorithms, this depends on the initial population, and how well the settings allow for new results. This difference can be eliminated if for the NSGA II algorithms, the extreme points will be used in the initialization of the population. But this will increase the computation time.

4.4. Decision support

The mathematical formulation and optimization technique for the investment project portfolio optimization problem are very important for the decision-making process. However, these aspects are included in the back end of a decision support tool. For a decision support tool to work successfully, the front end of the tool has to be designed carefully, and adjusted for the specific needs of an organization. Although the complete design of such a decision support tool is out of scope for this thesis, some thought has put into the decision-making process.

In Section 3.5, some statements regarding the decision support tool are presented. Some of the desired features are implemented in a decision support tool demo that has been developed in AIMMS. In Appendix A, some of the screens of the user interface are presented. Some remarks on the functionality and other points of attention are included in the conclusions.

4.5. Summary

In this chapter, the proposed investment project portfolio optimization approach is put into practice. The performance of the repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II are examined. First some simple benchmark problems, for which the Pareto optimal set is known, are used to gain insight in the effect of the different settings of the proposed algorithms on the performance. For the benchmark problems, the use of lower and upper bounds in the problem formulation is also analyzed.

The problem formulation as proposed in Chapter 3, is used as a case study with an artificial data set, for the investment project portfolio optimization problem in railway infrastructure networks. Since the set of Pareto optimal solutions for this case study is unknown, the set of all obtained solutions is used as a reference set. The repeated ε -constraint algorithm, implemented in AIMMS, provides constant results. The different settings for the algorithm score very well on the coverage of the non-dominated set and the generational distance to the non-dominated set, especially for a large number of ε -bounds. However, optimizing many single-objective optimization problems is time consuming.

The multi-objective genetic algorithm NSGA II, implemented in MATLAB, is less consistent than the repeated ε -constraint algorithm. A genetic algorithm is dependent on the randomly chosen initial population, and also on the random genetic operators. For a large initial population and many generations the coverage of the NSGA II algorithms is acceptable. The tested settings provide less unique non-dominated solutions than most of the tested settings for the repeated ε -constraint algorithm. The biggest advantage of using genetic algorithms to approximate the Pareto optimal set of project portfolios, is that the computation time is very good.

Conclusions and recommendations

In this chapter, the findings of this thesis are presented. First some general remarks and insights that are gathered throughout the thesis project are listed. The research questions that are introduced in Section 1.3 are reviewed. Answers to the research questions are formulated according to the findings and results of this thesis. Then, a discussion on some of the assumptions and results are presented. Lastly, the findings of this thesis that are relevant for further expanding this thesis project to a usable decision support tool, are presented as recommendations for ORTEC.

5.1. Conclusions

The assessment of different aspects of the decision-making process in investment project portfolio planning is presented in this section.

5.1.1. Problem formulation

In the decision-making for project portfolio optimization in infrastructure networks, multiple objectives are considered due to multiple stakeholders in the infrastructure network with different desires. It is therefore strongly recommended to include a multi-objective optimization problem in a decision support tool. Only the most important objectives with respect to the strategical goals of an organization should be included as objective in the optimization problem. A situation with strong differences in the preference of Pareto optimal solutions exist, should be avoided. Insight into the less important objectives can be obtained in the evaluation phase of Pareto optimal project portfolios.

In infrastructure networks, investment projects are often related to the assets. Since maintenance projects and investment projects often make use of the same resources, and effect the same organizational objectives, it is suggested that these type of projects are considered in the same planning problem. In the formulation for these projects, maintenance objectives can be selected multiple times in the prediction horizon, whereas investment projects are allowed to start only once. The effect of a maintenance project is considered as an improvement of the current asset states, whereas an investment project might change the behavior of the asset.

Assets in infrastructure networks can not be considered as an individual, the connectivity of the assets should be taken into account. For example in the proposed formulation for project portfolio optimization in railway networks, the capacity of the assets is influenced by the states of neighboring assets. Furthermore, a combination of multiple executed projects does not necessarily result in the sum of all individual project benefits. The effect of executing complementary projects and substitutive projects should be taken into account, to what extent they can be estimated.

Due to the long lifespan of assets in infrastructure networks, and that the nature of an investment is that it provides long-term benefits, a high-level point of view should be considered in the project portfolio optimization problem. This means a long prediction horizon and corresponding

long time periods. Since the resulting project portfolios cannot be used for daily project management and execution, low-level optimization could be used to obtain daily schedules, based on the results of the high-level planning.

The available information about the costs, duration, and benefits of investment projects is often based on the judgment and expectations of experts. The realizations could deviate from the expected values for costs and benefits, due to for example errors in calculation, non-fulfilled agreements with external parties, and unexpected delays. To obtain a set of solutions that could be the basis for confident decision making, it is preferable that uncertainty is included in the problem formulation. However, regarding the computation time, it is not recommended to include all possible realizations of the uncertainties. An approach is proposed to include the uncertain nature of investment projects in the problem formulation, without increasing the complexity of the problem too much. This is done by introducing lower and upper bounds for the uncertain variables. By optimizing not only the expected values of the objective functions, but also the lower and the upper bounds for these values, one is able to identify solutions that give satisfactory results for the three considered scenarios. Project portfolios that are Pareto optimal or non-dominated for the lower bounds, the expected values, and the upper bounds of the objective functions, are considered as robust choices with respect to the uncertainties in the objective values.

5.1.2. Multi-objective optimization methods

In the optimization of multiple objectives for the decision support of project portfolio selection, it is preferable to explicitly consider the multiple objectives. This means that an a posteriori preference structure is used in the optimization method. In the selection of an optimization method, a trade-off has to be made regarding the computation time and the quality of the approximation of the Pareto optimal set. This trade-off has to be made according to the preferences of the users or the organization. Here it is argued that the results regarding the solution quality of the repeated ϵ -constraint algorithm is not sufficient for any combination of settings, to outweigh the excessive amount of computation time that is needed to obtain the solution set. Since for large-sized optimization problems the set of Pareto optimal solutions is not attainable within acceptable time, a reference set is used for comparison of the methods. The multi-objective genetic algorithm NSGA II is proposed, at the expense of the repeated ϵ -constraint algorithm. The genetic algorithm NSGA II obtained the approximation set within 5 minutes, for all tested settings. The optimal settings might be different for other problems, but for the railway network case studies, a population size of 400 is selected, with 200 generations and a mutation probability of 0.01 or 0.02. The computation time is important in the decision support tool, especially since the proposed formulation includes three optimization problems, that for the lower bounds of the objectives, the expected values, and the upper bounds. The computation of the repeated ϵ -constraint algorithm is not suitable for optimizing multiple scenarios and giving the user the opportunity to re-optimize the problem with different constraints.

5.1.3. Visualization and decision support

The investment project portfolio optimization approach is intended to be used to support the decision-making process. A well designed user interface, with the right amount of detailed information, easy to understand visualizations, and an intuitive user flow is crucial for the success of a decision support tool. A fully operating decision support tool that meets all the requirements that are posed here is out of scope for this thesis. However, a demo version has been developed in AIMMS, including some general requirements. Some screenshots of this demo version are presented in Appendix A.

The decision support tool should be interactive in the sense that multiple scenarios regarding local and general constraints can be optimized, and that the problem can be adapted after the de-

cision maker gains insight in the possible solutions. During the decision-making process, a set of non-dominated solutions is provided to the decision maker. To prevent that an excessive amount of data creates the issue that the overview is lost, the decision maker should be able to use some features such as filtering and selecting solutions for detailed information. Also additional information should be provided for decision-making, such as the core index of the projects. The decision maker should be able to adjust constraints, and re-optimize the problem. Interaction with the tool should lead the decision maker step-by-step towards the final project portfolio selection. By considering the lower and upper bounds of multiple objectives, it is considered that the decision maker is able to gain sufficient insight into the possible solutions to make well-founded decisions and select a robust project portfolio.

A point of concern is the easy to understand visualization of the obtained set of non-dominated project portfolios. Visualizations for 4 objectives tend to be less intuitive or do not fully manage to provide a clear overview of the solutions. This is especially the case if many possible solutions are included in the visualization. Therefore, including the lower bound, expected value, and upper bounds for the objective values of many solutions in the same visualization is not recommended. Examples of possible options are to provide the three visualizations next to each other, or to select a single solution for which the upper and lower bound are visualized.

5.1.4. Research questions

The main research question of this thesis is given by:

How can multi-objective optimization be used efficiently in the decision support for investment project portfolio planning under uncertainty?

In order to answer the main research question, the sub-questions are addressed and reflected on, with the findings that are presented in the previous sections.

- *What is the most suitable way to include uncertainty in parameters and future scenarios in the optimization of maintenance and investment project planning?*

Considering more than just the expected values of variables increases the computation time of algorithms. Including uncertainty in the problem would help the credibility of the provided solutions, but another desire for a decision support approach is that the computation time is not extravagant. To make sure the computation time does not suffer too much, only three realizations of each uncertain variable are considered. Next to the expected value of that variable θ , also the lower bound $\underline{\theta}$ and the upper bound $\bar{\theta}$ are considered. Solving three optimization problems, one for the lower bounds of the objective values, one for the expected values of the objective values, and one for the upper bounds of the objective values, will provide insight in which solutions are robust choices. One would typically prefer a solution that is Pareto optimal (or non-dominated) for all three problems, since such a solution is considered to be a robust project portfolio planning.

- *In the high-level project portfolio optimization problem, what are the trade-offs between accuracy and computation time when using different time steps, network segments, asset information, and number of project dependencies?*

When a high-level project portfolio selection and planning problem is considered, it goes without saying that the information used in optimization should not be too detailed. For realistic, accurate results one would expect that it is valuable to consider the states of all assets individually, and for a large number of time steps. However, this will imply that large computational effort is needed to compute all the states for the time steps in the prediction horizon.

For the sake of computation time, it is better to combine individual assets into groups of assets based on for example asset type, location, and age. Furthermore, the combined states of the assets in a group are likely to be closer to their expected values, since deviations are averaged. Project dependencies are considered too important for the accuracy to omit.

- *How can we cope with different requirements for the decision support tool in the railway infrastructure business, regarding computation time, solution quality, and the process of the support?*

A trade-off needs to be made for the computation time and the solution quality. The preference of different organizations in this trade-off could be different, but for the general high-level investment project portfolio optimization problem in railway infrastructure networks, it is considered that there exists a fast optimization method that provides a sufficient level of solution quality. By selecting an optimization method that is computationally inexpensive, the decision makers in the railway infrastructure business are able to consider multiple scenarios and realizations of uncertainty. The settings for the selected optimization method could be tuned to provide the best possible solution quality in terms of for example the coverage of the non-dominated set of solutions, and the generational distance to the non-dominated set.

- *Which optimization methods are best suited for the multi-objective project portfolio problem, regarding solution quality and computation time?*

The repeated ε -constraint algorithm is considered suitable for the final optimization of a project portfolio, but too computationally expensive to use in an interactive decision support tool. The solution quality in optimizing different scenarios of constraints, and different realizations of uncertainty, is considered to be less important than the computation time to obtain these results. For a fast and interactive decision support tool, the multi-objective genetic algorithm NSGA II is proposed. The solution quality of the genetic algorithm is influenced by the settings for population size, number of generations, and mutation probability. These settings can be tuned for a specific problem to meet the preferences of the organization.

5.2. Discussion and further research

Some issues that would benefit from some more attention are discussed here, as well as the possibilities to conduct further research in this area.

- Realistic railway network representation

The deterioration of the condition and states of the assets, as well as the dependencies between the states, are highly simplified in the railway network representation in this thesis. Improving the behavior of the states is out of the scope for this thesis, but it would certainly improve the accuracy of the model, so it should be implemented. It is proposed that the decision support tool and the optimization part are connected to an asset management tool such as Maximo. In Maximo, the deterioration models, risk assignment, and all the asset relations are recorded.

- Including uncertainty

In the problem formulation for investment project portfolio optimization, the uncertainties in expected results of investment projects are considered. The railway network representation is subject to more uncertain influences than just the effects of investment projects. Even if the deterioration models are more accurate, there still might be some modelling errors or

noise. The proposed problem formulation can be adapted to include other uncertainties as well. However, attention must be paid that the lower and upper bounds do not diverge over time. If for each condition in each time period, a lower and an upper bound is considered that is depending on the values of the previous time period, the interval between the lower and the upper bound will grow very fast.

- Real life case study data

The best possible way to investigate whether a decision support tool, or any tool for that matter, works properly, is by testing the methods with real life data. Real life data was expected, but due to procurement rules at the concerning railway organization, this was not possible. The current case study data is based on other infrastructure data that was available from the IBM Maximo Asset Management tool. Unfortunately, the size of the artificial data set is not representative for a real life railway network case study.

- Comparison of different software tools

The repeated ε -constraint algorithm and the multi-objective genetic algorithm NSGA II are not implemented in the same software tool. The reason for this is AIMMS is the standard optimization software for ORTEC Consulting, so a suitable choice for a demo version of a tool with single-objective optimization problems. However, it is not straightforward to include multi-objective genetic algorithms in AIMMS. For sake fast implementation, MATLAB is selected for the implementation of NSGA II. The drawback of this choice is that the two optimizers are not equally able to adress the cores of the processor of the computer. So despite that all the results are obtained with the same computer, without other computationally expensive processes to influence the results, the computation time that is needed for the optimization is not representative for the computational effort that is needed. However, since NSGA II actually performs better than the repeated ε -constraint algorithm in terms of computation time, this did not influence the selection for the best suited optimization method. If the MATLAB implementation would be adapted such that the used computational power of the computer would be increased, NSGA II would perform even better with respect to the computation time.

- Optimization method improvements

Tuning the settings of an algorithm could further improve the performance of an algorithm to meet the preferences of the decision makers. Just recently, a MATLAB Platform for Evolutionary Multi-Objective Optimization (PlatEMO) is published (Tian et al., 2017). This platform will make future implementations of genetic algorithms such as NSGA II a lot easier, as well as tuning the settings. In this platform, the selection of different visualization methods and performance indices is expected to be well supported. If this platform is able to keep up with the state of the art optimization methods, it will be a very valuable and widely used MATLAB-based platform. The current version of PlatEMO includes 50 multi-objective optimization algorithms. Other methods can therefore be easily tested for their performance. A method that has not been tested thoroughly for the case studies, but is expected to give good results is to include the extreme points in the initial population of the multi-objective genetic algorithm NSGA II. Other methods that could be compared in future research are for example Pareto ant colony optimization and multi-objective evolutionary algorithm based on ε -dominance.

- Visualization method

It is stated in this thesis that a decision support tool would benefit from a simple, clear visualization of the Pareto optimal or non-dominated solutions. Also for comparison of case

study results in this thesis, a clear visualization is desirable. However, it has been found that distinguishing individual solutions in the parallel coordinates plot requires a good pair of eyes. This issue could be solved partly by increasing the height of the figure. But still it will not be clear if the lower and upper bounds of the objective values are included in the visualization. Therefore, it is suggested that some further research could be conducted to find or develop a clear, intuitive way of visualizing the values of multiple objective values, for multiple scenarios.

- Application

The application of investment project portfolio optimization in this thesis is selected to be railway infrastructures. The capacity of the railway network is presented for passenger transport. For use in a network with combined passenger and cargo transport, the capacity calculation should be adjusted accordingly. The infrastructure network representation of other infrastructures such as road networks, water networks, or electricity networks is expected to be only slightly different than that of a railway infrastructure network. It would therefore be interesting to dedicate some research in these other fields, and investigate whether the requirements for such a decision support tool are similar. Next to other infrastructure networks, also more general project portfolio optimization problems can be addressed. The network specific dependencies can be replaced by other problem specific adjustments.

- Hierarchical decision-making process

In the proposed problem formulation and solution process, different objectives are checked for non-dominance. So the optimization problem is solved for the lower bounds of the objectives, the expected values of the objectives, and the upper bounds for the objectives. With the aim to provide even more robust solutions, the problem can be extended in the following way. The solutions are now checked for non-dominance for a long-term prediction horizon. It would be interesting to consider the set of non-dominated solutions, and use these solutions to compute the objective values for a medium-term prediction horizon or a short-term prediction horizon. Since only the number of non-dominated solutions that are provided for the initial problem have to be checked for the objectives, this step is not expected to be computationally expensive. If the solutions are then checked for non-dominance regarding the objectives for a shorter prediction horizon, the non-dominated solutions of this set are considered to be robust in the sense that both the long-term goals as the short-term goals are considered. The problem for the short term prediction horizon could also be solved for all possible solutions, but this will take more time and will probably does not provide the same solutions as the initial problem.

5.3. Recommendations for ORTEC

It would be a huge success if the proposed formulation and decision support tool outline from this thesis are used in the development of a real life user application. This attempt is of-course supported and encouraged, and here are some recommendations presented for ORTEC.

First of all, the experiences of a two-day conference of the UIC Railway Asset Management Conference, has led to the conclusions that organizations are definitely interested in the optimization of their maintenance planning, by using asset data. However, most railway organizations are not there yet. The railway industry is quite conservative, and often limited in their expenses because of government cuts. Some people are really willing, and convinced that some innovation should take place, but in most organizations this is not widely supported yet. Furthermore, the railway organizations do not have all the required data present. Therefore, it is recommended to use this thesis as an ultimate goal, but guide the organization step by step to achieve this goal. A

data-maturity assessment could be used to check what data is available in the organization, and what are the possibilities for optimization with this data.

Even if the optimization problem is less complex, and other optimization methods could be used, it is important for the decision-making process that the optimization problem formulation reflects reality as good as possible. Decision makers need to trust the results obtained by the optimization method. So it will be good to cooperate with the decision makers when creating the mathematical formulation.

Furthermore, it is suggested that a professional user interface designer is involved in the development of a real life decision support tool. Firstly, selling a good looking, intuitive decision support tool would be a lot easier. And if decision makers need to work with a tool that is not user friendly, they are not likely to consider other ORTEC solutions, and the long term relationship is damaged.

Lastly, if some complex optimization problems need to be solved, and the client does not want to make sacrifices regarding the computation time, it could be an option to use cloud computing.

A

Decision support tool

The main goal of a decision support tool for project portfolio optimization in infrastructure networks is to gain insight into the possible scenarios, such that a well-founded, robust project portfolio can be selected. Before this can be realized, the decision maker first needs to have insight into the current situation, so insight into the current states of the network and the possible projects. Then, some interaction with the tool is needed to include the preferences of the decision maker, and to make sure the results are fully aligned with the concerned organization. Here the features that are deemed necessary, and some basic user interface pages that are implemented in AIMMS, are presented.

A.1. Insight into the current situation

For insight into the current situation, data is needed. It would be most convenient if the decision support tool is linked to an Asset Management system such as Maximo, such that the information in the tool is always up to date and less sensitive to human errors in data import or export. If this is not done automatically, the data has to be imported by the user. In that case, some guidance is recommended to make sure the right data is submitted. This can be accomplished with some checks and feedback to the user. In Figure A.1, a data import page is showed. Some checks are performed to make sure that all the data is imported and up to date.

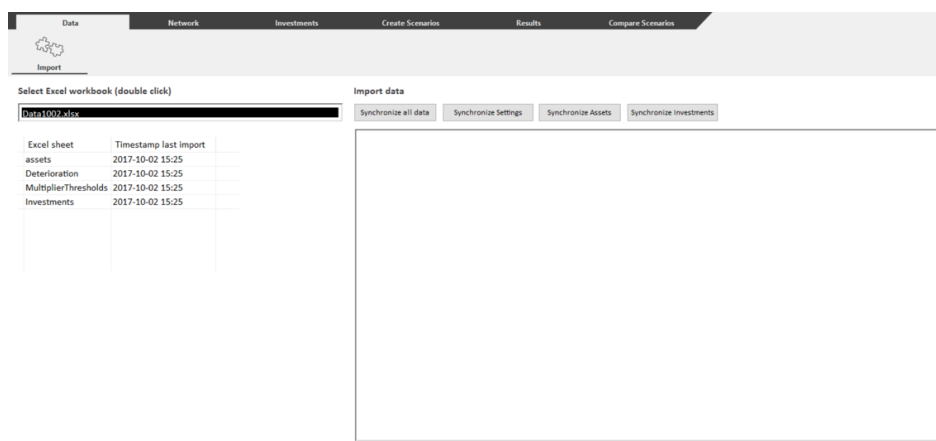


Figure A.1: Data import page from the decision support tool demo in AIMMS.

In presenting the current situation, a network overview is very helpful. Figure A.2 shows a very simple network representation, with some basic data. The attractiveness of the representation should be improved, as well as the use of colors and graphs for quick insight.

In Figure A.3, an overview of the proposed projects is shown. This is just a very basic representation, but could be used to check the imported data. Furthermore, this page could be used to enter

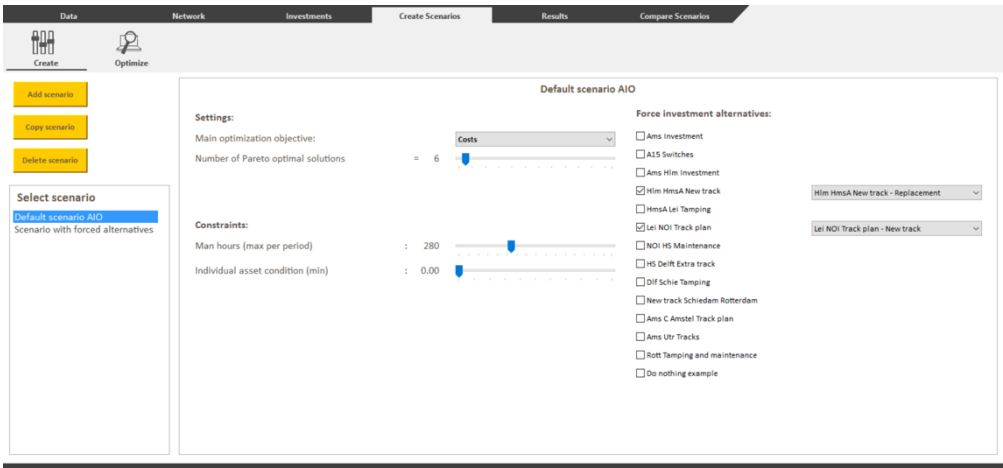


Figure A.4: Scenarios overview page from the decision support tool demo in AIMMS.

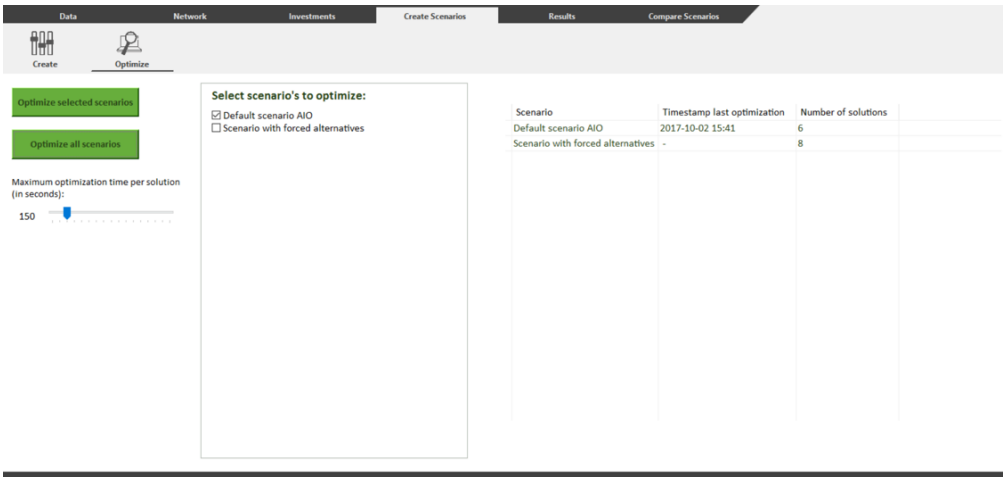


Figure A.5: Optimization settings page from the decision support tool demo in AIMMS.

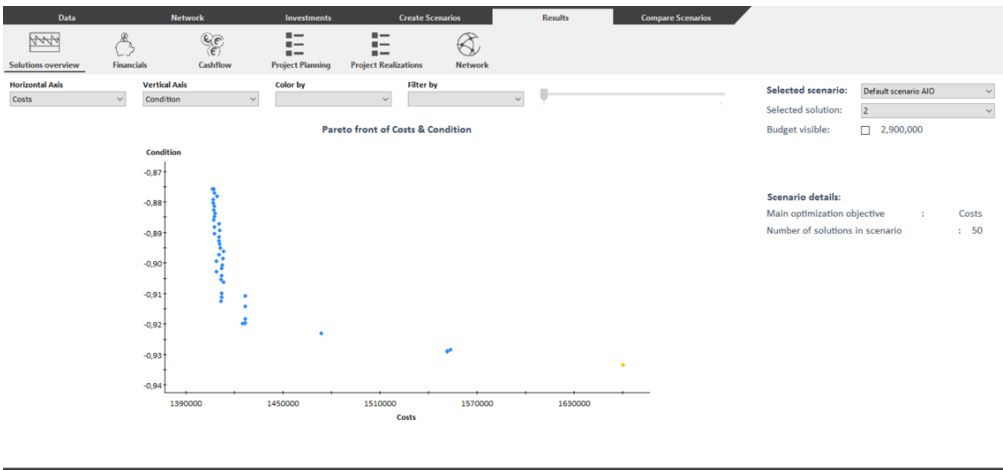


Figure A.6: Results in pairwise comparison plots, included on the results overview page from the decision support tool demo in AIMMS.

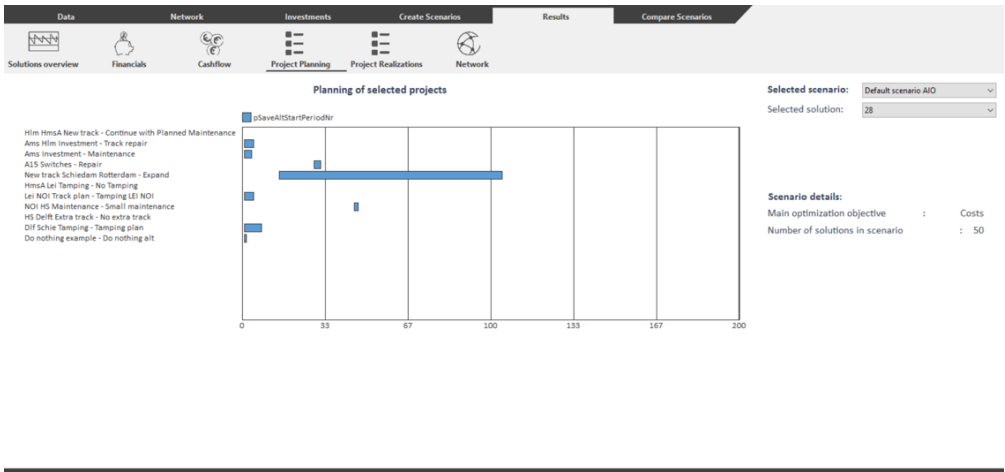


Figure A.7: Planning of projects of a selected project portfolio, included on the detailed results page from the decision support tool demo in AIMMS.

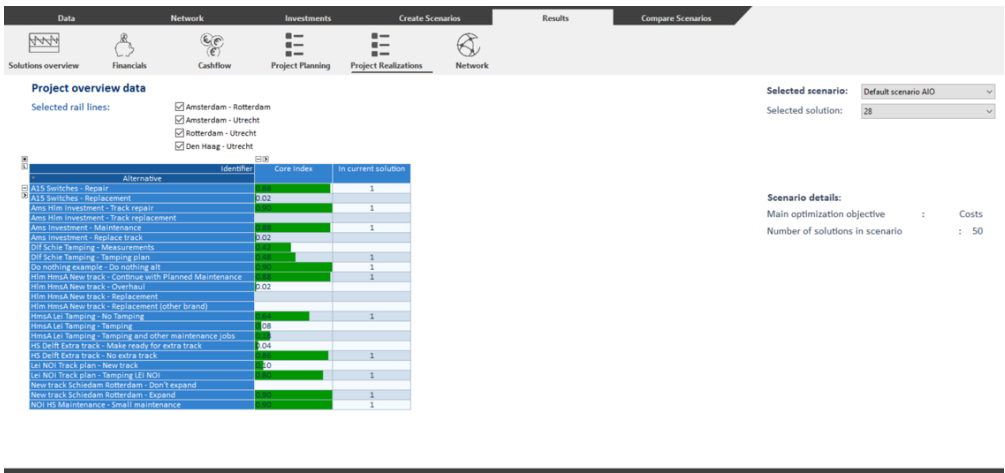


Figure A.8: The core index listed for the projects, included on the decision support page from the decision support tool demo in AIMMS.

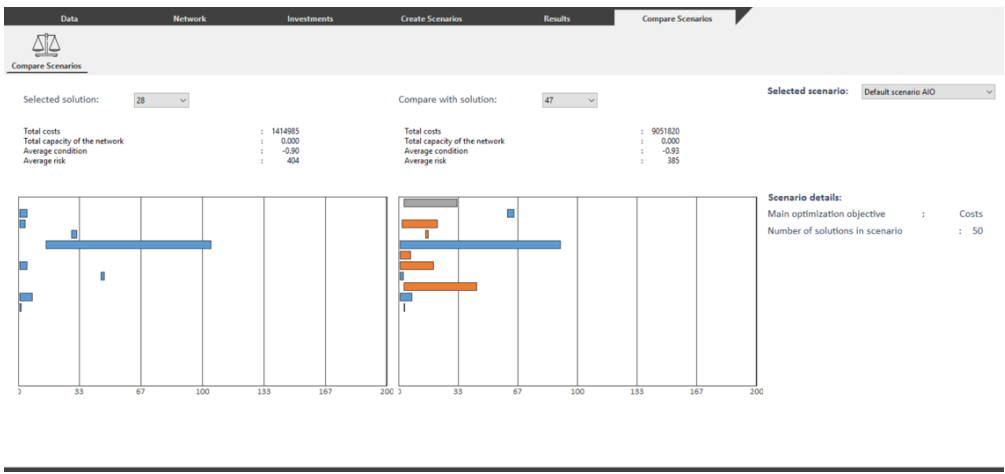


Figure A.9: Comparison of the planning and objective values for two project portfolios, included on one of the detailed results pages from the decision support tool demo in AIMMS.

Bibliography

- A. Ahern and G. Anandarajah. Railway projects prioritisation for investment: Application of goal programming. *Transport Policy*, 14(1):70–80, 2007.
- N.P. Archer and F. Ghasemzadeh. An integrated framework for project portfolio selection. *International Journal of Project Management*, 17(4):207–216, 1999.
- T.W. Athan and P.Y. Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2):155–176, 1996.
- Q. Bai, A. Ahmed, Z. Li, and S. Labi. A hybrid pareto frontier generation method for trade-off analysis in transportation asset management. *Computer-Aided Civil and Infrastructure Engineering*, 30(3):163–180, 2015.
- A. Berrichi, F. Yalaoui, L. Amodeo, and M. Mezghiche. Bi-objective ant colony optimization approach to optimize production and maintenance scheduling. *Computers & Operations Research*, 37(9):1584–1596, 2010.
- G. Budai, D. Huisman, and R. Dekker. Scheduling preventive railway maintenance activities. *Journal of the Operational Research Society*, 57(9):1035–1044, 2006.
- G. Budai-Balke. *Operations research models for scheduling railway infrastructure maintenance*. Number 456 in Tinbergen Institute Research Series. Rozenberg Publishers, 2009.
- A.F. Carazo, T. Gómez, J. Molina, A.G. Hernández-Díaz, F.M. Guerrero, and R. Caballero. Solving a comprehensive model for multiobjective project portfolio selection. *Computers & Operations Research*, 37(4):630–639, 2010.
- D.G. Carmichael. Computation of pareto optima in structural design. *International Journal for Numerical Methods in Engineering*, 15(6):925–929, 1980.
- A. Charnes and W.W. Cooper. Goal programming and multiple objective optimizations: Part 1. *European Journal of Operational Research*, 1(1):39–54, 1977.
- L. Chen, T.F.P. Henning, A. Raith, and A.Y. Shamseldin. Multiobjective optimization for maintenance decision making in infrastructure asset management. *Journal of Management in Engineering*, 31(6):04015015, 2015.
- C. Chikezie, A.T. Olowosulu, and O.S. Abejide. Multiobjective optimization for pavement maintenance and rehabilitation programming using genetic algorithms. *Archives of Applied Science Research*, 5(4):76–83, 2013.
- R.G. Cooper, S.J. Edgett, and E.J. Kleinschmidt. Portfolio management in new product development: Lessons from the leaders—i. *Research-Technology Management*, 40(5):16–28, 1997.
- I. Das and J.E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

- D. de Poot. A decision support model for project portfolio optimization (confidential). Master's thesis, Erasmus University Rotterdam, 2013.
- B. De Schutter and T. van den Boom. Optimization in systems and control. *Lecture notes for the course SC4091*, 2015.
- K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer. Ant colony optimization in multiobjective portfolio selection. In *Proc. 4th Metaheuristics International Conference*, pages 243–248, 2001.
- K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, and C. Stummer. Pareto ant colony optimization with ilp preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, 171(3):830–841, 2006.
- O. Engelhardt-Funke and M. Kolonko. Analysing stability and investments in railway networks using advanced evolutionary algorithms. *International Transactions in Operational Research*, 11(4):381–394, 2004.
- European-Union. Rail transport, June 2017. URL http://www.europarl.europa.eu/atyourservice/en/displayFtu.html?ftuId=FTU_5.6.6.html.
- T.F. Fwa, C.Y. Tan, and W.T. Chan. Road-maintenance planning using genetic algorithms. ii: Analysis. *Journal of Transportation Engineering*, 120(5):710–722, 1994.
- S.A. Gabriel, S. Kumar, J. Ordonez, and A. Nasserian. A multiobjective optimization model for project selection with probabilistic considerations. *Socio-Economic Planning Sciences*, 40(4):297–313, 2006.
- M. Gen and R. Cheng. *Genetic Algorithms and Engineering Design*, volume 2. John Wiley & Sons, 1997.
- L.E. Ghaoui, M. Oks, and F. Oustry. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations research*, 51(4):543–556, 2003.
- A.H. Ghapanchi, M. Tavana, M.H. Khakbaz, and G. Low. A methodology for selecting portfolios of projects with interactions and under uncertainty. *International Journal of Project Management*, 30(7):791–803, 2012.
- F. Ghasemzadeh and N.P. Archer. Project portfolio selection through decision support. *Decision Support Systems*, 29(1):73–88, 2000.
- F. Ghasemzadeh, N.P. Archer, and P. Iyogun. A zero-one model for project portfolio selection and scheduling. *Journal of the Operational Research Society*, 50(7):745–755, 1999.
- C.A. Grimes. Application of genetic techniques to the planning of railway track maintenance work. In *Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. GALESIA. First International Conference on (Conf. Publ. No. 414)*, pages 467–472. IET, 1995.
- Y.Y. Haimes, L.S. Ladson, and D.A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems Man and Cybernetics*, SMC-1(3):296, 1971.

- A. Higgins. Scheduling of railway track maintenance activities and crews. *Journal of the Operational Research Society*, 49(10):1026–1033, 1998.
- J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- C.L. Hwang and A.S.M. Masud. *Multiple objective decision making—methods and applications: a state-of-the-art survey*, volume 164 of *Lecture Notes in Economics and Mathematical Systems*. Berlin Springer-Verlag, 1979.
- J.G. Iniestra and J.G. Gutiérrez. Multicriteria decisions on interdependent infrastructure transportation projects using an evolutionary-based framework. *Applied Soft Computing*, 9(2):512–526, 2009.
- ISO. Iso 55000 : 2014, 2014. URL <https://www.iso.org/standard/55088.html>.
- M. Jafarzadeh, H.R. Tareghian, F. Rahbarnia, and R. Ghanbari. Optimal selection of project portfolios using reinvestment strategy within a flexible time horizon. *European Journal of Operational Research*, 243(2):658–664, 2015.
- C.P. Killen, R.A. Hunt, and E.J. Kleinschmidt. Managing the new product development project portfolio: a review of the literature and empirical evidence. In *Portland International Center for Management of Engineering and Technology*, pages 1864–1874. IEEE, 2007.
- Z. Li and K.C. Sinha. Application of Shackle’s model and system optimization for highway investment decision making under uncertainty. *Journal of Transportation Engineering*, 135(3):129–139, 2009.
- J. Liesiö, P. Mild, and A. Salo. Preference programming for robust portfolio modeling and project selection. *European Journal of Operational Research*, 181(3):1488–1505, 2007.
- R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
- G. Mavrotas. Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465, 2009.
- A. Messac and C.A. Mattson. Generating well-distributed sets of pareto points for engineering design using physical programming. *Optimization and Engineering*, 3(4):431–450, 2002.
- V. Mohagheghi, S.M. Mousavi, and A. Siadat. A new approach in considering vagueness and lack of knowledge for selecting sustainable portfolio of production projects. In *Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference on*, pages 1732–1736. IEEE, 2015.
- G. Morcoux and Z. Lounis. Maintenance optimization of infrastructure networks using genetic algorithms. *Automation in Construction*, 14(1):129–142, 2005.
- M. Nourelfath, M.C. Fitouhi, and M. Machani. An integrated model for production and preventive maintenance planning in multi-state systems. *IEEE Transactions on Reliability*, 59(3):496–506, 2010.
- T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *Evolutionary Computation, 2003. CEC’03. The 2003 Congress on*, volume 2, pages 878–885. IEEE, 2003.

- L. Podofillini, E. Zio, and J. Vatn. Risk-informed optimisation of railway tracks inspection and maintenance procedures. *Reliability Engineering & System Safety*, 91(1):20–35, 2006.
- G. Quan, G.W. Greenwood, D. Liu, and S. Hu. Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms. *European Journal of Operational Research*, 177(3):1969–1984, 2007.
- M. Rabbani, M.A. Bajestani, and G.B. Khoshkhou. A multi-objective particle swarm optimization for project selection problem. *Expert Systems with Applications*, 37(1):315–321, 2010.
- J. Sadeghi and H. Askarinejad. Development of improved railway track degradation models. *Structure and infrastructure engineering*, 6(6):675–688, 2010.
- M. Sakawa, H. Yano, and T. Yumine. An interactive fuzzy satisficing method for multiobjective linear-programming problems and its application. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(4):654–661, 1987.
- C. Sriskandarajah, A.K.S. Jardine, and C.K. Chan. Maintenance scheduling of rolling stock using a genetic algorithm. *Journal of the Operational Research Society*, 49(11):1130–1145, 1998.
- I. Stamelos and L. Angelis. Managing uncertainty in project portfolio cost estimation. *Information and Software Technology*, 43(13):759–768, 2001.
- C. Stummer and K. Heidenberger. Interactive R&D portfolio analysis with project interdependencies and time profiles of multiple objectives. *IEEE Transactions on Engineering Management*, 50(2):175–183, 2003.
- Z. Su, A. Jamshidi, A. Núñez, S. Baldi, and B. De Schutter. Multi-level condition-based maintenance planning for railway infrastructures—a scenario-based chance-constrained approach. *Transportation Research Part C: Emerging Technologies*, 84:92–123, 2017.
- J.Y. Teng and G.H. Tzeng. A multiobjective programming approach for selecting non-independent transportation investment alternatives. *Transportation Research Part B: Methodological*, 30(4):291–307, 1996.
- Y. Tian, R. Cheng, X. Zhang, and Y. Jin. Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):73–87, Nov 2017. ISSN 1556-603X. doi: 10.1109/MCI.2017.2742868.
- T. Tušar and B. Filipič. Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosecution method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245, 2015.
- K. Tzanakakis. *The railway track and its long term behaviour: a handbook for a railway track of high quality*, volume 2. Springer Science & Business Media, 2013.
- Asset Management Working Group UIC. Asset management, 2017. URL <http://uic.org/asset-management#UIC-Asset-Management-Working-Group>.
- D.A. Van Veldhuizen and G.B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 204–211. IEEE, 2000.
- G.A. Vijayalakshmi Pai. Fuzzy decision theory based metaheuristic portfolio optimization and active rebalancing using interval type-2 fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 2016.

- M.R. Walls. Combining decision analysis and portfolio management to improve project selection in the exploration and production firm. *Journal of Petroleum Science and Engineering*, 44(1): 55–65, 2004.
- L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, 1963.
- E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.