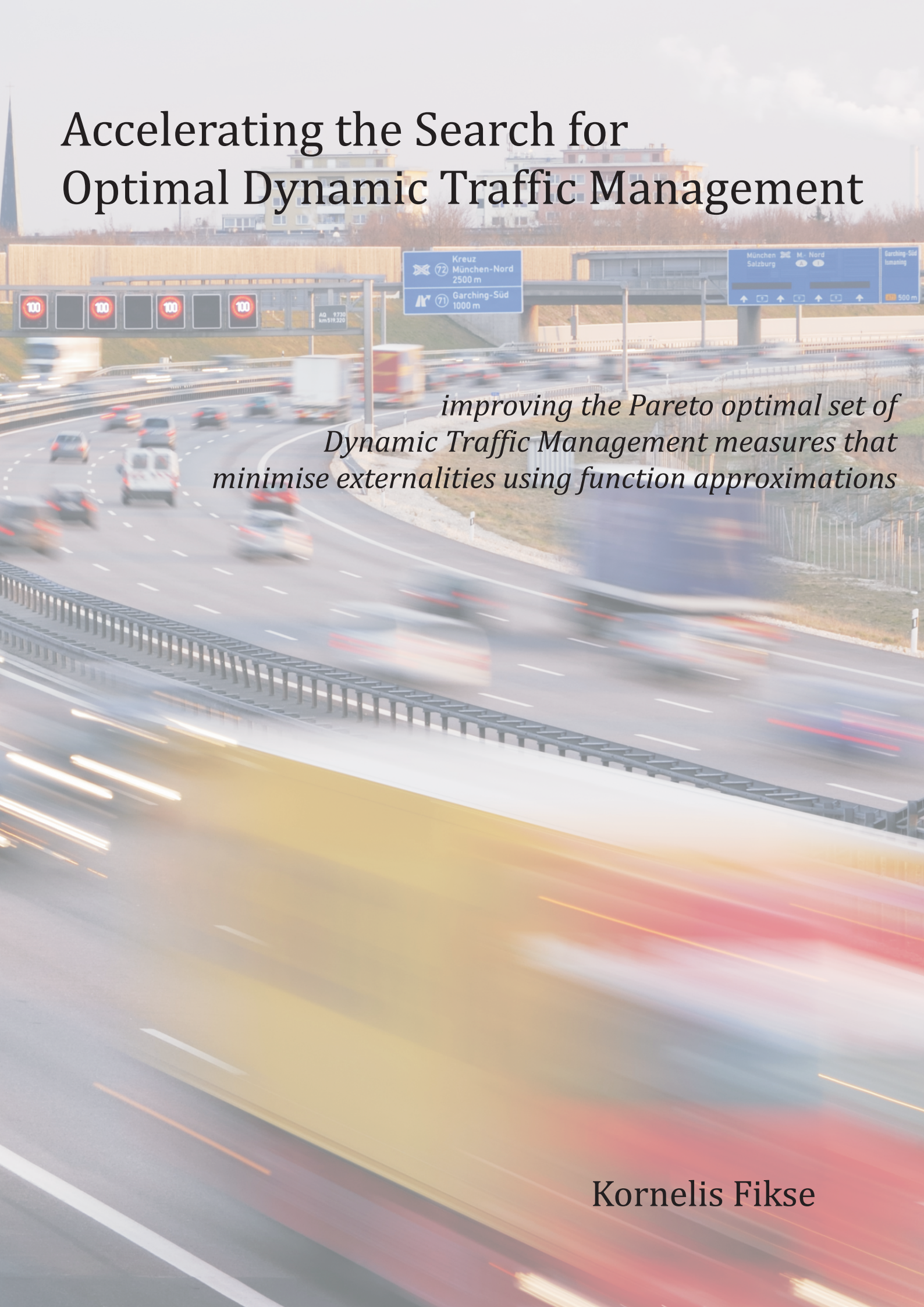
Accelerating the Search for Optimal Dynamic Traffic Management



improving the Pareto optimal set of Dynamic Traffic Management measures that minimise externalities using function approximations

Kornelis Fikse

Accelerating the Search for Optimal Dynamic Traffic Management

improving the Pareto optimal set of Dynamic Traffic Management measures that minimise externalities using function approximations

Kornelis Fikse

Enschede, 3rd January 2011

In fulfilment of the Master Degree
Civil Engineering & Management, University of Twente, The Netherlands

Graduation Committee

Prof. dr. ir. E.C. van Berkum

Dr. M.C.J. Bliemer

Dr. T. Thomas

Ir. L.J.J. Wismans

UNIVERSITY OF TWENTE.

Goudappel Coffeng
Consultants on traffic and transport

Summary

I don't think many people have ever read the report [...]
How many read the summary?

John Sherman Cooper (1901 – 1991)

In the past decades traffic demand has been increasing nearly continuously, which has provided governments all over the world with significant challenges. In the Netherlands constructing new roads is, due to various reasons, not longer considered to be the solution, the focus is now more on efficient use of existing infrastructure.

One of the instruments that is frequently used to increase the efficiency of infrastructure is Dynamic Traffic Management (DTM). In DTM we use different measures such as directing traffic through traffic lights, adding or removing lanes and variable speed limits to provide road users with the 'best possible' infrastructure. It is however difficult to determine what is 'best', especially now environmental and safety issues are becoming more and more important. The best possible set of measures from a travel time perspective, may very well result in very high CO_2 emissions, annoyance due to excessive noise and many fatalities.

It is therefore that research is being done on determining a set of possible DTM applications that can be considered the best solutions. Here 'best' means that these solutions are not outperformed by any other solution on all objectives. Unfortunately finding all solutions in this set is impossible, it would easily take millennia to find them. Science has therefore resorted to finding only a part of this set (but a representative one) using heuristics such as Genetic Algorithms. However finding a part of this set using this method still takes months, which is unacceptable in the traffic and transport consultancy business. It is here where our research takes off.

Main goal of our research is therefore to accelerate the search for this set of best solutions (also known as Pareto optimal set). In our research we focus

solely on accelerations that can be obtained by using approximation techniques, which is why our research goal is defined as *'accelerating the search for the Pareto optimal set found by multiobjective genetic algorithms for multiobjective network design problems, in which externalities are the objectives and DTM measures the decision variables, using function approximations'.*

It is therefore that we performed a literature study into approximation techniques, from which we derived three main techniques: the Response Surface Method (RSM), the Radial Basis Function (RBF) and Kriging/DACE. Because all of the approximation techniques have parameters that can be set, we were able to develop 148 different variants. In order to be able to determine which variant would provide the best results, we chose two simple road networks which could be used for testing and selected a set of quality measures from literature.

We found that variants that score very good on one quality measure, do not necessarily perform well on another. Furthermore we found that selecting the right parameters can significantly influence the results of the approximation techniques. However eventually we can conclude that the Kriging/DACE approach without optimising the power in the cost function is always amongst the best performing approaches. Benefit of the Kriging/DACE approach is that it does not only provide estimated objective values, but also the corresponding estimated errors. Another solution which performs reasonably well, and best on one quality measure, is the RSM approach with only cubic squared interaction terms. Main benefits of the latter approach are that it is easy to understand (it is the basis of the Least Squares Method) and that the approach is extremely fast (it can determine objective values in less than a second). It is therefore that we selected these two approaches as possible approximation methods for the remainder of the research.

We also performed a literature study into how Genetic Algorithms (and NSGA-II in particular) can be accelerated. It became clear that many of the approaches are quite complicated and/or require further optimisation, which would lead to high computational effort. We therefore selected two approaches which could easily be integrated into the original NSGA-II algorithm. The first is the Inexact Pre Evaluation (IPE) which is a deterministic approach and evaluates only those solutions which are, based on the approximated objective values, part of the Pareto optimal set. The second is the Probability of Improvement (PoI) approach, which is stochastic and determines for each solution the probability that it improves the Pareto optimal set. Next it only evaluates the n best solutions or the solutions with a probability higher than $x\%$.

We combined the two approximation methods (RSM and DACE) and the two acceleration approaches (IPE and PoI) into three different Approximation

Method Assisted NSGA-II (AMAN) algorithms. The fourth combination was impossible since PoI requires the expected error for each objective value and RSM is not able to provide this information. In order to determine which of the three approaches is best, we performed a literature study to find performance measures which can be used to compare Pareto fronts, and applied the approaches to the two test networks mentioned earlier. Unfortunately we only had time for a single run, which makes that the results are not indisputable.

We found that the results between the different AMANs (when compared with the original NSGA-II algorithm) do not point towards a single 'best' approach. In fact, an approach that scores well on one performance measure can easily score quite bad on another. However based on the combined results over the two test networks, we find that PoI-DACE provides the most promising results. Not only did it provide results that were comparable to the results of the original NSGA-II algorithm, it also provided those results in only 50% of the time that was needed by the NSGA-II algorithm. It is therefore that we selected this approach to be used in the last phase of this research.

In the last phase we tested the PoI-DACE algorithm on the (more realistic) case of Almelo. In this network we had seven controlled traffic lights and two sections of motorway with variable speed limits. In order to determine the performance of the PoI-DACE approach (in comparison with the original NSGA-II algorithm) we used the performance measures which were also used for comparing the AMANs on the test networks. Due to the fact that performing a run for both the NSGA-II and the AMAN algorithm takes about three weeks, we were, again, only able to perform a single run.

The results of the analysis were quite promising. The area that was dominated by the NSGA-II, but not by the AMAN was only 3% of the total area dominated by the NSGA-II algorithm. Furthermore we found that the spread of solutions over the Pareto front was better and that a reduction of 30% in calculation time is realisable. Unfortunately we also found that the influence of stochasticity (there are a lot of random processes involved in NSGA-II), is significant. In order to reduce the uncertainty in these conclusions, we would have to perform dozens, if not hundreds, of runs.

We furthermore tried to interpret the Pareto optimal set that was found from a traffic and transport engineering perspective, which appeared to be a difficult task. Using grouped data and a multitude of boxplots we could, for some of the DTM measures, determine a relation between the settings and the resulting objective values. Unfortunately we were not able to find correlation effects between different DTM measures, something that might be caused by a lack of data.

Based on the results on the different test networks and the Almelo case we find that it is highly likely that the proposed AMAN (and probably also the other AMANs) can achieve a Pareto front that is comparable to the one found by

NSGA-II. Besides PoI-DACE is able to do so with a reduction in calculation time of 30%. We therefore can state that we can indeed accelerate the search for the Pareto optimal set by applying approximation techniques.

It does however seem wise to do some further research. Especially the performance of AMANs can be disputed, since only a single run has been performed. In order to provide reliable results at least dozens of runs should be performed before we can conclude, statistically, that a specific AMAN is equal to the original NSGA-II algorithm.

We also recommend that the behaviour of the PoI approach, or more specifically the change of approximated values and errors over time, is studied. We were unable to apply a 'better than x% policy' because it appeared that after a few iterations all solutions were accepted.

Finally we suggest that more time and effort is spend in analysing the resulting Pareto front. Unfortunately we were unable to detect important relationships between DTM measures, however that might be possible if sufficient data and time is available.

Nederlandse Samenvatting

De boodschap is vaak omgekeerd evenredig
met de dikte van het boek [...]
de essentie zou je in twee A-viertjes
kunnen samenvatten.

Doede Keuning (1943 –)

In de afgelopen jaren is de verkeersvraag sterk toegenomen, niet alleen in Nederland, maar ook in de rest van de wereld. Om de bijbehorende problemen het hoofd te bieden kan de Nederlandse overheid zich, mede door de Europese milieuwetgeving, niet langer richten op de aanleg van nieuwe wegen zoals vroeger gebruikelijk was. De focus ligt daarom nu op het efficiënter gebruiken van de bestaande infrastructuur.

Een van de technieken die daarvoor wordt ingezet is Dynamisch Verkeers Management (DVM). DVM maakt gebruik van verkeerslichten (VRI's) om verkeersstromen te beïnvloeden, matrixborden om het aantal rijbanen of de maximale toegestane snelheid te veranderen en Dynamische Route Informatie Panelen (DRIPs) om de weggebruikers te voorzien van hoogwaardige informatie over de toestand van het wegennet. Het uiteindelijke doel van de wegbeheerder is een zo optimaal mogelijk verkeersnetwerk te presenteren voor de gebruikers. De vraag is echter wat een 'optimaal' verkeersnetwerk is; de set met maatregelen die leidt tot een minimale reistijd kan tevens de oplossing zijn die leidt tot enorme CO_2 uitstoot, veel geluidsoverlast en een groot aantal verkeersslachtoffers.

Op dit moment wordt daarom onderzoek gedaan om een verzameling oplossingen te bepalen, die gezamenlijk als 'beste' aangemerkt kunnen worden. Kortom, voor elke oplossing binnen deze verzameling bestaat er geen alternatief dat beter scoort op alle doelfuncties. Deze verzameling kan echter zeer groot zijn en het duurt daarom millenia voordat deze is gevonden. Er wordt daarom vaak gebruik gemaakt van intelligente heuristieken, zoals Genetische

Algoritmen, om een representatieve deelverzameling te vinden. Het vinden van een dergelijke deelverzameling duurt echter nog steeds maanden en dat is onacceptabel in de verkeerskundige advieswereld.

Het hoofddoel van dit onderzoek is dan ook om de zoektocht naar deze verzameling beste oplossingen (beter bekend als Paretoverzameling) te versnellen. Het onderzoek beperkt zich echter tot versnellingen die bereikt kunnen worden door middel van approximatie technieken. De doelstelling is daarom gedefinieerd als: *'het versnellen van de zoektocht naar de Paretoverzameling voor netwerkontwerp problemen met meervoudige doelfuncties zoals die door de Genetische Algoritmen voor meervoudige doelfuncties gevonden worden, waar de externe effecten van verkeer de doelfuncties zijn en de DVM maatregelen de beslissingsvariabelen, gebruik makend van functie benaderingen'.*

Het onderzoek begint daarom met een literatuurstudie naar approximatie technieken op basis waarvan er drie zijn geselecteerd, te weten: de Response Surface Method (RSM), Radial Basis Function (RBF) en Kriging/DACE. Op basis van deze drie hoofdtechnieken zijn in totaal 148 verschillende approximatie methoden ontwikkeld die vervolgens getest zijn op twee test netwerken. De kwaliteit van de benaderingen is getest aan de hand van in de literatuur beschreven criteria.

Uit het onderzoek blijkt dat methoden die zeer goed scoren op een van de criteria, niet noodzakelijkerwijs ook goed scoren op een ander. Verder bleek dat de gekozen parameters de kwaliteit van de antwoorden sterk beïnvloeden. We kunnen concluderen dat de Kriging/DACE methoden waarbij de macht in de kostenfunctie niet wordt geoptimaliseerd vrijwel altijd het beste te scoren. Een ander groot voordeel van deze methode is dat deze niet alleen de verwachte doelfunctiewaarde maar ook de bijbehorende voorspelfout genereerd. Daarnaast bleek dat de meest eenvoudige methode, RSM met alleen kwadratische interactietermen, vaak redelijk goede voorspellingen geeft. Voordelen van deze methodiek zijn dat deze eenvoudig uit te leggen is (het vormt de basis van de kleinste-kwadratenmethode) en dat deze erg snel is (resultaten kunnen binnen een seconde bepaald worden). Mede op basis van deze conclusies zijn beide technieken uitgekozen om in het vervolg van dit onderzoek gebruikt te worden.

Daarnaast is onderzoek gedaan naar de manier waarop deze benaderde functiewaarden gebruikt kunnen worden binnen de bestaande Genetische Algoritmen (en NSGA-II in het bijzonder). Het werd vrij snel duidelijk dat veel methoden te gecompliceerd zijn of verdere optimalisatie vragen, wat de rekentijd alleen maar doet toenemen. Daarom is voor twee relatief eenvoudige methoden gekozen. De eerste methode is deterministich en is de Inexact Pre Evaluation (IPE), waarbij alleen die oplossingen exact worden geëvalueerd die op basis van de benaderde doelfunctiewaarden deel uitmaken van de Paretoverzameling. De tweede methode is stochastisch en is de Probabilty of Improvement

(PoI) methode waarbij voor elke oplossing de kans wordt bepaald dat deze deel uitmaakt van de Paretoverzameling. Vervolgens worden alleen de n beste oplossingen of de oplossingen met een kans groter dan $x\%$ exact geëvalueerd.

De twee approximatietechnieken (RSM en DACE) en de twee versnellingsmethoden (IPE en PoI) zijn vervolgens gecombineerd tot een drietal Approximation Method Assisted NSGA-II aproaches (AMANs). De vierde combinatie was niet mogelijk omdat RSM geen voorspelfout bepaald en deze wel benodigd is voor de stochastische Probability of Improvement methode. Om te kunnen bepalen welke methode de beste is, is er in de literatuur gezocht naar kwaliteitscriteria voor Paretoverzamelingen, waarna de methoden zijn toegepast op de eerder genoemde test netwerken. Helaas was er niet voldoende tijd voor meerdere 'runs', waardoor de resultaten onzeker zijn.

Het bleek onmogelijk om uit de resultaten een beste methode te kiezen. Sterker, ook hier bleek dat een methode die goed scoort op het ene criterium niet per definitie goed scoort op het andere. Echter alle resultaten in ogenschouw nemend, kan geconcludeerd worden dat de PoI-DACE methode de beste resultaten lijkt op te leveren. Niet alleen leek de Paretoverzameling sterk op die van de originele NSGA-II, ook bleek dat deze resultaten haalbaar waren in 50% van de rekentijd die het originele GA nodig had.

In de laatste fase van dit onderzoek is daarom de PoI-DACE methode toegepast op de (meer realistische) situatie van Almelo. Dit netwerk bestaat uit zeven geregelde VRI's en twee trajecten op de snelweg waar door middel van matrixborden de maximale snelheid aangepast kan worden. Vanwege de beperkte beschikbare tijd is ook hier slechts een 'run' uitgevoerd.

De resultaten bleken veelbeloved. Het gedeelte van de doelfunctieruimte dat werd gedomineerd door NSGA-II maar niet door de AMAN was slechts 3% van het totale gebied dat door NSGA-II werd gedomineerd. Daarnaast bleek dat de oplossingen beter over de doelfunctieruimte verdeeld waren en dat een rekentijdreductie van 30% haalbaar was. Helaas is de invloed van stochasticiteit aanzienlijk, waardoor er tientallen, zo niet honderden, 'runs' nodig zijn om statistisch betrouwbare resultaten te kunnen presenteren.

Daarnaast is getracht om de Paretoverzameling te interpreteren vanuit een verkeerskundig oogpunt, iets wat niet eenvoudig bleek. Door de data te groeperen konden er boxplots gemaakt, waarmee voor sommige DVM maatregelen een relatie tussen de doelfuncties en de instelling van de DVM maatregel aangetoond kon worden. Aantonen dat de instellingen van twee DVM maatregelen en de doelfunctiewaarden gecorreleerd zijn bleek echter, waarschijnlijk mede door een gebrek aan data, niet mogelijk.

Op basis van de resultaten van de test netwerken en de 'case' Almelo kunnen we concluderen dat het zeer waarschijnlijk is dat de voorgestelde AMAN (en mogelijkerwijs ook andere AMANs) een Paretoverzameling kunnen berei-

ken die vergelijkbaar is met de Paretoverzameling die door NSGA-II wordt gevonden. Daarnaast blijkt dat PoI-DACE dat kan in slechts 70% van de tijd die NSGA-II daarvoor nodig heeft. We kunnen daarom stellen dat we de zoektocht naar de Paretoverzameling inderdaad kunnen versnellen door het gebruik van approximatie technieken.

Het is echter noodzakelijk om meer onderzoek te doen, vooral op het gebied van de kwaliteit van de AMANs. In dit onderzoek zijn alle conclusies gebaseerd op een enkele 'run' terwijl tientallen, of honderden, 'runs' nodig zijn voordat statistisch juiste conclusies getrokken kunnen worden.

Daarnaast lijkt het verstandig om meer onderzoek te doen naar hoe de benaderde functiewaarden en voorspelfouten zich gedragen in de PoI methode. In dit onderzoek bleek het namelijk niet zinvol om een 'beter dan x% politiek' toe te passen, aangezien dit leidde tot het evalueren van alle oplossingen.

Tenslotte wordt aanbevolen om meer tijd en moeite te steken in het analyseren van de uitkomst, de Paretoverzameling. Het bleek in dit korte tijdsbestek niet mogelijk om duidelijke relaties te vinden tussen de DVM maatregelen, iets wat wellicht wel mogelijk is als er meer data en tijd beschikbaar is.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AE	Algorithmic Effort
AMAN	Approximation Model Assisted NSGA-II
ATS	Automated Traffic Control Signal (traffic light)
DTA	Dynamic Traffic Assignment
DTM	Dynamic Traffic Management
EA	Evolutionary Algorithm
EI	Expected Improvement
eRBF	extended Radial Basis Function
FAS	Fraction of Acception Solutions
GA	Genetic Algorithm
GTC	Generalised Travel Cost
IPE	Inexact Pre Evaluation
MAE	Mean Average Error
MAEA	Metamodel Assisted Evolutionary Algorithm
MLE	Maximum Likelihood Estimation
MOEA	Multiobjective Evolutionary Algorithm
MOGA	Multiobjective Genetic Algorithm
NDP	Network Design Problem
NSGA	Non-dominated Sorting Genetic Algorithm

PoI	Probability of Improvement
POS	Pareto optimal set
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
RNI	Rate of Non-dominated Individuals
RSM	Response Surface Method
SA	Simulated Annealing
SO	System Optimum
STA	Static Traffic Assignment
SUE	Stochastic User Equilibrium
TS	TABU Search
TT	Travel Time
TTT	Total Travel Time
VLS	Variable Lane Sign
VMS	Variable Message Sign
VSS	Variable Speed Sign

Chapter 1

Introduction

Traffic is only one of the side effects of growth.

Roy Barnes (1948 –)

The quote by Roy Barnes can, in a way, be considered the starting point of this research. Due to the continuous economic growth the demand for traffic has been increasing over the past decades. Not just in the United States of America, to which Roy Barnes probably was referring, but also in Europe and especially in a densely populated area such as the Netherlands.

In the past the solution to the traffic demand problem was found in constructing new infrastructure, but this is no longer a viable option as we will infer in section 1.1. The solution that is currently in favour, the use of Dynamic Traffic Management, brings along some other challenges. One of the problems is that there are many different ways in which Dynamic Traffic Management can be applied and we therefore have to define which solutions are considered to be optimal.

In order to be able to determine the effect of different solutions of Dynamic Traffic Management, we first have to define a framework which can be used to model Dynamic Traffic Management measures. In section 1.2 we will therefore explain why the Network Design Problem is a suitable framework for modelling DTM measures. Unfortunately we will also show that it is virtually impossible to find optimal solutions, which is why we have to resolve to algorithms to find good solutions. Consequently we introduce three different algorithms in section 1.3 and will elaborate more on one specific family of algorithms, which are the Genetic Algorithms.

Up to this point we have been diverging the subjects of our research to an extend where we are unable to complete the research within a reasonable period of time. In section 1.4 we therefore determine the scope of this research, by limiting the number of objectives that we are trying to attain. Furthermore

we will select a single modelling framework (from section 1.2) and a single algorithm (from section 1.3) with which we will continue our research.

Something that is probably just as important, is defining the main goal of this research. We therefore first have to determine which problems we can identify and decide how we would like to solve these problems. In section 1.5 we will accordingly briefly discuss two problems that we have identified and select one specific problem, after which the (main) goal of this research can be formulated.

It is at this point that we can, using the results from section 1.4 and 1.5, determine which subjects are relevant for the remainder of the research. In section 1.6 we therefore start by creating a research model, which provides an overview of the different subjects we need to study in detail. In section 1.7 we continue by defining the questions that have to be answered, before we have enough knowledge about the subjects from the research model. Finally in section 1.8 we will explain how we will obtain the information to answer the research questions.

Finally, having defined the main goal of our research and the strategy which we will follow to attain this goal, we will provide an outline of this thesis (section 1.9). In this outline we will explain where you are able to find the answers to the different research questions, and as such where the different subjects are discussed.

Let us now start by introducing the Dutch problem and Dynamic Traffic Management.

1.1 The Dutch Road Network & Dynamic Traffic Management Measures

In the past decade(s) the Dutch road network has become increasingly busy and traffic-jams are a day-to-day practice for most commuters. In the past these problems might have been tackled by expanding the existing road network by constructing new roads or expanding existing ones. European legislation, however, restricts the construction of new roads, by enforcing new rules concerning air and noise pollution. Furthermore there are problems related to the increasing costs of expanding road networks, the time that is required before work can actually start, and a lack of space. Dutch authorities have therefore resolved to using the existing road network more efficiently rather than expanding the current road network.

One of the options that is quite popular in the Netherlands is the use of Dynamic Traffic Management (DTM). Dynamic Traffic Management is a term that is used to describe many different (time or traffic dependent) measures that influence the characteristics of the road network or the behaviour of road users. There are (generally speaking) two different types of DTM measures,

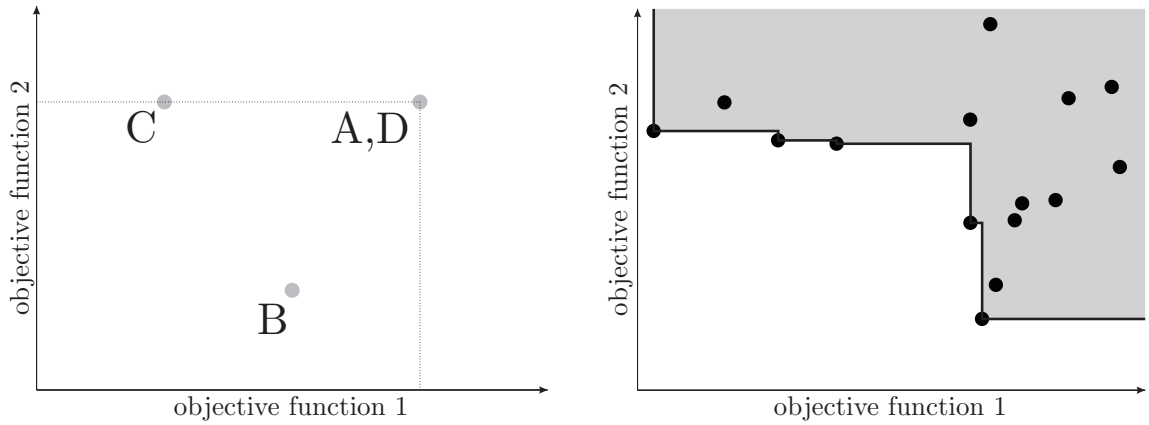
the first is the DTM measure that can be adjusted quite swiftly (but not instantaneous). The most widely known example of such a DTM measure is the Automated Traffic Control Signal (ATS). It is quite easy to change the settings of an ATS (which influences the capacity of the crossing in a certain direction) but this is rarely done in real-time.[1] The second type of DTM measure is able to make changes instantly, thus enabling the authorities to react upon the current state of the network (real-time adjustment of the DTM measure), or can be made in a quite short period of time. One of the most commonly used examples of this type of DTM measures is the so-called Variable Message Sign (VMS). These signs can be used to limit or increase the number of lanes ('crossing off' lanes, allowing shoulder lanes to be used) which directly influences the capacity of a specific road section, impose variable speed limits and provide travel time, traffic-jam and other information to road users which they can use to alter their route choice. ATS and VMS are therefore amongst the most powerful tools in directing traffic.

In order to determine the resulting traffic conditions of a solution usually a Dynamic Traffic Assignment (DTA) is used, which propagates traffic through a network, simulating the behaviour of traffic over a period of time. These DTA models are well suited to predict the results of different DTM measures, as long as they influence the characteristics of the network (i.e. they should influence speed or capacity of a specific road section). Although DTA models can also be used to predict the effects of non-network changing DTM measures, such as advanced traffic information, this does require a good behavioural model, which is often not available.[2] Using the results of these DTAs (flows and speeds on road sections) the effects on travel time, air and noise pollution and road safety (or other objectives) can be estimated.

As mentioned earlier it is possible to use DTM to influence the behaviour of road users in real-time. However it is also possible to use DTM to provide the road users (in fact all those involved) with the 'best' road network possible. In that case for each time of the day a decision should be made concerning the settings of the DTM measures, a so called 'strategic' policy. Deciding which DTM measures should be implemented and when (which is what makes them dynamic) is one of the most difficult decisions in traffic engineering. Good examples of such 'strategic' policies are the speed limits of 100 and 80 km/h on motorways around major cities and the use of additional lanes during peek hours. However the application of these measures seems quite arbitrary. The measures are implemented to attain a single objective, for instance reduction

[1] The ATS under consideration here is the ATS with a fixed cycle, the more and more common ATS with detection loops do of course adapt their cycle in real-time.

[2] The behavioural model is here defined as a model that predicts which fraction of people is going to react in which way on the information provided. The 'ordinary' network-changing DTM measures only require a model that determines the effect of the changes on the utility of a specific route, since model split and route choice (see e.g. Ortúzar & Willumsen, 2001) are usually based on utility functions.



(a) Example of Dominance

(b) Example of Pareto front

Figure 1.1: Dominance and Pareto fronts

of noise, reduction of air pollution, reduction of travel times or (although less frequently used) reduction of the number of casualties and fatalities. Therefore the question arises whether the DTM measures, that are currently applied, might have a deteriorating effect on other objectives. Therefore research has started that tries to find a set of possible settings for DTM measures (for a certain problem area), which are not dominated by other solutions.

In order to understand which solutions are called non-dominated we first have to study the concept of dominance. We will explain dominance using Figure 1.1a. Let i be the index of the objective functions, a and b are two solutions and $f_i(a)$ is the objective value for solution a on objective i. Furthermore assume a minimisation problem. First there is the concept of *weak dominance*, we say that a weakly dominates b (denoted by $a \succeq b$) when $\forall\, i\ f_i(a) \leq f_i(b)$. In Figure 1.1a this means that B, C and D all weakly dominate A. Next there is *dominance*, a is said to dominate b (denoted by $a \succ b$) when $\forall\, i\ f_i(a) \leq f_i(b) \wedge \exists\, i : f_i(a) < f_i(b)$. In Figure 1.1a we can therefore say that both B and C dominate A. Finally there is *strong dominance*, a strongly dominates b (denoted by $a \succ\succ b$) when $\forall\, i\ f_i(a) < f_i(b)$. In Figure 1.1a B strongly dominates A.

Back to our original problem we can now state that we are looking for solutions b that are not dominated, i.e. $a \not\succ b$. We do explicitly allow solutions to be weakly dominated. We can now construct a so called Pareto front (black line in Figure 1.1b) from all non dominated solutions. Furthermore we find an area (grey) that is dominated by the solutions in our Pareto front, i.e. solutions in this dominated area can be improved by using one of the solutions that is on the Pareto front instead.

In the next section we will explain why (and how) our problem can be described as a Network Design Problem. Furthermore we will provide a brief overview of traffic and transport related Network Design Problems in literature.

1.2 Network Design Problems

A formal (mathematical) Network Design Problem (NDP) usually starts with a given (un)directed graph $G = (V, E)$ a cost c_e for each $e \in E$ (or for each arc in the directed case), and we like to find a minimum cost subset E' of the edges E that meets certain design criteria. The problem described above (selecting DTM measures in order to attain certain objectives) can easily be translated to a directed NDP. The graph G consists of a set of links (E), which are connected to each other at a vertex (V). In this case each DTM measure adds one or more arcs e to an edge E, which gives the possibility to select a subset E' that optimises the objectives.

Literature suggests two ways of modelling the design variables, either discrete using the Discrete NDP (DNDP) or continuous using the Continuous NDP (CNDP). The DNDP models are used when the construction of new links (or even complete networks) is considered (see: Poorzahedy & Turnquist, 1982; Drezner & Wesolowsky, 2003; Gao, Wu & Sun, 2005), whilst the CNDP models are used when only the expansion of existing links (e.g. a change in capacity or maximum speed) is considered (see: Meng, Yang & Bell, 2001; Chiou, 2005; Zhang & Lu, 2007; Mathew & Sharma, 2009; Xu, Wei & Wang, 2009; Chen, Kim, Lee & Kim, 2010). However the expansion of an existing link is often a discrete problem, one either adds another lane or one does not. In that sense the use of a CNDP can be considered a relaxed version of the problem, which is why DNDP models can also be used to model expansion problems (see: LeBlanc & Abdulaal, 1978; Boyce & Janson, 1980). It is therefore that we decided to model our problem as a DNDP.

Our problem should be described as a bilevel optimisation problem (bilevel NDP). This is due to the fact that there are two decision makers involved (road users and authorities) which have different objectives (Chen et al., 2010). Due to the difference in objectives, a kind of game arises in which the authorities set their decision variables in such a way that their objectives are optimised (upper level optimisation), to which the road users respond by changing their route choice (lower level optimisation). To this change in route choice the authorities respond by adjusting their decision variables, and these reactions circle until convergence has been reached (Figure 1.2).

Road users tend to be opportunistic people that try to maximise their utility (or in case of travel minimise their disutility). In nearly all literature the objective of the lower level (in this case road users) is therefore to minimise

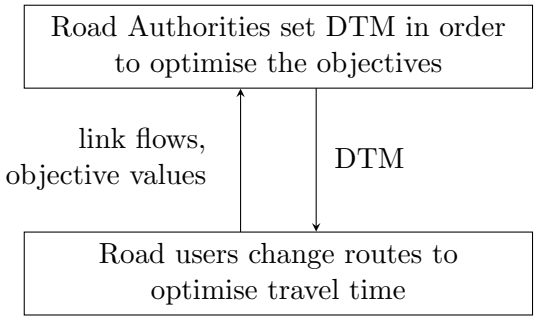


Figure 1.2: Bilevel Network Design Problem

travel time (TT) or (generalised) travel cost (GTC). This minimisation is attained when the so-called (Stochastic) User Equilibrium (SUE or UE, also known as user optimum) is reached, a point in which no road user can reduce his (or hers) objective by changing to another route. (see: Poorzahedy & Turnquist, 1982; Chiou, 2005; Gao et al., 2005; Poorzahedy & Rouhani, 2007; Zhang & Lu, 2007; Xu et al., 2009; Chen et al., 2010). This is in accordance with (and also known as) the first principle of Wardrop, which states *'the journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route'* (Wardrop, 1952).

For the upper level the objective is usually to minimise total travel time (Gao et al., 2005; Poorzahedy & Rouhani, 2007; Zhang & Lu, 2007) or travel cost (Poorzahedy & Turnquist, 1982) over the entire network, also known as the System Optimum (SO). At this SO the second principle of Wardrop *'at equilibrium the average journey time is minimum'* (Wardrop, 1952) applies. When no budget constraints are used in the bilevel NDP, the construction costs can be incorporated in the upper level objective function (Chiou, 2005; Xu et al., 2009). There are only a few papers which use multiple objective functions in the upper level, Chen et al. (2010) use travel time (SO) and construction costs as two separate objective functions, Cantarella and Vitetta (2006) use in-vehicle travel time, access and egress time as a result of parking and CO emissions as their upper level objective functions whilst Friesz et al. (1993) focus on minimising the transport costs, construction costs, vehicle miles travelled and house removal. Sharma, Ukkusuri and Mathew (2009) who provide an overview of multiobjective optimisation for transport NDP are only able to list six papers. This shows that there is very little experience with using externalities as objective functions in bilevel NDP.

Finally, a NDP is a NP-complete problem (Johnson, Lenstra & Rinnooy Kan, 1978), which means that it is not possible to solve it to optimality in polynomial time. In fact, in order to determine the exact Pareto optimal set, a full enumeration of all combinations of DTM measures is necessary. This however,

is not possible (at least in reality) because the number of possible solutions usually is very large.[3] Determining a single lower level optimisation (using DTA to determine the SUE) in a realistic network easily takes an hour, which means that a full enumeration would take forever.[4]

In the next section we will introduce Genetic Algorithms and explain why they can be used to reduce the computational effort of searching for the Pareto optimal set.

1.3 Genetic Algorithms

Because the bilevel NDP is a NP-complete problem, a more intelligent approach has to be used in order to find (or at least approximate) the Pareto optimal set (POS). For these kind of problems a lot of algorithms (also known as metaheuristics) have been developed. These metaheuristics, which are developed since the 90s of the previous century, have proven themselves to be flexible and are capable of finding good solutions, even when non-standard objectives and binary or integer variables are involved (D. F. Jones, Mirrazavi & Tamiz, 2002). Unfortunately most of these heuristics focus on single objective problems. If we limit ourselves to algorithms that can be modified to work with multiobjective problems, Genetic Algorithms (GAs, also known as Evolutionary Algorithms; EAs), Simulated Annealing (SA) and TABU Search (TS) are the most commonly used algorithms (see for instance the book by Pham and Karaboga (2000) for an overview of these algorithms).

There is very little literature available about which algorithm will perform best when being confronted with a multiobjective NDP. In fact even when only considering single objective problems, literature still is uncertain which algorithm performs better. Youssef, Sait and Adiche (2001) applied the three algorithms to a floor planning problem and concluded that TS was best (both in results and computational effort) but GA was a close second (though required a lot of computational effort). Arostegui, Kadipasaoglu and Khumawala (2006) applied the three algorithms to the facility location problem and concluded that TS was to be preferred, since it was a more simple approach and was less dependent on the selection of parameters. Strangely Kannan, Slochanal and Padhy (2005) concluded more or less the opposite when applying several algorithms to an investment planning problem, they found that TS is amongst the worst solutions. Drezner and Wesolowsky (2003) found that TS en GA were alternating the best solution, but decided that GA was in the end the better approach. Braun et al. (2001) compared eleven heuristics and concluded that GA was the best (although a relatively simple approach was a

[3] Consider a problem with two ATSs, each with ten possible settings and six time periods, the number of possible solutions is $\left(10^2\right)^6 = 10^{12}$ or one trillion solutions.

[4] In fact if each DTA took only 1 second, the full enumeration of the previous example would take about $31,710$ years.

good second) and Alabas, Altiparmak and Dengiz (2002) chose TS to be the best algorithm, but this was solely based on the fact that TS only needed to evaluate a small part of the solution space. Taking into account that in the multiobjective NDP searching a large part of the solution space could even be considered an asset (something that is also recognized by Lau, Ho, Cheng, Ning & Lee, 2007) it is difficult to determine which algorithm is better. Note that none of these papers focussed on multiobjective problems, something that was taken into account in Possel (2009). He applied GA and SA to a problem similar to the one under consideration now (a multiobjective NDP, with externalities as upper level objective functions) and concluded that GA is most likely the better algorithm. Based on these studies it seems that GA could be considered a practical algorithm, something that is also reflected in the use of this algorithm in studies towards NDP (see: Gen, Cheng & Oren, 2001; Chakroborty, 2003; Drezner & Wesolowsky, 2003; Gen, Kumar & Kim, 2005; Cantarella & Vitetta, 2006; Cantarella, Pavone & Vitetta, 2006; Poorzahedy & Rouhani, 2007; Zhang & Lu, 2007; Schmöcker, Ahuja & Bell, 2008; Mathew & Sharma, 2009; Sharma et al., 2009; Xu et al., 2009; Chen et al., 2010).

Genetic algorithms are the invention of John Holland (Holland, 1975) and are based on the biological process of 'natural selection'. The main idea is that each solution ('chromosome') can be described by a series of bits ('genes'), i.e. each solution is described by the state of each explanatory variable. These biological terms are used because the algorithm mimics the process of combining two strings of DNA into one or two others. The algorithm moves from one population of 'chromosomes' to another by crossover (combining two 'parents' into one or two 'children', the 'offspring'), mutation (randomly changing the 'genes' of the 'chromosome') and inversion (inverting the 'genes' of a 'chromosome'). By selecting only the best solutions found in the total set of 'parents' and 'offspring' the algorithm ensures that good solutions can be found, whilst preventing itself from finding only local optima. This algorithm has proven itself in the past decades, since it has been applied to numerous problems in the fields of optimisation, economics, immune systems, social systems etc. (Mitchell, 1996). Genetic algorithms, as discussed in the previous paragraph, are designed to find a single optimal solution. However, due to the nature of the algorithm, using a population of solutions, the algorithm can easily be modified in order to cope with multiobjective problems.[5] If one selects the population to be large enough, this population will (eventually) describe the Pareto optimal set. This is why at the end of the previous century (and at the beginning of the current one) a lot of research has been done in developing Multi Objective Genetic Algorithms (MOGAs), the best known examples are Non-dominated Sorting Genetic Algorithm (NSGA; Srinivas &

[5]Genetic Algorithms only require that one is able to determine whether a solution is better than another solution (rank the solutions). For multiobjective problems this can for instance be done using the non-dominated sorting algorithm presented by Deb et al. (2002).

Deb, 1994), Strength Pareto Evolutionary Algorithm (SPEA; Zitzler & Thiele, 1999), Pareto Envelope-based Selection Algorithm (PESA; Corne, Knowles & Oates, 2000) and Pareto Archived Evolution Strategy (PAES; J. D. Knowles & Corne, 2000a). In a fierce competition amongst followers of the different algorithms, each algorithm was proven to be better than others in certain test problems. Therefore additions and alterations were made to each of the algorithms, which resulted in M-PAES (J. D. Knowles & Corne, 2000b), PESA-II (Corne, Jerram, Knowles & Oates, 2001), SPEA2 (Zitzler, 2001), NSGA-II (Deb, Pratap, Agarwal & Meyarivan, 2002) and finally SPEA2+ (M. Kim, Hiroyasu, Miki & Watanabe, 2004).

It is difficult to determine which algorithm is better, since each one outperforms others in specific test problems. It is also not clear if any of these approaches should be preferred when considering traffic related problems. Most papers (see the list mentioned earlier) do use GAs, but do not use a specific predefined GA. In fact only three studies that use a specific predefined GA have been found. Sumalee, Shepherd and May (2009) use NSGA-II in their optimisation of road charges and both Possel (2009) and Sharma et al. (2009) studied a NDP with budget constraints. Although not using predefined GAs might have advantages (one can optimise the GA for a specific case) it fails to take advantage of research that has already been done in this field.

1.4 Research Scope

In the previous three sections we described how DTM measures could be used to optimise traffic flows, how such a process could be modelled and which algorithms can be used to find (or better: approximate) the Pareto optimal set. In this section we will be more specific and decide which specific solutions and approaches we will use throughout this research.

This research will focus solely on DTM measures that directly influence network properties. This means that a DTM measure either influences the speed (in fact speed limit) or the capacity of certain links in the network. This is done because these DTM measures can fairly easy be modelled in existing transportation models, whereas modelling DTM measures that influence behaviour (e.g. traffic jam information) require extensive behavioural models. This leaves only three specific DTM measures that will be considered, which are listed below.

Automated Traffic Control System (ATS) In reality an ATS would require a control that specifies which direction gets green light when and for how long, however when using macroscopic models defining the capacity in a certain direction (which can be determined using the fractional green time, road capacity and a factor that accounts for turning) gives sufficient information;

Variable Speed Sign (VSS) These signs alter the maximum speed on certain links;

Variable Message Sign (VMS) Although in reality these signs can be used for a multitude of things, in this research we will limit its possibilities to adding or removing additional lanes. These lanes can be either a rush-hour lane, which usually is the hard shoulder of a motorway, or a reversible lane. We will refer to this specific use of VMS as Variable Lane Sign (VLS).

It is important to note that these measures will be applied dynamically, i.e. they are allowed to change over time. This means that authorities can create different optimal settings, e.g. for night, morning rush hour, daytime and evening rush hour periods. Of course it is also possible to use different settings within a single rush hour period. This does however also affect the method that is used to determine the user equilibrium that is attained, something that will be addressed later on.

We also limit the number of objectives that we like to attain by using DTM. The main reason for limiting the number of objectives is that adding more objectives only increases computational effort, without significantly contributing to this research. Furthermore it is important that the selected objectives do not have a positive proportionality constant, otherwise minimising one objective would automatically minimise the other. It should be possible to determine the objective value using the information from the network model and the DTA, this means that objective values should be determined using nothing more then maximum speed, capacity, road type, speed and intensity. Therefore three objectives have been selected, each representing another part of the effects that are caused by traffic. In the equations used to describe the objective values, the notation from Table 1.1 is used.
The first objective is the minimisation of congestion, which is measured using the Total Travel Time (TTT; hours). This is probably one of the most used objectives because it tries to attain an optimal solution from a transportation system point of view (SO). Note that this is not the same as the stochastic user equilibrium (SUE) solution that is used in the lower level optimisation. The value of this objective function can easily be determined using:

$$z_1 = TTT = \sum_k \sum_t \sum_m \frac{f_k^m(t) l_k}{v_k^m(t)} \tag{1.1}$$

The second objective is to minimise pollution, which is measured using CO_2 emissions (g). This objective is used because it gives a good view of the environmental effects of traffic, especially when global warning is concerned. The objective value can be determined using the European ARTEMIS emission

l_k	length of link k
δ_{kd}	indicator for road type, 1 if road k is of type d, 0 otherwise
δ_{kw}	indicator for urbanisation level, 1 if road k is of level w, 0 otherwise
η_w	correction factor for urbanisation level w (dB(A))
α_m, β_m	noise parameters for mode m
V_m^{ref}	reference speed for mode m (km/h)
$f_k^m(t)$	vehicle inflow on link k during time period t (veh)
$v_k^m(t)$	average speed on link k during time period t for mode m (km/h)
$E_{md}^{CO_2}(\cdot)$	CO_2 emissions of mode m depending on average speed (grams / veh $\cdot$ km)
$L_m(\cdot)$	average sound power level for mode m, depending on average speed (dB(A))
$\bar{L}_w$	weighted average sound power level on the network with urbanisation level w (dB(A))

Table 1.1: Notation in Objective Functions

model database and the following equation:

$$z_2 = CO_2 = \sum_k \sum_t \sum_m \sum_d f_k^m(t) \delta_{kd} E_{md}^{CO_2}\left(v_k^m(t)\right) l_k \tag{1.2}$$

The third and last objective is to minimise noise, which is measured using the weighted average sound power level at the source (dB(A)). This can be determined using the Dutch standard method (Ministry of Housing, Spatial Planning and the Environment, 2002):

$$\begin{aligned} z_3 = noise &= 10 \cdot \log\left(\frac{\sum_k \sum_w \delta_{kw} l_k 10^{\frac{\bar{L}_w - \eta_w}{10}}}{\sum_k \sum_w \delta_{kw} l_k}\right) \\ &\text{with } \bar{L}_w = 10 \cdot \log\left(\frac{\sum_k \sum_w \delta_{kw} l_k \sum_m 10^{\frac{L_m(\cdot)}{10}}}{\sum_k \sum_w \delta_{kw} l_k}\right) \\ &\text{with } L_m(\cdot) = \alpha_m + \beta_m \log\left(\frac{v_k^m(t)}{v_m^{\text{ref}}}\right) + 10 \cdot \log\left(\frac{q_{km}(t)}{v_k^m(t)}\right) \end{aligned} \tag{1.3}$$

As mentioned earlier, the NDP will be formulated as a bilevel optimisation problem. However in contrast to most of the bilevel NDPs found in literature (see the overview in section 1.2 about NDPs) which use a Static Traffic

Assignment model (STA) to optimise the lower level problem, we will use a Dynamic Traffic Assignment model. The main reason for using a DTA instead of an STA is that we would like to be able to change the settings of the DTM measures during the day (in order to allow specific schedules for e.g. morning and evening rush hours or even changing schedules within a rush hour). Although this could be modelled using STA in quite small networks (the point at which one reaches a DTM measure is more or less the same as the departure time) this is virtually impossible in larger (real sized) networks. In that case only DTAs are able to give (fairly) good predictions of the changing traffic flows over the network.

In this research we therefore use the software program OmniTRANS (version 5.1) which uses the Macroscopic DTA called Streamline. This model is based on an adaptation of the fluid transmission model by Messmer and Papageorgiou (1990), which was based on the model by Payne (1971), and uses the single-regime speed-flow-density relationships of Van Aerde (1995).

Because the upper level of the bilevel NDP is a NP-complete problem, a metaheuristic is used to approximate the optimal solution. From literature (see the discussion in section 1.3 about Genetic Algorithms) it is clear that genetic algorithms prove to be a good approach when struggling with complex (hard) problems. Especially NSGA-II (Deb et al., 2002) appears to be robust and capable of creating good solutions when applied to traffic related bilevel NDPs (Possel, 2009; Sumalee et al., 2009). Therefore the NSGA-II algorithm is chosen as the algorithm that will be used in this study.

The approach that is used to solve the bilevel NDP combines the knowledge about bilevel programming, Dynamic Traffic Assignments, Genetic Algorithms and externalities. Let us start by explaining the NSGA-II algorithm, which is shown in Algorithm 1.1 in more detail.

The algorithm starts with N exactly evaluated parent solutions (P_0), which can be generated using Random Sampling (RS), Stratified Sampling (SS), Latin Hypercube Sampling (LHS) or any other sampling approach (McKay, Beckman & Conover, 1979). Based on the objective values the fitness of each solution can be calculated using the non-dominated sorting and crowding distance algorithm. The non-dominated sorting algorithm which is shown in Algorithm 1.2 determines for all solutions the rank of the Pareto front in which they are located. The crowding distance algorithm, shown in Algorithm 1.3, determines per front which solutions are farthest apart from all the other solutions in that front. Solutions that are farther apart are considered to be more valuable and thus get a higher fitness value. The N best performing solutions are stored as the parents for the next generation (P_{g+1}) and the remainder are stored in a database D that is used to prevent exactly evaluating the same solutions twice. If we have reached the maximum number of generations G we terminate the algorithm, otherwise we select $2N$ random parents (P'_{g+1})

Algorithm 1.1 NSGA-II

1. Initialisation
Start with N exactly evaluated parent solutions P_0
Define a set of offspring $Q_0 = \emptyset$
Define a database for previously evaluated solutions $D = \emptyset$
Furthermore define the maximum number of generations G and set $g = 0$

2. Fitness Assignment
Combine $R_g = P_g \cup Q_g$
Determine fitness value by dominance and crowding distance

3. Selection
Select N best solutions (based on fitness) from R_g and store as P_{g+1}
Store remaining solutions in database, $D = D \cup (R_g \setminus P_{g+1})$

4. Termination
If $g \geq G$ terminate the algorithm, otherwise continue with step 5

5. Mating selection
Perform binary tournament selection with replacement and repair on P_{g+1} to determine mating pool P'_{g+1}

6. Variation
Apply recombination and mutation to mating pool P'_{g+1} to create offspring Q_{g+1}

7. Function Evaluation
Determine objective values for all solutions in Q_{g+1}
Set $g = g + 1$ and continue with step 2

which will mate in order to create N new children (Q_{g+1}). We then exactly evaluate the solutions in the offspring set (Q_{g+1}), update g and continue with step 2, i.e. we determine the fitness of the solutions in the combined set R_g.

When we state that we 'exactly evaluate the solutions' we follow the procedure as shown in Figure 1.3. The NSGA-II algorithm provides OmniTRANS with the settings for the different DTM measures[6] for all time periods. For each solution OmniTRANS first determines the resulting links flows and link speeds, after which these results (i.e. link flows and link speeds) are converted to objective values using the objective functions (equations 1.1–1.3), which is actually done by Matlab® for computational convenience. The final objective

[6] We will elaborate on the settings of DTM measures in chapter 2.

Algorithm 1.2 Non-dominated Sorting Algorithm (Deb et al., 2002)

for all $p \in P$ **do**
  $S_p = \emptyset$
  $n_p = 0$
  for all $q \in P$ **do**
    if $p \prec q$ **then** — if p dominates q
      $S_p = S_p \cup \{q\}$ — add p to the set of solutions dominated by q
    else if $q \prec p$ **then**
      $n_p = n_p + 1$ — increment the domination counter of p
  if $n_p = 0$ **then** — p belongs to the first front
    $p_{rank} = 1$
    $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$
$i = 1$ — initialize the front counter
while $\mathcal{F}_i \neq \emptyset$ **do**
  $U = \emptyset$ — used to store the members of the next front
  for all $p \in \mathcal{F}_i$ **do**
    for all $q \in S_p$ **do**
      $n_q = n_q - 1$
      if $n_q = 0$ **then** — q belongs to the next front
        $q_{rank} = i + 1$
        $U = U \cup \{q\}$
    $i = i + 1$
    $\mathcal{F}_i = U$

Algorithm 1.3 Crowding Distance Algorithm (Deb et al., 2002)

$l = |\mathcal{F}|$ — l is the number of solutions in $\mathcal{F}$
for all $i = 1 \ldots l$ **do**
  $\mathcal{F}[i]_{distance} = 0$ — initialise distance
for all objectives o **do**
  sort $\mathcal{F}$ on objective o
  $\mathcal{F}[1]_{distance} = \mathcal{F}[l]_{distance} = \infty$ — ensure that boundary points are always selected
  for $i = 2 \ldots l - 1$ **do** — for all other points
    $\mathcal{F}[i]_{distance} = \mathcal{F}[i]_{distance} + \frac{\mathcal{F}[i+1].m - \mathcal{F}[i-1].m}{f_m^{max} - f_m^{min}}$

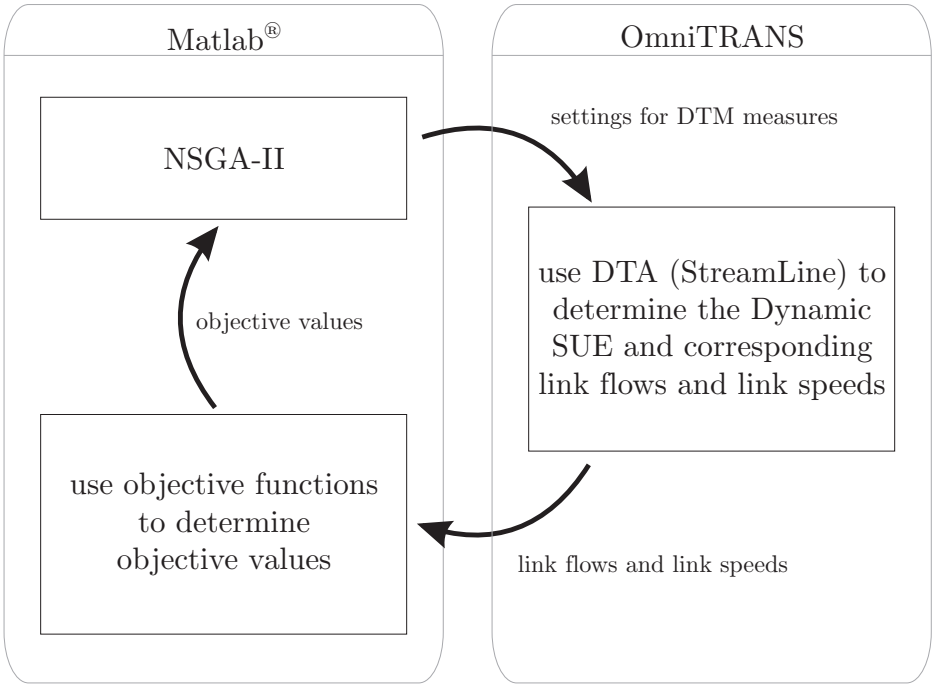


Figure 1.3: Procedure for Exactly Evaluating Solutions

values are returned to NSGA-II to be used for the fitness assignment.

The problem solving method described above gives indeed good solutions to the problem of minimising traffic externalities using DTM measures (Wismans, Van Berkum & Bliemer, 2009, 2010). And although a MOGA is already a much more intelligent approach than complete enumeration, it still needs to evaluate a lot of solutions (DTAs and objective evaluations) which makes that it easily takes months to solve a real-life problem. Secondly, there is very little protection from evaluating solutions that are prospectless, which means that computational time is spend in vain.

1.5 Research Goal

From this introduction two major drawbacks of the method that is now being used arise. First of all creating a single DTA takes quite a lot of time, which means that only a limited number of solutions can be evaluated. Secondly the method also generates solutions that do not stand a chance, which could be considered a loss of computational time, as that time also could have been spend on evaluating a favourable solution. In order to use the computational time more efficiently and improve the Pareto optimal set, it is desirable that these two drawbacks are tackled. This means that the computational time

should be used more efficiently by only evaluating favourable solutions.

A possible way to attain this situation is by developing a method that is able to quickly estimate the objective value of a possible solution (i.e. without evaluating the solution using a DTA) and use the result of this evaluation within the optimisation process (e.g. to determine whether this solution should be evaluated by a DTA). When focussing on this specific problem, we see that there are already a number of solutions available (these solutions are needed as 'parents' for the GA) which could be used in the problem solving method. For single objective optimisation problems a well-known method is the trust region optimisation (see e.g. Conn, Gould & Tointe, 2000). This method uses an approximation method (also known as metamodel or surrogate model) to evaluate solutions. There are many different approaches that can be used to approximate the objective values. Another well-known method (often used in social sciences) is regression analysis, which tries to fit a model to a set of known solutions. Besides the standard linear regression functions a variety of non-linear and multivariate functions exist, all serving a specific type of parameters and goal functions. These functions approximate the results of the DTA and could be used to determine whether a specific solution is interesting enough to be evaluated by the DTA.

It seems that a well chosen approximation model, in combination with the right explanatory parameters could be used to give a first estimate of the results of a specific set of DTM measures. To the best of our knowledge such an approach has not yet been developed or applied to a NDP, and the main goal of this research is therefore defined as:

> *accelerating the search for the Pareto optimal set found by multiobjective genetic algorithms for multiobjective network design problems, in which externalities are the objectives and DTM measures the decision variables, using function approximations.*

Besides predicting whether a solution is interesting enough to be evaluated by a DTA, approximations can be used in a number of ways. It would for instance be possible to use the approximations to determine in which area more data is required to improve the approximations, or the approximations can be used to determine the results in areas in which a lot of information is available. This means that several ways in which the Pareto optimal set can be improved will be investigated, however which ways are viable is dependent on the accuracy of the approximation function.

1.6 Research Model

Literature suggest that we start by creating a research model (Verschuren & Doorewaard, 2005). Main goal of a research model is to create an overview of the different subjects that should be studied, before a final conclusion can

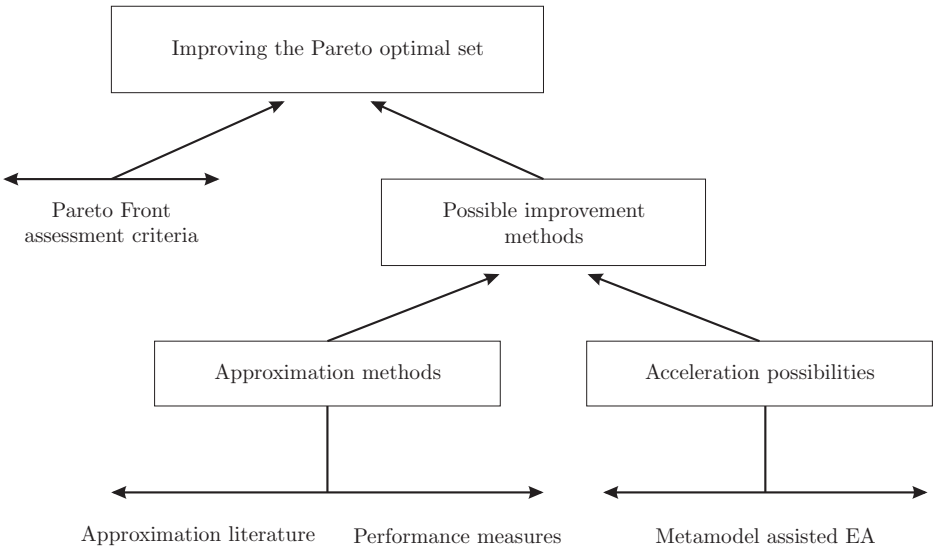


Figure 1.4: Research Model

be drawn. This framework can then be used to define the research questions and research methodology. We therefore developed a compact research model, which is shown in Figure 1.4.

As mentioned above, the main goal is to *improve the Pareto optimal set*. Improving implies that there is a clear view on what is a better solution (the same solution in less time, a better solution in the same time, a solution of which a higher percentage is indeed part of the Pareto optimal set, etc.) and therefore the subject of Pareto front assessment criteria should be investigated. These criteria can than be used to determine whether a solution (i.e. Pareto front) is better. Based on these criteria a set of *possible improvement methods* can be evaluated.

This set of possible improvement methods should be derived from the combination of *approximation methods* and *acceleration possibilities*. The approximation methods will be derived from literature and will be selected based on their performance on test cases. The acceleration possibilities will also be derived from literature, but we will select suitable ones based on how well they can be integrated in NSGA-II.

1.7 Research Questions

Based on the research goal and the research model described above four main research questions can be developed. The first research question focuses on the approximation methods and how well they approximate the objective functions. The second research question focuses on the genetic algorithm and

where there are opportunities to incorporate 'intelligence' into this genetic algorithm. The third research question focuses on how the results of the two previous questions can be combined into a single improvement method. And finally the fourth and last research question applies the developed methods from question three onto a realistic problem network. For each research question a set of sub questions can be defined that explore the subject in greater detail. This results in the following research questions:

1. How can the objective values of the bilevel NDP be approximated?
    a) What approximation methods are used for complex problems in literature?
    b) What data is needed in order to 'feed' these methods?
    c) How 'good' do these methods approximate the 'true' objective values?
        i. Which criteria can be used to determine 'good'?
        ii. How do these methods score on these criteria when applied to two different test networks?
2. In what way can the genetic algorithm be accelerated?
    a) How can the genetic algorithm be accelerated according to literature?
    b) Where in the genetic algorithm could 'intelligence' be incorporated?
    c) What intelligence can be incorporated in the genetic algorithm?
3. Which improvement methods do indeed improve the search for the Pareto optimal set?
    a) Which approximation methods can be incorporated in the genetic algorithm?
    b) Which criteria can be used to describe an improvement of a Pareto optimal set?
    c) How do the improvement methods score on these criteria?
4. How do these methods cope with realistic networks?

Finally a conclusion can be drawn about which improvement method is most suitable in which case, and to what extend this method is able to improve the Pareto optimal set.

1.8 Research Methodology

Main goal of defining a research methodology is to determine up front which approach is to be used to answer a specific research question. Suitable approaches could for instance be literature study, interviews, testing and simulation studies. In this thesis we decided to opt for two main approaches which are literature study and testing. The former approach is mainly used to determine what the current 'state of the art' is on a specific topic, whereas the latter is used to check whether a chosen model or approach is suitable for our problem.

In the remainder of this section we will discuss the methodology used, by answering the question *'what do we do, and why do we do it?'*

In order to answer research question 1 we decided to perform a literature study into approximation methods, where we focus on approximation methods that are commonly used in combination with optimisation problems. Furthermore we prefer approximation methods that are known to be combined with Genetic Algorithms. So why do we perform a literature study?

Probably the most important reason is that we would like to get a good idea how the different approximation techniques work. It would for instance be interesting to see which equations and/or algorithms are required to obtain the approximated objective values for, after all, applying an approximation method that is so complicated that it requires more time than exactly evaluating the objective values is of little use. Furthermore we need to know which information is needed as input data for the approximation techniques, since we have to be able to provide this information. Perhaps just as interesting are the different variants that might have been applied in the past, because they have often been optimised for a specific problem. Similarly it is interesting to see which (predefined) parameters are part of the approximation techniques, since they could be used to 'fine tune' the approximation technique.

It seems that after this literature study we should have a good and complete overview of the possibilities of the different approximation techniques. In order to reduce the complexity of the research, it seems requisite to reduce the number of variants (i.e. different variants or different parameter settings) of approximation techniques. It seems appropriate to combine this reduction with research question 1c which makes that we a) have to perform a literature study into quality measures for approximation techniques; and b) test which of the variants of the approximation techniques appear to be the 'best' according to these quality measures.

The literature study which we perform to answer research question 1(c)i focusses primarily on well-known statistical techniques, which are commonly used in determining whether a model is a reliable estimate of the real situation. However in our situation it is probably just as interesting whether a variant of an approximation technique is able to predict whether a solution is going

to be part of the Pareto front, independent of the estimated objective value. We therefore also try to find quality measures that measure this phenomenon.

Finally we will use the 'testing' approach to determine how good the variants are able to approximate the objective values (research question 1(c)ii). However at the same time we will use this approach to determine which of the variants are 'best' (and will therefore be used in the remainder of this research). Of course there is the risk that a variant, which appears to perform extremely well in one case, performs extremely poor in another. We therefore decided to use two different road networks for the test, only variants that perform 'good' on both networks are considered as 'best' variants of approximation techniques. Using two different networks therefore increases the chances that the chosen variants also perform 'good' on other networks.

For answering the second research question we follow a slightly different approach, although the start is similar. We start by performing a literature study into the field of the Metamodel Approximated Evolutionary Algorithms (MAEAs). The goal of this literature study is to provide an overview of different MAEA approaches that have been used in the past, and we thereby do not limit ourselves to our field of interest (i.e. we do not limit ourselves to traffic and transport related research).

It is likely that we have to reduce the number of MAEA approaches that we are going to consider in the remainder of the research. Theoretically we could use the testing approach again, however, it is likely that this would take to much time since it would require us to combine different variants of approximation techniques with all of the MAEA approaches. Furthermore it is likely that not all MAEA approaches can be combined with the NSGA-II algorithm or the approximation techniques that we decided to use. We therefore limit the number of MAEA approaches using more qualitative criteria such as: a) is the approach intuitive; b) can the MAEA approach be used in combination with the chosen approximation techniques; and c) what is the computational effort of the MAEA approach. Especially the latter criterion can be considered relevant, since approaches that require new optimisation and approximation models would probably require to much effort (i.e. evaluating solutions exactly might be less expensive).

At this point we have arrived at research question 3 which we will answer mainly by testing. However we start by combining the results from the previous two research questions into Approximation Method Assisted NSGA-II algorithms (AMANs). We will also check whether the proposed AMANs are indeed viable, i.e. we check whether the data that is provided by the approximation methods is sufficient for the MAEA approaches.

What is furthermore important is that we perform a literature study in order to answer research question 3b. Main goal is to determine how previous research has defined an improvement in the Pareto optimal set, which can then

be used to compare the different AMANs with the original NSGA-II algorithm. Since it is likely that there are many different performance measures present in literature, we decide to choose the criteria in such a way that they measure the quality of the Pareto optimal set on different aspects.

We will test the results of the different AMANs and the original NSGA-II algorithm on two different road networks, for reasons we already explained. Main goal of the testing is to determine whether the results of the AMANs are comparable to the results of the original (much more computational expensive) NSGA-II algorithm. At the end we will choose one 'best' AMAN based on critical discussion of the scores on the different performance measures.

The AMAN that is chosen as 'best' approach cannot be regarded the single best approach, but it is the approach we will use to answer research question 4. Again we will use the testing approach to answer this research question, since we want to compare the solutions that are found by the original NSGA-II algorithm and the selected AMAN algorithm. In order to compare the solutions found by the two algorithms, we do again use the performance measures that are defined in research question 3b.

Having answered all our research questions we are able to draw conclusions, and determine whether we have achieved our research goal. Based on the results from research questions 3 and 4 we are able to determine whether the AMANs are indeed capable of producing Pareto optimal sets that are comparable to the sets that are found by the original NSGA-II algorithm. Furthermore we can determine what the possible reduction in computational effort (calculation time) is. It is this reduction in calculation time which, in combination with a comparable Pareto optimal set, can be considered the improvement of the search.

1.9 Outline

In the previous section we have explained in detail which literature studies and tests we are going to perform to answer our research questions and obtain our research goal. In this section we will provide an outline of the remainder of this thesis and explain which research questions are answered where in this document. Figure 1.5 provides an overview of this outline.

Chapter 1 (Introduction), which you probably just read, starts with an introduction to the problem and a brief literature overview of Network Design Problems and Genetic Algorithms. Furthermore it contains the research scope, research goal and research model which result in the research questions and the research methodology. Summarised we can state that chapter 1 provides an introduction to our research and discusses the research design.

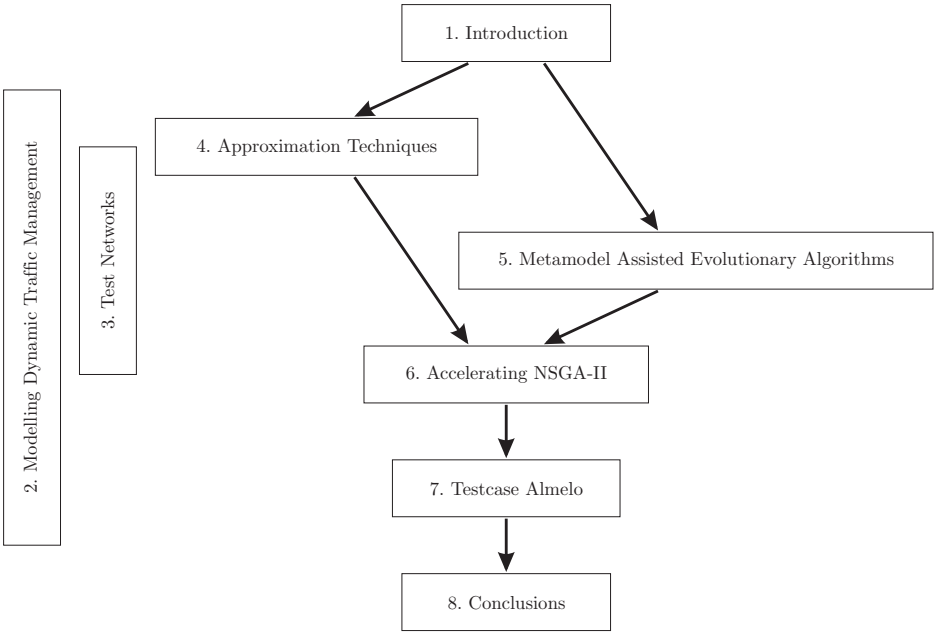


Figure 1.5: Outline of the Thesis

In chapter 2 (Modelling Dynamic Traffic Management) we discuss the problem of how we should model the different Dynamic Traffic Management measures in our Genetic Algorithm and approximation methods. As such, it is a somewhat independent chapter that provides a more theoretical background for our research. Besides presenting how we model our DTM measures, we explain what the consequences of this specific model are and provide a brief discussion on how the problem size can be reduced. Furthermore we briefly explain how more complicated DTM measures than the ones used in this thesis can be modelled using our framework.

The test networks that are used in chapters 4 and 6 of this thesis, are introduced in chapter 3 (Test Networks). Not only do we discuss the characteristics of these networks, we also explain why these specific networks are used.

Chapter 4 (Approximation Techniques) start by introducing three different approximation techniques. We do discuss the mathematical background of the approaches, provide an overview of how different parameters can be estimated and discuss different variants that have been used in literature. We continue by introducing different quality measures that describe how 'good' an approximation method approximates the objective values. We also provide a quality measure that can be used specifically for multiobjective problems, and indicates how 'good' an approximation method is in determining whether the solutions are part of the Pareto optimal set (that has been found so far). After presenting the results for the different quality measures for the different

variants of the approximation techniques, we select two approaches that we consider to be the 'best' amongst the evaluated variants.

We continue in chapter 5 (Metamodel Assisted Evolutionary Algorithms; MAEAs) by discussing different MAEAs which we found in literature. Using the criteria we discussed before we select two different MAEA approaches that can be combined with the selected approximation techniques and can be incorporated into the NSGA-II algorithm.

The results from chapter 4 and 5 are combined in chapter 6 (Accelerating NSGA-II) in which we develop three new Approximation Method Assisted NSGA-II (AMAN) algorithms. In order to be able to determine which of these AMANs performs best, we develop a set of performance measures that can be used to compare different Pareto fronts. We thereby distinguish three categories of performance measures, dominance metrics, diversity metrics and other performance measures. Furthermore we present the results for the three AMANs and the original NSGA-II algorithm on both test networks. We conclude the chapter by selecting one AMAN that will be used in the next chapter.

We apply the selected AMAN and the original NSGA-II algorithm on the slightly altered road network of Almelo in chapter 7 (Testcase Almelo). We start by presenting the Almelo network, after which we discuss the results for both the AMAN and the NSGA-II algorithm, using the performance measures that were defined in the previous chapter. We also provide an overview of how the Pareto fronts can be interpreted from a traffic and transport point of view. This analysis is not performed in depth, as it is not part of our research. We conclude by determining whether the Pareto front found by the AMAN is comparable to the Pareto front found by the original NSGA-II algorithm.

We end this thesis with chapter 8 (Conclusions), which consists of the combined conclusions of all the previous chapters, but also contains a final conclusion that determines whether we attained our research goal. We provide an overview of the subjects that require further research, before we can draw firm conclusions about AMANs. Furthermore we provide the reader with a couple of interesting subjects that could be used to improve or extend our knowledge of AMANs and their application to Dynamic Traffic Management.

Chapter 2

Modelling Dynamic Traffic Management

That which is static and repetitive is boring.
That which is dynamic and random is confusing.
In between lies art.

John A. Locke (1899 – 1961)

In this chapter we describe how different Dynamic Traffic Management measures affect the properties of the road network. We therefore start by introducing a typology of scales (section 2.1) after which we introduce a problem framework and decide how DTM measures should be modelled. We further specify these decisions, based on how the road network is affected by the DTM measure, for each of the three measures in section 2.3.

In section 2.4 we provide a final overview of how we decided to model the different DTM measures in both the Genetic Algorithm and the approximation methods. Furthermore we provide a discussion (section 2.5) on how researchers and principals can improve the results by redefining the DTM measures.

2.1 Typology of Scales

Genetic Algorithms as well as approximation methods need input variables that describe the current setting of each DTM measure. However, as we will show in the next paragraphs, the ways in which these settings can be described differ significantly. Even more important, the input variables for the Genetic Algorithm are not necessarily the same as the input variables for the approximation methods. Therefore we will first introduce a typology of *scales of measurement*, after which we will explain for each of the selected DTM measures how a setting influences the state of the road network. Finally

we will select the input variables for both the Genetic Algorithm and the approximation methods.

The typology of scales that will be used is the one introduced by Stevens (1946). He defines four types of scale:

nominal where one uses labels, but cannot say anything about whether a 'value' is better or more than another 'value';

ordinal where one uses ranks, thus one can state that one is better or more, but it is impossible to say something about the difference between the two;

interval where one uses values, however only the difference between two values have meaning, the values itself are 'meaningless';

ratio where one uses values relative to an absolute zero, therefore not only the differences but also the values itself have meaning.

Note that the former two are qualitative descriptors, whilst the latter two are quantitative descriptors (and thus are always numeric values). The difference between the latter two is somewhat difficult to grasp, but the easiest way is to compare two important scales of temperature. The first is Celsius, which has chosen an arbitrary zero value and thus is an interval scale, whereas the second is Kelvin which has chosen the absolute zero (the temperature at which particles have zero kinetic energy) and thus is a ratio scale. This difference becomes even more clear if we apply the kinetic theory of gasses, doubling the temperature in Celsius does not double the average energy of atoms, whereas doubling the temperature in Kelvin does.

2.2 Problem Framework

In order to determine which type of scale we have to use when modelling the DTM measures in Genetic Algorithms and approximation methods we have to create a more formal framework. The main reason for creating this framework is that it aids us in explaining clearly why certain typologies can and why others cannot be used in a specific situation. In order to do so, we define three sets of data.

The first set consists of the different DTM measures that are applied to a road network. By definition it is not possible to have an 'inactive' DTM measure, something that is perhaps best explained using an example. Consider a situation in which we have a motorway where the traffic managers have the possibility to cross off a specific lane using a Variable Lane Sign (VLS). However if in the current situation the managers decided to allow traffic to use this specific lane, we say that the DTM measure is 'active' by allowing this lane to be used. If they would decide to cross off the lane, the DTM measure

would still be 'active' although this time by denying the use of this lane. If we say that set D consists of all DTM measures on the network and we use d, $d \in D$, to denote a specific DTM measure. Now $\forall\ d \in D$ we can define a set of possible settings, which we will call S_d, which is considered to be unique for DTM measure d.

Secondly we define a set of controls C, we let c indicate a specific control and obviously it holds that $c \in C$. Each of these controls is part of exactly one DTM measure, so we can define $C_d \subseteq C$ as the subset of controls influenced by DTM measure d. Furthermore because of the aforementioned uniqueness (each control is linked to exactly one DTM measure) we can state that $C_i \cap C_j = \emptyset$ $\forall\ i, j \in D,\ i \neq j$. Last we can define $|C_d|$ as the cardinality of the set C_d.

Finally we define set K as a set containing all the links in the network, where k is used to denote a specific link, $k \in K$. In theory it is possible that an individual link k is influenced by multiple DTM measures, and thus by multiple controls. A good example would be a motorway section where it is possible to add an additional link and simultaneously change the allowable speed, both DTM measures being controlled separately. We can now define $K_c \subseteq K$ as the set of links controlled by control c, again $|K_c|$ is used to denote the cardinality.

It is perhaps necessary to explain how the framework introduced above works. When changing a DTM measure d to a specific setting $s_d \in S_d$ we also change all underlying controls $c \in C_d$ to this specific setting s_d. The controls store the information on how the link properties should be changed when a specific setting is applied, in fact it stores the exact value to which the link properties should be set. Therefore all links $k \in K_c$ have the same property value for the properties that are changed using control c, which is changed by DTM measure d ($c \in C_d$). Because each control can only change the property of all underlying links to a single property value, we might require multiple controls to describe a single DTM measure.[1] We will apply this methodology to the ATS, VSS and VLS DTM measures in the next sections.

In order to be able to incorporate our DTM measures into a Genetic Algorithm we only require crude information, i.e. we do not require specific knowledge of the network. In this case simply knowing which setting is applied for each of the DTM measures is sufficient . Historically Genetic Algorithms were designed requiring only binary data to represent the solution applied (see e.g. Mitchell, 1996; Pham & Karaboga, 2000). However we would not like to limit ourselves to DTM measures which can have only 2^n solutions,[2] especially since it eliminates the possibility to use the quite common three-point (low-medium-high) and five-point settings. It is therefore that we prefer to use an integer

[1] The ATS which we will discuss shortly, would be a good example of a case in which we need multiple controls to describe a single DTM.

[2] Using binary data one would have to use DTM measures with 2, 4, 8, 16, $\ldots 2^n$ settings.

value to describe the setting that is applied for a specific DTM measure, i.e. $s_d \in \mathbb{N}^*$. This means we only require a string of $t \cdot |D|$ integers, where t denotes the number of time periods for which we can select different settings and $|D|$ is the cardinality of the set of DTM measures.

Whereas we would like to use the crudest data available for the Genetic Algorithms, we would like to get the most detailed information for the approximation methods. In theory this would mean that we use all property information (e.g. maximum free-flow speed, outflow capacity, number of lanes, etc.) from all the links in the network, however this would most likely result in solutions in which all known points are fit exactly (heavily overdimensioned problem) but none of the individual variables can be considered significant. Furthermore it is likely that the computational effort of fitting such an approximation model would be so large, that the reduction in calculation time would be marginal at best. We therefore tried to reduce the problem size using the following criteria:

1. we use the most detailed property information available;

2. we use the property information at the highest level possible; and

3. we only use property information that is changed by the DTM measures.

Based on these criteria we decided to use the property information at control level as variables for our approximation methods, since:

1. at the control level we have property information for all links $k \in K_c$ and therefore using the links k themselves does not contribute more knowledge about the properties;

2. at the control level we know the property information, if we would go any higher we would only know the setting s_d; and

3. by using the data at the control level we do not use property information that is unchanged, because that information is only available at link level.

Because we now use the control level instead of the DTM measure level, we require much more variables to describe a single solution. In fact, we require $t \cdot \sum_{d=1}^{|D|} |C_d|$ values to describe a single solution (note that $|C_d| \geq 1$ by definition).

In the following section we will explain how this framework affects the typology of scale that can be used for the variables in the Genetic Algorithms and the approximation methods.

2.3 Modelling DTM measures

Because there are differences between each of the DTM measures (ATS, VSS, VLS), we will discuss each of the proposed DTM measures seperately in the following subsections. For each of the DTM measures we will discuss how they affect the road network and we will subsequently decide how we should model the DTM measures in the Genetic Algorithm and the approximation methods.

Automated Traffic Control Signals

Starting with Automated Traffic Control Signals (ATSs), we see that each ATS usually consists of all the signals that are needed to control a single junction. There are many different ways in which traffic lights can be modelled, from very simple methods that can be used in macroscopic assignment to very complex methods that try to mimic individual vehicle behaviour in microscopic assignments. In our situation it is not necessary to have a very detailed model of a junction, since we use a macroscopic model and focus on strategic DTM. We therefore decided to model an ATS using a set of outflow restricting devices.

Going back to the framework introduced in the previous subsection we see that in this case the property that is being changed is the outflow capacity (for link k denoted by f_k^{max}). In order to be able to determine which type of scale is to be used for the approximation method variable, we need to determine on what scale the outflow capacity can be measured. Since outflow capacity is obviously a quantitative value and an outflow of zero has a clear scientific meaning (there is no traffic whatsoever on this link) we can state that the outflow capacity can be measured on a ratio scale. We therefore can also state that we can use the ratio scale for the approximation method variables for the ATS DTM measures.

Unfortunately typifying the scale of variables for the Genetic Algorithm is slightly more complicated. We will explain this using the framework from the previous subsection and a couple of (relatively simple) examples.

First, if an ATS would control only a single link (road section), it is clear that this ATS only requires a single control and this control only influences a single link, thus the cardinality of each subset is one ($|C_d| = 1,\ |K_c| = 1$). We could now easily name each setting (and thus the variable for the Genetic Algorithm) with the outflow capacity of this link and the variable could be measured on a ratio scale (see Table 2.1a).

Second, it is possible that an ATS controls two different links (road sections). We can model this problem in two ways, although one approach is clearly unrealistic. The first approach is to use a single control ($|C_d| = 1$) which influences both links ($|K_c| = 2,\ k = 1, 2$), this would however result in

setting	link 1
100	100
200	200
400	400

(a) ATS with single link

setting	link 1	link 2
1	100	400
2	200	300
4	400	100

(b) ATS with two links

setting	link 1	link 2	link 3
a	100	400	200
b	100	200	400
c	200	300	200
d	200	200	300
e	400	100	200

(c) ATS with more links

Table 2.1: Examples of Different ATS Setting Scales

a intersection with a variable capacity, since $f_1^{max} = f_2^{max}$ and thus the total capacity of the intersection is equal to $2 \cdot f_1^{max}$. Obviously an intersection in which the total capacity increases depending on the setting that is applied is unrealistic, and we therefore use the second approach which is to use a dual control system ($|C_d| = 2,\ c = 1, 2$) where each control influences a single link ($|K_c| = 1\ \forall\ c = 1, 2$). In order to maintain a fixed intersection capacity (denoted by Q) it holds that $|K_1| \cdot f_1^{max} + |K_2| \cdot f_2^{max} = Q$. Table 2.1b gives an example of such an ATS, and it is clear that we can no longer use the ratio scale, since the value of zero has no scientific meaning anymore. We can however use the interval scale by applying the setting numbers as shown in Table 2.1b. Note that the difference between setting 2 and 4 is indeed twice the difference between setting 1 and 2 (and that this holds for all links), which is a requirement for the interval scale.

Third, we could now devise an ATS which controls three different links (road sections) and, following the explanation from the previous paragraph, we would have to model this problem using three controls ($|C_d| = 3,\ c = 1, 2, 3$) each controlling a single link $|K_c| = 1\ \forall\ c = 1, 2, 3$). Furthermore it should hold that $|K_1| \cdot f_1^{max} + |K_2| \cdot f_2^{max} + |K_3| \cdot f_3^{max} = Q$, in order to maintain the intersection capacity. However in this case we are unable to apply the interval scale to the problem (see Table 2.1c) because the difference between setting 2 and 4 is no longer twice the difference between setting 1 and 2 for all the links. In fact, it is impossible to rank the settings, since no setting is 'higher' or 'lower' than any other setting. This leaves us no other solution but using the nominal scale for the variables in the Genetic Algorithm.

Obviously the same problem would remain to exist when a fourth, fifth,

etc. link is added to the ATS. However in order to keep a consistent structure (i.e. not dependent on the number of links that is being controlled) we decided to use the nominal scale for the settings of the ATS measures and the variables in the Genetic Algorithms.

Variable Speed Signs

Variable Speed Signs (VSSs) usually reduce the maximum speed (free-flow speed) for vehicles on one or more (usually consecutive) road sections. However in contrast to the ATS from the previous subsection a VSS often changes more than one link property. To illustrate this we can refer to the most common usage of VSSs, which is on a motorway. When for some reason road managers decide to reduce the speed limit from 120 km/h to 100 km/h, this only affects cars, lorries etc. still have to obey the 80 km/h speed limit specified for their mode. VSSs therefore do not (necessarily) affect all modes of traffic equal, which is why we assume that VSSs change two link properties, the speed limit for cars ($v_k^{max;\text{car}}$) and the speed limit for trucks and lorries ($v_k^{max;\text{lorry}}$).

However it seems unnecessary to use both speed limits for our approximation methods, mainly because freight traffic accounts usually for only 10% of the total traffic flow. The speed limit for car traffic therefore seems much more important as a traffic predictor than the speed limit for trucks and lorries. Furthermore it is likely that the dynamic speed limits for car traffic usually are 120, 100 and 80 km/h, which means that the speed limit for freight traffic becomes irrelevant (the speed limit is never changed, since the minimum speed limit for car traffic is equal to the maximum speed limit for freight traffic). We therefore decided to use the maximum speed for cars $v_k^{max;\text{car}}$ as variable for the approximation methods. Because a speed limit is a quantitative value and a speed limit of zero has a scientific meaning (there is no traffic on this link, since travel time would equal infinity), we are allowed to use the ratio scale for this type of variable.

If we go back to the framework described earlier, we see that all the links which are affected by the DTM measure can be directed by a single control, since for all related links the same new speed limits are imposed. We can therefore state that the cardinality $|C_d| = 1$, which has some benefits. In this case however we do not change a single but two properties of each link (speed limits of both cars and freight traffic) and although these properties are quantitative, they behave differently (i.e. they have different intervals). We can therefore not use the ratio or interval scale, however since the direction of both properties is the same, we can use the ordinal scale to describe the variables in the Genetic Algorithm.

Variable Lane Signs

Finally Variable Lane Signs add additional lanes to the road network, thereby expanding the capacity of a specific road section (or set of road sections). However an increasing number of lanes is usually accompanied by a decreasing capacity per lane, thus we effectively change two link properties, i.e. the number of lanes s_k and the capacity per lane q_k^{lane}.

However we can easily merge these two properties into a single indicator that truly influences traffic behaviour, which is capacity per link, $q_k = s_k \cdot q_k^{lane}$. This new capacity per link can be measured using a ratio scale, since it is a quantitative variable where zero has a scientific meaning (when the capacity per lane or the number of lanes is zero, there is no traffic). We therefore decided to use the capacity per link q_k as the variable used in the approximation methods.

Similar to the VSS described in the previous subsection, we only need a single control to direct all links affected by the VLS ($|C_d| = 1$). Although it appears that we can simply follow the reasoning from the previous subsection, there is one major difference, which is that in this case the two properties behave exactly opposite. If we however look at the effect on the road network (i.e. the capacity of a link q_k) we see that there clearly is a single direction and we therefore do use the ordinal scale to describe the variables in the Genetic Algorithm. Note that the interval scale cannot be applied because doubling the number of lanes does not double the link outflow capacity.

2.4 Conclusion

For the approximation methods it seems fit to select those input variables that have the strongest relation to what happens with the road network. We therefore decided to select the characteristic attributes as input variables. Note that all these variables are measured on a ratio scale, which means that all mathematical operators can be applied.

It is a bit more difficult to select good variables that can be used in the Genetic Algorithm. For the ATS we decided to use the nominal scale, whereas for the VSS and VLS we are able to use the ordinal scale. It is important to note that both scales do not require us to use specific values to describe the variables, as would for instance be the case when interval scales were used (in that case we would have to make sure that the property values of variable value 2 are indeed twice the property values of variable value 1).

The reason why we have to determine the scales that are going to be used up front, is that the nominal scale does not allow intelligent mutations. Intelligent mutation is defined as mutations which assumes knowledge of the problem. The most widely used intelligent mutation is ± 1, in which a setting is mutated to a setting that is one higher or one lower than the current one. However in order to make this type of mutation legit, it is required that a

setting higher of lower does have a meaning, which is the case in ordinal, interval or ratio scales. For any of those scales, ± 1 means that one decides to use a setting that is close to the current one. When however a setting is measured on a nominal scale ± 1 implies selecting a random solution, since the solution stored one above or below the current one does not necessarily have any relationship with the current one. This 'random selection' however is strongly biased, because we only consider solutions that we accidentally stored close together. It is therefore that, when a setting is measured on a nominal scale, only random mutation should be allowed.

Finally Table 2.2 provides an overview of the variables that are used in the approximation methods as well as the scale that is used for the variables in the Genetic Algorithms.

DTM measure	Approximation Method Variable	Scale in GA
ATS	$f_k^{max} \ \forall \ k \in K_c, \ \lvert C_d \rvert \geq 1$	nominal
VSS	$v_k^{max;\text{car}} \ \forall \ k \in K_c, \ \lvert C_d \rvert = 1$	ordinal
VLS	$s_k \cdot q_k^{lane} \ \forall \ k \in K_c, \ \lvert C_d \rvert = 1$	ordinal

Table 2.2: Overview of Variables for each of the DTM Measures

2.5 Discussion

There are some additional interesting issues we would like to address in this chapter. The first issue is the influence of the number of time periods in which the DTM measures are allowed to take different settings. Earlier we showed that the number of variables in the Genetic Algorithm is equal to $t \cdot |D|$ and the number of variables in the approximation methods is $t \cdot \sum_{d=1}^{|D|} |C_d|$. Clearly the number of variables is linearly dependent on the number of different time periods that are considered. If we however look at the size of the solution space (i.e. the total number of solutions that is theoretically possible, denoted by Θ) we find that:

$$\Theta = \left\{ \prod_{d=1}^{|D|} |S_d| \right\}^t \tag{2.1}$$

Where $|D|$ denotes the cardinality of the set of DTM measures (i.e. the total number of DTM measures), $|S_d|$ denotes the cardinality of the set of settings of the d^{th} DTM measure and t is the number of time periods.[3] Since the

[3]If we, for example, want to determine the 'optimal' DTM measure schedule for the morning peek hour (6.00am - 9.00am) we could subdivide this time period into six *time periods* (6.00am - 6.30am, 6.30am - 7.00am, etc.). Main reason for using these six time periods is that it is likely that we want different DTM settings in the first (6.00am - 6.30am) and last (8.30am - 9.00am) time period.

number of time periods is in the power, it is clear that the effect of t on the size of the solution space Θ is enormous. Furthermore it is highly likely that the larger the solution space, the more difficult it is to come close to the true Pareto front. We therefore recommend that principals and researchers rethink the number of time periods they find necessary, because the effects on the outcome might be larger than expected. A possible solution is to first apply the approach with two or three time periods. Based on these results one can decide whether increasing the number of time periods might improve the results and for which specific DTM measures such an increase would be beneficiary. If a DTM measure is not changed within the two or three time periods test run, it is unlikely that it will change in the 'real' six time period run.

A second issue that might be interesting for principals and researchers alike is the ability to incorporate 'advanced' DTM measures. We can best explain this feature using an example: consider a situation in which we have two ATS systems relatively close together with one major traffic flow using both systems. If we would schedule both ATS systems independently solutions in which this traffic flow is blocked at one ATS and given more or less free-flow on the other are not unlikely. However it makes more sense that the two traffic lights work together in the sense that the capacity that is reserved for this major flow is equal on both intersections. We could model such a situation by 'merging' the two ATS systems into a single DTM measure. Consider the situation in which ATS 1 has four controls and ATS 2 has 5 controls (for instance due to a one way bus lane). If we schedule both ATS systems individually the size of the solution space Θ is $(4 \cdot 5)^t = 20^t$. However if we would merge the two ATS systems into a single one, we find that we only have $(4 + 5)^t = 9^t$ possible solutions. Actually we could reduce the size of the solution space to 8^t, since the two links on which the major traffic flow is present can be directed by a single control.

There is however a catch when using these 'advanced' DTM measures that require preknowledge. Because we severely limit the size of the solution space it is very well possible that we 'remove' solutions that are actually really good. One of the main advantages of letting a Genetic Algorithm determine possible solutions is that the GA will also find unconventional solutions, which might have been rejected by consultants or principals. The question is whether they would still reject those solutions if they appeared to outperform many of the other solutions.

In conclusion we can say that these approaches allow principals and researchers to create a network with DTM measures as they like. It should however be noted that some decisions can have a major effect on the computational effort or on the quality of the solution. Care should therefore be taken when one wants to reduce the size of the solution space by merging different DTM

measures or increasing the solution space by increasing the number of time periods.

Chapter 3

Test Networks

Testing leads to failure,
and failure leads to understanding.

Elbert Leander "Burt" Rutan (1943 –)

In order to understand how the different approximation models (chapter 4) and Metamodel Assisted NSGA-II algorithms (AMANs; chapter 6) work, we have to test these approaches. In this chapter we will therefore introduce the two test networks that will be used to test the approximation methods and AMANs. We decided to use two different networks, because testing an approach on different situations will lead to better understanding of how well this approach works in general.

In section 3.1 we therefore introduce a road network that has been used in previous research on the topic of finding optimal Dynamic Traffic Management. This enables future researchers to compare the results of both studies. Furthermore we developed a new test network (section 3.2) which incorporates more complicated issues found in real-life road networks.

3.1 Test Network I

The first test network was designed for the research by Wismans et al. (2009, 2010). They wanted a small test network in which there were only a limited number of easily identifyable routes between one origin and one destination. Furthermore they wanted to incorporate all issues that can be found in a realistic network such as different levels of urbanisation, different levels of 'Sustainable Safety'[1] and roads with different emission properties. Similarly they wanted a set of DTM measures which, when changed, would cause the behaviour of road users to change significantly. Although this resulted in

[1] Dutch: Duurzaam Veilig Verkeer

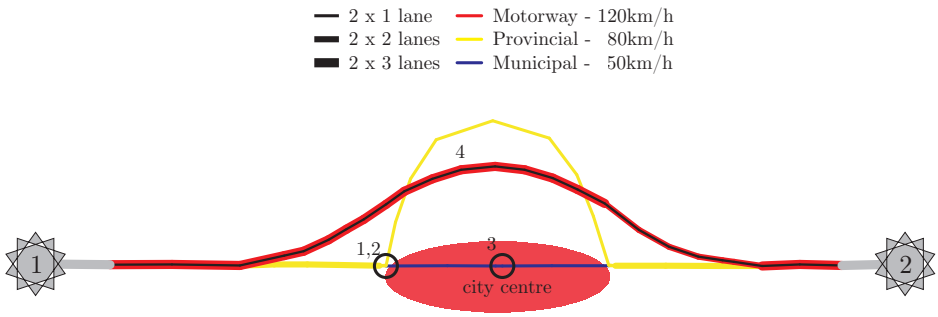


Figure 3.1: Layout of Test Network I

setting	outflow capacity
a	500
b	600
c	700
d	800
e	900
f	1000
g	1100
h	1200
i	1300
j	1400
k	1500

(a) ATS - Regional Road → Regional Road

setting	outflow capacity
a	500
b	600
c	700
d	800
e	900
f	1000
g	1100
h	1200
i	1300
j	1400
k	1500

(b) ATS - Regional Road → Local Road

setting	outflow capacity
a	500
b	600
c	700
d	800
e	900
f	1000
g	1100
h	1200
i	1300
j	1400
k	1500

(c) ATS city centre

setting	free-flow speed
1	120
2	100
3	80

(d) VSS motorway

Table 3.1: Settings of DTM Measures for Test Network I

a somewhat theoretical construct, it does contain all elements that can be found in a realistic network, whilst the results of a solution can be determined relatively fast.

The road network comprises of two centroids with a one directional flow of traffic (6000 pae/hr) and three different routes: a motorway (red), a regional road (yellow) and a local road (blue) through a town (see Figure 3.1).

The motorway consists of 2×3 lanes with a 120 km/h speed limit, but at about 2/3 of the length of the motorway there is a fixed lane drop from three to two lanes. The regional road starts with a 2×2 dual carriage way until the fork with the local road, after that the regional road continues in a 2×1 lane configuration, both sections have a 80 km/h speed limit. The local road is a normal urban main road with a 50 km/h speed limit.

There is an ATS (black circle; 1,2) that controls the flow of traffic at the fork where the regional road and local road part (both flows are controlled independently), an ATS in the town centre that limits the flow on the local road (black circle; 3) and a VSS that limits the speed on a large part of the motorway (black line along the motorway; 4). For each DTM measure the possible settings and the corresponding characteristic attribute values can be found in Table 3.1. Appendix A contains an overview of all attributes for each DTM control separately.

The route over the motorway is the fastest and takes only 37 minutes when travelling at free-flow speed, the local road is the first runner-up and takes 43 minutes at free-flow speed and finally the regional road takes 44 minutes at free-flow speed. Because all these free-flow travel times are quite close together the effects of traffic and DTM measures can be significant.

When all DTM measures are set to the median[2] we can create a kind of 'reference situation'. In this reference situation we find that after some time long queues arise in front of ATS 1 and 2. Furthermore we see a large increase in vehicle intensity just before the lane drop on the motorway. Both problems delay traffic and probably lead to very high total travel times. Furthermore we see that quite a lot of traffic passes through the city centre (1000 veh/hr) which probably causes noise nuisance.

It is therefore that optimising the selected settings for the different DTM measures could be interesting, could we reduce the total travel time and simultaneously reduce the noise in the city area?

[2]Note: since all ATS systems in this network control only a single link, we are able to use the ratio scale for these ATS systems and thus we can determine the median.

3.2 Test Network II

Although the first test network already contains a lot of different features, we felt that there were some issues that were not incorporated in the network of Wismans et al. (2009, 2010). We therefore decided to design a second test network which is based on the same principles as the test network by Wismans et al., i.e. we want a relatively small network were we can easily identify a limited number of routes, whilst having different levels of urbanisation, 'Sustainable Safety' and emission.

The additions we made were the introduction of multiple origins and destinations (three and two respectively) and the interaction between different traffic flows, using intersections and merging traffic on the motorway. We also introduced DTM measures that directly or indirectly affect multiple traffic flows. Furthermore we ensured that the intersections controlled by an ATS have a fixed capacity Q, which was not the case in Test Network I. It is therefore that we consider our test network, although still a theoretical construct, a more realistic network than Test Network I.

The second test network comprises of five centroids with a one directional traffic flow from centroid one to centroid two of 7200 pae/hr, a one directional traffic flow from centroid five to centroid 2 of 1200 pae/hr and a one directional traffic flow from centroid three to centroid four of 2400 pae/hr. The traffic from centroid one to centroid two has three possible routes: a motorway (red), a regional road (yellow) which goes through some local areas (red ellipsoids) and a local road (blue) through a town. The traffic from centroid three to centroid four has only two possible routes: a regional road (yellow) or a local road (blue) through a town centre (see Figure 3.2).

From centroid one to the three separate routes and from the three separate routes to centroid two the road network consists of regional roads that are closed for slow traffic and consist of 2×2 lanes with a 80 km/h speed limit. The motorway, the first route, consists of 2×2 lanes with a 120 km/h speed limit and halfway the traffic from centroid five to centroid two merges with the traffic from centroid one to centroid two. The regional roads, the second route, consist of 2×1 lane with a 80 km/h speed limit and are closed for slow traffic. In the local areas the speed limit changes to 50 km/h. The route through the town, the third one, starts with a part of regional road with 2×1 lane that is open to slow traffic, with a speed limit of 80 km/h. In the town itself the road is a normal urban road with a speed limit of 50 km/h. The junctions inside the town are a somewhat theoretical construct, because they can only go straight ahead, i.e. traffic from centroid one to centroid two cannot use the upper curve.

There are two ATSs present in this network that control the flow inside the town by dividing the capacity of 1400 pae/h over the two directions. There also is a VSS that limits the speed on the first half of the motorway, thereby making

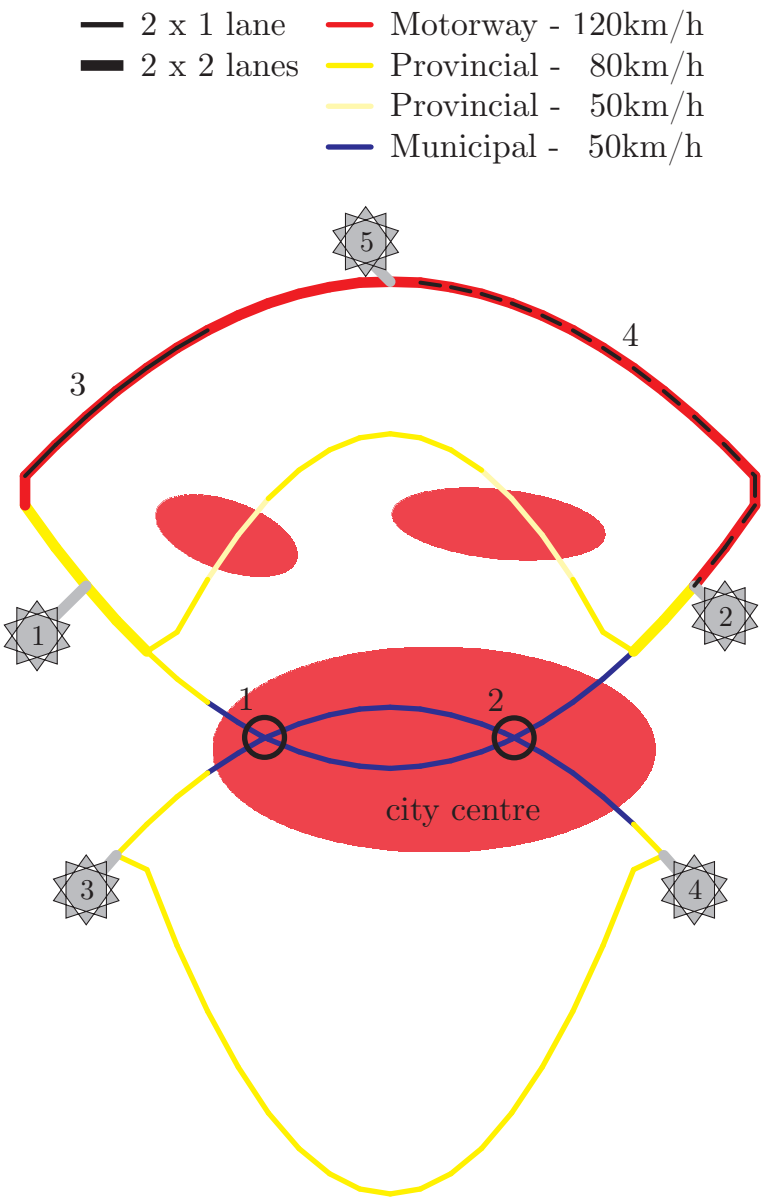


Figure 3.2: Layout of Test Network II

the motorway less attractive for traffic (and thereby reducing the problems on the second part of the motorway). Furthermore there is a VLS that offers the possibility to add an extra lane to the second half of the motorway, thereby reducing the problems that are caused by merging traffic and thus making the motorway more attractive. An overview of the corresponding characteristic attribute values can be found in Table 3.2. A full overview of all attributes for each DTM control can be found in appendix B.

setting	outflow capacity 1 – 2	outflow capacity 3 – 4
a	1200	200
b	1100	300
c	1000	400
d	900	500
e	800	600
f	700	700
g	600	800
h	500	900
i	400	1000
j	300	1100
k	200	1200

(a) ATS 1

setting	outflow capacity 1 – 2	outflow capacity 3 – 4
a	200	1200
b	300	1100
c	400	1000
d	500	900
e	600	800
f	700	700
g	800	600
h	900	500
i	1000	400
j	1100	300
k	1200	200

(b) ATS 2

setting	free-flow speed
1	120
2	100
3	80

(c) VSS

setting	road capacity
1	4400
2	6600

(d) VLS

Table 3.2: Settings of DTM Measures for Test Network II

Between centroid one and centroid two the fastest route is the motorway, which takes only 36 minutes when travelling at free-flow speed. The routes using the regional route and through the town centre are more or less equally fast and take 42 minutes at free-flow speed. Between centroid three and centroid four both routes are more or less comparable with a free-flow travel time of 33 minutes.

When both ATSs are switched to setting f, the VSS is switched to setting 2 and the VLS is switched off (i.e. we keep a two lane system) we have created a 'reference situation'. The network is modelled in such a way that the majority of the issues occur in the upper part of the network. First we find that queues arise just before the first intersection inside the city centre. These queues arise for both the traffic from centroid one to two as well as the traffic from centroid three to four. Because both ATSs allow traffic flows of 700 veh/hr in both directions we do not have any problems in front of the second ATS. These problems may arise when the settings of ATS 1 and 2 are conflicting, i.e. when one ATS prioritises the traffic from centroid one to two and the other ATS prioritises traffic from centroid three to four.

As said, most problems arise in the upper part of the network. One of the largest problems arises where the traffic from centroid five is merged with the traffic from centroid one to centroid two that decided to use the motorway. Here speed drops to less than 10% of the free-flow speed, thus causing massive delays. On the motorway were the VLS is installed we also see that slowly but surely the traffic becomes more and more dense, causing speed to drop significantly ($\pm$50% of the free-flow speed). Another problem arises just before the branching of the regional road into a ring road and a local road through the city centre. Here again queues arise, although not as serious as on the motorway, caused by the fact that too much traffic is forced on the regional road due to delays on the motorway.

At the end of the simulation we find that nearly all traffic from centroid three to four has been dealt with, whereas we still have traffic flowing from centroid one to two. This suggest that we have not distributed the disutility equally over all the routes, which might be reason to reconsider the active settings of the DTM measures. Furthermore we find that the total travel time on the network is probably average, although traffic on the motorway is hindred significantly, we also see that traffic on the regional roads (especially in the south) hardly faces delays. Similarly due to the limited traffic through the city centre, we find that noise nuisance is only limited. It might however be possible to reduce this even further by allowing more traffic to use the motorway, thereby reducing the traffic through the urban areas around the regional road. Finally we see that the CO_2 emissions are relatively high, which might be caused by the queues that arise throughout the network.

Concluding we can state that it is likely that we can improve all three objectives, e.g. using the third-lane expansion. This would cause travel time to drop

and might therefore also reduce the CO_2 emissions. Furthermore it may lead to traffic avoiding the city centre and urban areas, because the alternatives are faster, which reduces noise hindrance. In short, it is likely that changing the settings of some of the DTM measures may very well improve the overal situation on this, still somewhat imaginary, road network.

We can now use these two test networks to assess how 'good' certain approaches are. We will therefore use these test networks in chapter 4 to determine the quality of different variants of the proposed approximation methods. Furthermore they will be used to assess the three different Approximation Method Assisted NSGA-II algorithms that are developed in chapter 6.

Chapter 4

Approximation Techniques

All exact science is dominated by the idea of approximation.

Bertrand Russell (1872 – 1970)

The main goal of this chapter is to answer research question 1 (as defined in section 1.7): *'How can the objective values of the bilevel NDP be approximated?'* In section 4.1 we therefore start with a brief overview of approximation techniques and their taxonomy. It also provides a notation standard for approximation equations that will be used throughout this thesis.

The three algorithms that are found in literature (RSM, RBF and Kriging/DACE) are discussed in depth in sections 4.2 – 4.4. For each of the algorithms an explanation of the functioning of the algorithm is given and the advantages and drawbacks are discussed. Furthermore recent extensions and adaptations of the algorithms are mentioned, since they often provide improved results or require less computational effort. After this we will have answered subquestions 1a and 1b which focus on the application of approximation techniques in complex problems and their requirements.

Next we aim at answering subquestion 1c: *'How 'good' do these methods approximate the 'true' objective values?'* and section 4.5 therefore introduces the quality measures that will be used to assess the quality of the different approximation methods. The methodology used can be found in section 4.6 and the results are presented in section 4.7.

Finally we are able to determine which of the proposed approximation methods (and which variants) appear to be the best performing approximation techniques. In section 4.8 we will therefore select two (different) approximation techniques that will be used in chapter 6 to assist the NSGA-II algorithm.

4.1 Literature Overview

In literature there are three main techniques that a) are used to approximate objective functions; and b) are used in combination with GAs. These techniques are Response Surface Method (RSM), Radial Basis Functions (RBFs) and Kriging. The latter however is often used in a specific way and is then referred to as DACE. D. R. Jones (2001) provides us with a taxonomy of all metamodelling approaches and gives a good overview of what choices were made in developing these approaches. An overview of metamodelling in general can be found in the work by J. Knowles and Nakayama (2008) who clearly describe the three approaches, whereas the work of Y. Jin (2005) gives leads to which method may be preferred in which situation.

Researchers however cannot identify one single approach that is 'best', Y. Jin (2005) argues that generally speaking RSM should be applied to small simple problems whilst RBF and Kriging/DACE should be applied when difficult problems with many parameters are considered. Similarly Simpson, Peplinski, Koch and Allen (2001) state that RSM should be applied to well established problems with less than 10 parameters, whereas Kriging/DACE should be used for complex problems that require a lot of flexibility and RBF (Neural Networks; NN) should be used when many parameters are involved. However both Y. Jin and Simpson et al. find that there is no single good algorithm that can be selected up front.

Georgopoulou and Giannakoglou (2009) provide three different ways to incorporate metamodels into GAs, which are listed below:

off-line trained metamodels without feedback Based on a training set, generated using some sampling method, a metamodel is created. All other solutions are based on this metamodel. This approach is therefore especially suitable when there is no possibility to evaluate new samples.

off-line trained metamodels with feedback Based on a training set (generated using some sampling method) a metamodel is created, which is used to determine a new set of 'interesting' solutions. These solutions are then evaluated by the exact model. Every g generations this information is used to update the metamodel in order to improve the estimates.

on-line trained local[1] metamodels For each population member (in the offspring set) a specially designed metamodel is created, using the information from the previous evaluated solutions. Based on the results of the metamodels a selection of promising new population members is evaluated.

[1]Local metamodels are here defined as models which only use neighbouring solutions to approximate the objective value of a certain point, whereas global metamodels use all known solutions and approximate the complete solution space.

n	number of solutions (points)
d	number of dimensions of the problem, in this case the number of DTM measures
f	exact value of f
$\tilde{f}$	estimated value of f
$\hat{f}$	best estimator of f
$\|f_i - f_j\|$	distance between f_i and f_j
$\mathbf{f}$	a vector of f
$\mathbf{F}$	a matrix of f
$\|\mathbf{F}\|$	determinant of matrix $\mathbf{F}$
$\mathbf{F}^T$	transpose of matrix $\mathbf{F}$
$\mathbf{F}^{-1}$	inverse of matrix $\mathbf{F}$
$\mathbf{F}^+$	pseudoinverse of matrix $\mathbf{F}$, an inverse which can also be applied to non-square matrices

Table 4.1: Notation in Approximation Methods

The three algorithms are each capable of working in combination with any of these three approaches. In our case the first approach is not advisable, for it is possible to exactly evaluate solutions. It therefore seems wise to use this exact knowledge to update the metamodel. The second approach tries to create a global metamodel, however this is a difficult task, especially when a lot of parameters are involved. It gives quite a lot of power to the metamodel, which is responsible for searching good solutions in a complex environment. Finally the third and last approach creates local metamodels, that are specifically designed to predict the value of a single solution.

In this research we focus on global metamodels, since we would like to find an approximation model for all solutions. Although local metamodels are much easier to optimise,[2] it requires a new optimisation for each solution that has to be evaluated. This seems undesirable, especially since GAs require us to evaluate many solutions.

In order to maintain a clear view on what is meant in the approximation related equations, we will use the notation from Table 4.1 throughout this thesis.

[2]The term optimise is used to indicate the process of finding the best variable values for the approximation method. The best values are those values that ensure that the goal function of the approximation method is optimised (e.g. minimising the error or maximising the likelihood).

4.2 Response Surface Method

The Response Surface Method (RSM) was introduced by Box and Wilson (1951) and was originally intended as a guideline to designing experiments. They consider a problem in which they approximate an objective using a polynomial function, which is usually notated as follows:

$$f = b_0 + \mathbf{bx} + \mathbf{x}^T\mathbf{Bx} + \ldots + \epsilon \tag{4.1}$$

Here $\mathbf{b}$ is a vector of b_i and $\mathbf{B}$ is a square matrix of B_{ij} $(i, j = 1 \ldots n)$. If we assume a problem with two variables this gives the following approximation (note that $B_{12} = B_{21}$):

$$f = b_0 + b_1x_1 + b_2x_2 + B_{11}x_1^2 + B_{22}x_2^2 + 2B_{12}x_1x_2 + \epsilon \tag{4.2}$$

If we consider a problem in which we have investigated a number of points in our problem space, we would like to find the Least Square Fit through these points. However for this type of problem it is much more convenient to rewrite the problem into a different formulation (equation 4.3), where x_{im} denotes the value of the i^{th} variable for the m^{th} point.

$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{11}^2 & x_{21}^2 & x_{11}x_{21} \\ 1 & x_{12} & x_{22} & x_{12}^2 & x_{22}^2 & x_{12}x_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & x_{1n}^2 & x_{2n}^2 & x_{1n}x_{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \tag{4.3}$$

If we assume that $\epsilon \sim (0, \sigma)$, we can rewrite this problem to:

$$\mathbf{f} = \mathbf{Xa} + \boldsymbol{\epsilon} \tag{4.4}$$

Then the solution vector $\mathbf{a}$ can be found by determining:

$$\mathbf{a} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{f} \tag{4.5}$$

Or the much shorter version:

$$\mathbf{a} = \mathbf{X}^+\mathbf{f} \tag{4.6}$$

And the approximation of the objective value for a new point can be described as:

$$\tilde{f} = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 + a_5x_1x_2 \tag{4.7}$$

Box and Wilson (1951) suggest the use of second-order polynomials, because higher order polynomials require a lot of points (measurements) just to estimate the approximation function. For a k-order full polynomial with d variables the minimum number of points needed is:

$$n_{min} = \sum_{i=1}^{k} d^k + 1 \tag{4.8}$$

Which is a function that increases rapidly with k. Therefore often a cubic polynomial is used, which reduces the total number of points that is needed to:

$$n_{min} = kd + 1 \tag{4.9}$$

Box & Wilson propose a method in which one, when provided with multiple designs, can determine for which design the bias, caused by leaving out higher order polynomials, is larger. This gives researchers a tool to choose better experimental designs in order to be able to maximise the results of their approximation.

Another useful tool that is provided by Box and Wilson (1951) in their paper is the updating of the estimators and error terms without inverting large arrays. This is beneficiary, because inverting large arrays is complex and costs quite large amounts of computational power. If we would now consider the situation in which already N_1 experiments have been used for the approximation function and another N_2 experiments have just been performed. If we let $\mathbf{C}^{-1}$ denote $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ of the set of N_1 points and $\mathbf{W}$ denotes a similar matrix as $\mathbf{X}$ only for the N_2 new points with objective values $\mathbf{Z}$, then the updated matrix $\mathbf{C}_0^{-1}$, the new estimators $\mathbf{a}_0^T$ and the new residual sums of squares s_0 can be determined using:

$$\begin{aligned} \mathbf{C}_0^{-1} &= \mathbf{C}^{-1} - \mathbf{J}^T\mathbf{G}\mathbf{J} \\ \mathbf{a}_0^T &= \mathbf{a}^T + \mathbf{\Delta}^T\mathbf{G}\mathbf{J} \\ s_0 &= s + \mathbf{\Delta}^T\mathbf{G}\mathbf{\Delta} \end{aligned} \tag{4.10}$$

where $\mathbf{J} = \mathbf{W}\mathbf{C}^{-1}$, $\mathbf{G} = (\mathbf{I} + \mathbf{R})^{-1}$, $\mathbf{R} = \mathbf{W}\mathbf{C}^{-1}\mathbf{W}^T$ and $\mathbf{\Delta} = \mathbf{Z} - \mathbf{W}\mathbf{a}$.
Another issue that deserves some attention is underdetermined problems. A problem is said to be underdetermined if the number of known points that are used to estimate the parameters (in this case the vector $\mathbf{a}$), is larger than the number of parameters that have to be estimated. This situation can easily be recognised since in this case the matrix $\mathbf{X}$ has more columns than rows. In that case we are unable to find a unique solution that minimises the error. In fact we could find a set of solutions that all have the same minimal error, but have very different parameter values. We therefore use the Moore-Penrose pseudoinverse, which is a generalized inverse that can also be applied to rectangular matrices (see e.g. Ben-Israel & Greville, 2003). One of the advantages of this specific pseudoinverse is that we exactly know which solution is found, i.e. we find a unique solution. This unique solution does not only minimise the error in objective value, but also minimises the differences between the parameter values $\mathbf{a}$.

The main reason why RSM is so popular is that it is quite simple, easy to understand and is able give a good indication which solutions one should investigate next. A good example of such an approach can be found in Bucher

and Bourgund (1990) who apply RSM to the reliability analysis of structural problems. They show that RSM is capable of finding good solutions whilst still being much faster than finding an exact solution. Similarly Allen, Yu and Bernshteyn (2000) applied RSM to the problem of designing fasteners in the automotive industry. Not only do they use the factorial design approach[3] for finding a near optimal solution, they also use multiple models (i.e. they exclude some of the combination parameters) and thus make use of the ability to determine which model is less sensitive to bias caused by higher order polynomials. Finally Yunker and Tew (1994) use both RSM as well as a GA to set the parameters of a computer network in such a way that the user costs are minimised. Although GA outperforms RSM (by nearly 25%) they did not consider computational time to be an issue. Would we also take computational effort into account, we see that RSM only needs 25 runs per search, whilst GA needs 682. This clearly indicates that RSM might very well be an interesting technique to quickly find good solutions.

Although RSM is quite often mentioned as a possible metamodel to improve or accelerate GAs (see e.g. Büche, Schraudolph & Koumoutsakos, 2005; Bhattacharya, 2008; Georgopoulou & Giannakoglou, 2009) the true application of RSM is not that frequent. This is probably due to the fact that the behaviour of objectives of problems that require GAs are too complex to be approximated by RSM. Y. Jin (2005) states that RSM can be used when the number of variables is quite low and the complexity is not too high. For problems with a high complexity, i.e. problems with objective functions which are difficult to predict, usually higher order (cubic, quartic or even higher) RSM is needed. According to Y. Jin in those cases more intelligent approaches such as RBF or Kriging should be applied. However there are a couple of recent studies that do apply RSM (although often modified) to improve the results of GAs. Already in 2001 Khoo and Chen designed a GA that used RSM to determine possible interesting locations. Ölvander (2005) applies RSM in a different way, he uses the partial derivative of the approximation function and the sensitivity (on the Pareto front) to understand the underlying processes of his (hydraulic pump) design problem. C. Kim, Wang and Choi (2005) improve the existing RSM approach by using multiple approximation functions and determining the expected value of a certain point by weighing these different functions based on a (predefined) distance function. They thereby prevent RSM from fitting a straight line through all the points and create a much more versatile approximation function. Finally Zhou, Ong, Lim and Lee (2007) use both RSM and RBF as local search techniques in order to improve the results that have been evaluated by the exact objective function.

It can be concluded that although RSM is a simple and intuitive approach to

[3]See the book by Box, Hunter and Hunter (2005) or Law and Kelton (2000) for an overview of experimental design and factorial design in particular.

approximation it has not been used that much in literature. Major drawbacks of the approach are that higher order relations as well as problems with many variables (which require many observations) are expensive to model. However for less complex relationships RSM can provide a very quick and accurate approximation of the objective values.

4.3 Radial Basis Functions

Among the people who more or less introduced RBF were Moody and Darken (1989). They proposed a network approach in which the radial between a centre point and a solution measured the weight to which that centre point contributed to the solution. Currently Gutmann (2001) is considered to be one of the main contributors, by proving the necessity of additional variables for specific distance functions. The concept of a RBF (or more precise a Radial Basis Function Network) is depicted in Figure 4.1.

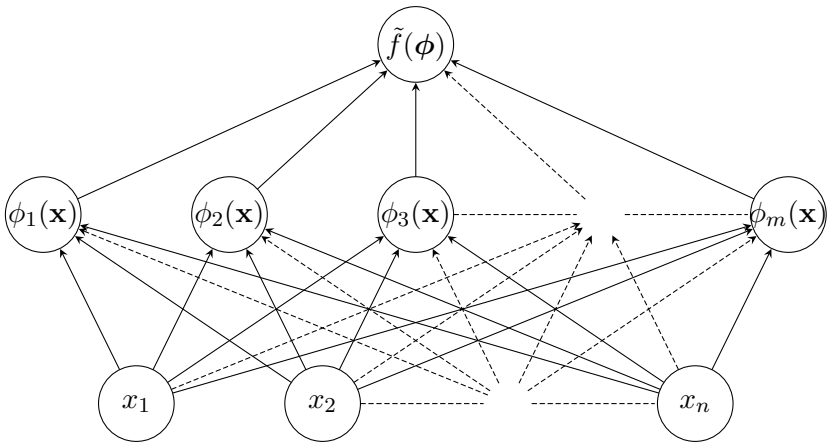


Figure 4.1: Radial Basis Function Network

Here the output $\tilde{f}$ consists of the linear weighted sum of m radial basis functions, which each are functions of n different values for x. If we now assume that $m = n$ and $\phi_i(\mathbf{x}_i)$ (i.e. the number of known solutions is equal to the number of individual RBF distance functions and each RBF distance function is only dependent on a single known point), we can use the slightly more generic approach by Gutmann (2001) who defines the RBF as follows:

$$\tilde{f}(\mathbf{x}) = \sum_{i=1}^{m} \lambda_i \phi \left(\|\mathbf{x} - \mathbf{x}_i\| \right) + p(\mathbf{x}) \tag{4.11}$$

Here $p(\mathbf{x})$ represents a polynomial function of $\mathbf{x}$. We can however, as is often done, abbreviate $\phi \left(\|\mathbf{x}_i - \mathbf{x}_j\| \right)$ to $\phi \left(r_{ij} \right)$. Next Gutmann proposes five different methods to determine the distance between the two points, a sixth is

added based on Billings and Zheng (1995):

$$\begin{array}{ll} \phi(r) = r & \text{(linear)} \\ \phi(r) = r^3 & \text{(cubic)} \\ \phi(r) = r^2 \log r & \text{(thin plate spline)} \\ \phi(r) = \sqrt{r^2 + c^2} & \text{(multiquadratic)} \\ \phi(r) = e^{-cr^2} & \text{(Gaussian)} \\ \phi(r) = \frac{1}{\sqrt{r^2+c^2}} & \text{(inverse multiquadratic)} \end{array} \tag{4.12}$$

Gutmann (2001) gives a proof of the order of $p(\mathbf{x})$ that is needed, something that is summarised in Holmström, Quttineh and Edvall (2008). He states that in a cubic or thin plate spline RBF a first order polynomial is needed, whilst only a fixed value is needed when linear, multiquadratic or inverse multiquadratic RBFs are used. Finally when Gaussian RBFs are used there is no need to add a polynomial function. Note that this does not mean that one cannot find a solution if no polynomial function $p(\mathbf{x})$ is used. Research has been done to determine which RBF provides the best solutions, Coulomb, Kobetski, Costa, Maréchal and Jönsson (2003) states that Gaussian has overall a good performance, whilst multiquadratic approaches only perform well on specific test problems (and score poorly in real-life cases).

Mullur and Messac (2006) clearly describe how a solution can be found for a RBF problem. For notation assume that we have n measurements $\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ where $\mathbf{x}_i$ describes a known solution in d dimensions (which means that there are d design variables). For now we assume single objective optimisation and thus has each of these measurements an exact objective value $f_1, \ldots, f_n \in \mathbb{R}$. Mullur and Messac use a multiquadratic RBF and use a static value for c.[4] Now define:

$$\mathbf{\Phi} = \begin{bmatrix} \phi(r_{11}) & \phi(r_{12}) & \ldots & \phi(r_{1n}) \\ \phi(r_{21}) & \phi(r_{22}) & \ldots & \phi(r_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(r_{n1}) & \phi(r_{n2}) & \ldots & \phi(r_{nn}) \end{bmatrix} \tag{4.13}$$

If we combine equation 4.11 with the knowledge that $p(\mathbf{x})$ is empty when using Gaussian RBF and the new notation described above and in equation 4.13 then we can describe the problem in the same way Mullur and Messac (2006) do:

$$\mathbf{f} = \mathbf{\Phi}\boldsymbol{\lambda} \tag{4.14}$$

Note that this indicates that f is assumed to be a function of the distances in solution space. Of course then the main goal is to find the vector $\boldsymbol{\lambda}$ which is the vector that contains the values for λ_i which describe the linear weights

[4] It is not possible to optimise the value of c using the standard RBF approach, since c is not a linear but a power term. It is however possible to try to optimise the problem for different values of c and select that approach that provided the most plausible results.

for the basis functions that minimise the error. The solution can be found (as long as $\mathbf{\Phi}$ is not singular) by:

$$\boldsymbol{\lambda} = \mathbf{\Phi}^{-1}\mathbf{f} \tag{4.15}$$

Mullur and Messac (2006) continue by expanding the existing RBF for good reason. In Figure 4.2 the behaviour with respect to the distance between two points is shown for different values of c using a multiquadratic distance function $\phi(r) = \sqrt{c^2 + r^2}$. It is obvious that if two solutions are further apart the value of the RBF for that solution also increases. The suggestion, made by Mullur and Messac, however, is that for solutions that are far apart (i.e. large positive or negative values of r) the effects of distance are underestimated, since in the long run $\lim_{r\to\infty} \phi(r) = r$.

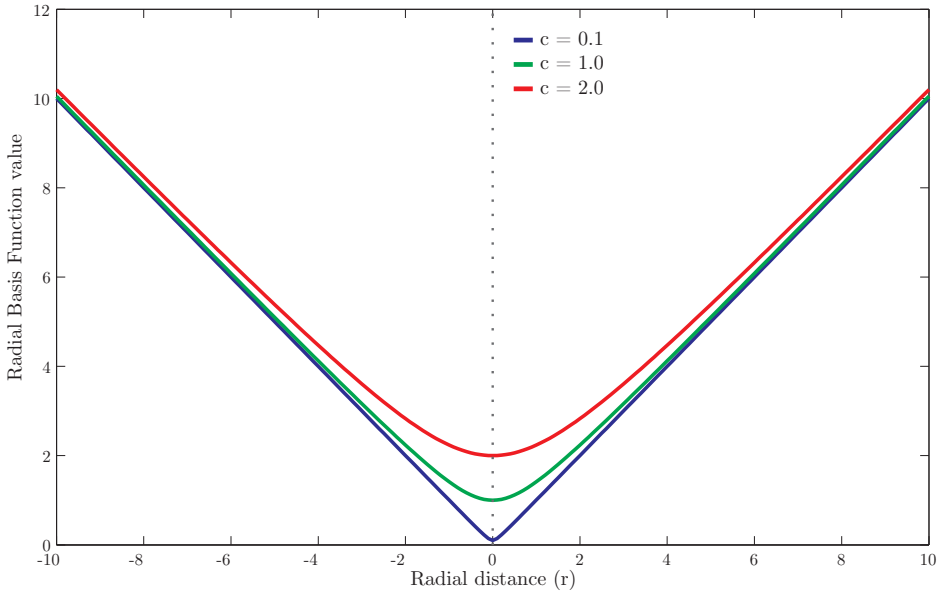


Figure 4.2: Multiquadratic Radial Basis Function

It is therefore that Mullur and Messac (2005, 2006) decided to improve their method by creating an extended RBF as shown in Figure 4.3 which can be described by:

$$\tilde{f}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \phi\left(\|\mathbf{x} - \mathbf{x}_i\|\right) + \sum_{i=1}^{n}\sum_{j=1}^{d} \left\{\alpha_{ij}^{L}\phi^{L}(\xi_j^i) + \alpha_{ij}^{R}\phi^{R}(\xi_j^i) + \beta_{ij}\phi^{\beta}(\xi_j^i)\right\} \tag{4.16}$$

The values of $\phi^L(\xi_j^i)$, $\phi^R(\xi_j^i)$ and $\phi^\beta(\xi_j^i)$ can be determined using the equations from Table 4.2 where $\xi_i = \|x - x_i\|$ and ξ_i^j is the value of ξ_i for the j^{th} dimension. Furthermore there are two (prefixed) scaling parameters $n \geq 2$ and γ.

For the example in Figure 4.3 we selected $n = 2$ and $\gamma = 5$. Furthermore (for convenience) we assumed that $\lambda_i = 1$ and $\alpha_{ij}^L = \alpha_{ij}^R = \beta_{ij} = 1$.

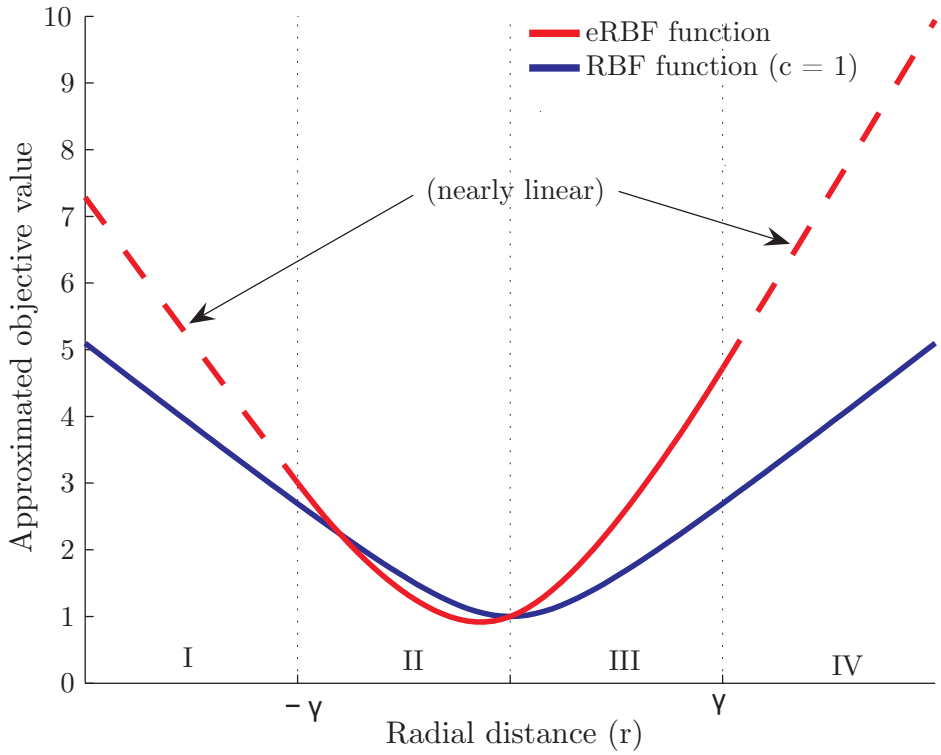


Figure 4.3: extended Radial Basis Function

Region	Range of ξ_i^j	ϕ^L	ϕ^R	ϕ^β
I	$\xi_i^j \leq -\gamma$	$(-n\gamma^{n-1})\xi_i^j + (1-n)\gamma^n$	0	ξ_i^j
II	$-\gamma \leq \xi_i^j \leq 0$	$(\xi_i^j)^n$	0	ξ_i^j
III	$0 \leq \xi_i^j \leq \gamma$	0	$(\xi_i^j)^n$	ξ_i^j
IV	$\xi_i^j \geq \gamma$	0	$(n\gamma^{n-1})\xi_i^j + (1-n)\gamma^n$	ξ_i^j

Table 4.2: extended Radial Basis Function Values for ϕ

The problem of equation 4.16 can be solved similarly to the method before, by describing a new matrix of coefficients $\bar{A}$ and solution vector $\bar{\alpha}$:

$$\bar{\alpha} = \begin{Bmatrix} \boldsymbol{\lambda}^T & (\boldsymbol{\alpha}^L)^T & (\boldsymbol{\alpha}^R)^T & \boldsymbol{\beta}^T \end{Bmatrix}^T \tag{4.17}$$

$$\bar{\mathbf{A}} = \begin{bmatrix} \phi(r_{11}) & \dots & \phi(r_{n1}) \\ \vdots & \ddots & \vdots \\ \phi(r_{1n}) & \dots & \phi(r_{nn}) \\ \phi^L_{111} & \dots & \phi^L_{n11} \\ \vdots & \ddots & \vdots \\ \phi^L_{1nd} & \dots & \phi^L_{nnd} \\ \phi^R_{111} & \dots & \phi^R_{n11} \\ \vdots & \ddots & \vdots \\ \phi^R_{1nd} & \dots & \phi^R_{nnd} \\ \phi^\beta_{111} & \dots & \phi^\beta_{n11} \\ \vdots & \ddots & \vdots \\ \phi^\beta_{1nd} & \dots & \phi^\beta_{nnd} \end{bmatrix}^T \tag{4.18}$$

The solution can then be found using the pseudoinverse of $\bar{\mathbf{A}}$, $\bar{\mathbf{A}}^+$, by solving:

$$\bar{\alpha} = \bar{\mathbf{A}}^+ \mathbf{f} \tag{4.19}$$

The main advantage of this approach is that the importance of the RBF decreases strongly outside a certain 'trusted region' (γ) when using the right scaling parameters. Especially for low values of c for multiquadratic distance functions or high values of c for Gaussian or inverse multiquadratic distance functions (i.e. when the distance function is quite flat) this could improve the accuracy of the approximation. Another benefit of the eRBF approach is that although the equations seem quite complex, the problem still is a linear problem and can thus be solved quite easily. A possible drawback of the approach is that it assumes that a good value of c can be determined in advance, whilst determining the parameter values is already a problem in itself (Büche et al., 2005). According to Mullur and Messac (2006) the estimation is not a problem since $c = 1, \gamma = 1$ provided consistently accurate results.

Praveen and Duvigneau (2009) suggests that one could estimate the value of c by minimising the error found by applying the 'leave-one-out' approach. Here one tries to estimate the value of a certain point i from the entire set of evaluated solutions, by using all solutions except i in the RBF function. Although this approach seems to make sense (one tries to minimise the error) it results in quite a complex problem, since for each i a problem of size $(n-1) \times (n-1)$ is created. One could however decide to use only a small

subset to estimate the value of c. Tanaka, Mizoguchi and Takami (2007) use a 'rule-of-thumb' approach and determine c based on maximum distance between two points, $r_{\max}$ and the dimension and size of the problem, according to:

$$c = \frac{\sqrt[0.5d]{dn}}{r_{\max}^2} \tag{4.20}$$

Büche et al. (2005) use a similar approach and determines c using:

$$c = r_{\max}(dn)^{\frac{-1}{d}} \tag{4.21}$$

Since the leave-one-out method is computationally expensive (one needs to build n RBF models) it seems that the approach of Mullur and Messac (2006), choosing a fixed value of c, is quite useful. However it would be a good idea to test different values of c and see whether the results are consistent.

Coulomb et al. (2003) identify another problem that arises when using RBF in solving real-life problems, which is determining the distance between two points in the solution space. From a mathematical point of view the solution is quite simple, one uses the Euclidean distance:

$$r_{ij} = \sqrt{\sum_d (x_j^d - x_j^d)^2} \tag{4.22}$$

However in reality the scale of different dimensions could be important, measuring in metres or millimetres does not give a different solution, but would affect the distance calculated using Euclidean distances. Coulomb et al. therefore suggest the use of a normalised radius, which is defined by:

$$r_{ij}^2 = \sum_{d=1}^{D} \left(\frac{x_i^d - x_j^d}{r_d} \right)^2 \tag{4.23}$$

Where r_d is a characteristic distance for dimension d. Coulomb et al. suggest using the minimum and maximum value of x^d, $x_{\min}^d$ and $x_{\max}^d$, and the number of intervals in dimension d, denoted by I_d.[5] Then r_d can be determined using:

$$r_d = \frac{x_{\max}^d - x_{\min}^d}{I_d - 1} \tag{4.24}$$

Radial Basis Functions are quite popular as a metamodelling method in recent publications, although often researchers resolve to different methods (see e.g. Büche et al., 2005; Emmerich, Giannakoglou & Naujoks, 2006; J. Knowles,

[5]This approach, where we need to determine the number of intervals, is especially convenient when optimising discrete problems. When continuous problems are considered other normalisation techniques, e.g. dividing by the average value, can be applied.

2006; Bhattacharya, 2008). Generally speaking are RBFs appraised for their ability to approximate the behaviour of difficult multivariable problems (see e.g. Hussain, Barton & Joshi, 2002; Coulomb et al., 2003; Büche et al., 2005; Mullur & Messac, 2006; Tanaka et al., 2007; Zhou, Ong, Nair, Keane & Lum, 2007), which is part of the reason why they are being used in the design of complex equipment (see: Karakasis & Giannakoglou, 2006; Karakasis, Koubogiannis & Giannakoglou, 2007; Messac & Mullur, 2008). Some however do not fully agree and claim that simpler approaches such as RSM might be better in some cases (Zhou, Ong, Lim & Lee, 2007).

The most common way of using RBF is to create a selection of solutions that is going to be exactly evaluated (Georgopoulou & Giannakoglou, 2009). This is also the way in which Karakasis et al. (2007) use RBF in their hierarchical model. In fact they approximate each solution that is generated by the EA with the RBF and select the best $x\%$ to be evaluated by the exact model. Georgopoulou and Giannakoglou give much more power to the RBF, in their opinion RBF should be used to determine the objective values for solutions that are in areas that are well known,[6] exact evaluations are then only needed to extend the search area and maintain the diversity in solutions. Finally Messac and Mullur (2008) have developed a method that focusses on identifying the Pareto set by adding a set of pseudo points. These points (that represent the corners of a m-dimensional box) are then given fitness values that are dependent on the distance towards the nearest evaluated solution. By underestimating the fitness of these pseudo points and including them in the eRBF method the approximation will strongly favour solutions that are close to the already evaluated solutions (and proved to be good), however solutions that are further away are more or less ignored. This method is therefore suitable for problems where there is a clear relation between solution space and objective space, i.e. solutions that are far away in solution space also are far away in objective space.

RBF are generally appraised in literature, especially their capability of mimicking difficult behaviour by using little more than the distance to a couple of near solutions. However in order to give good predictions in all parts of the solutions space a careful spread of solutions should be maintained.

4.4 Kriging/DACE

Kriging is named after the South African mining engineer Krige, who developed the approach in the 1950s. However he failed to formalise the method, something that was done by Matheron (1963). For now we follow the standard

[6] Determining whether a solution is in an area that is well known is dependent on the number of solutions that are close in solution space. Georgopoulou and Giannakoglou (2009) use a set of threshold values that is used to determine the number (and minimal number) of solutions that are close.

work by D. R. Jones, Schonlau and Welch (1998), in their explanation they give good insight in why certain assumptions can be made.

D. R. Jones et al. start working from the basics of approximation, which is regression. Now assume that $\mathbf{x} = (x_1, \ldots, x_d)$ denotes a single solution, β_h are unknown coefficients, $f(\mathbf{x})$ is a linear or nonlinear function of $\mathbf{x}$ and ϵ is a normally distributed independent error term with mean zero and variance σ^2. Then the objective value can be determined using:

$$y(\mathbf{x}) = \sum_h \beta_h f_h(\mathbf{x}) + \epsilon \tag{4.25}$$

Now D. R. Jones et al. (1998) state that the assumption of normally distributed independent error terms is *'blatantly false'*. When considering a continuous objective function it is likely that if two solutions (e.g. $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$) are close together, the errors $\epsilon(\mathbf{x}^{(i)})$ and $\epsilon(\mathbf{x}^{(j)})$ are also close together. This behaviour suggests that instead of being independent the error terms are correlated, the closer they are together the higher the correlation. The Kriging method therefore suggests the use of a specific distance function:

$$c\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sum_{h=1}^{d} \theta_h \left\| x_h^{(i)} - x_h^{(j)} \right\|^{p_h} \tag{4.26}$$

Here $\theta_h \geq 0$ and $p_h \in (1, 2)$ are scaling parameters for respectively the sensitivity and the smoothness of the weighted distance curve. Then the correlation of this relationship can be described by:

$$r_{ij} = \text{Corr}\left[\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(i)})\right] = e^{-c\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)} \tag{4.27}$$

D. R. Jones et al. conclude that this function is, due to the scaling parameters, so powerful that it is possible to dispose the regression terms, except the constant. This gives the model that is used in stochastic process approaches, where $\epsilon(\mathbf{x})$ is normally distributed with mean 0 and variance σ^2 but correlated as described in equation 4.27:

$$y(\mathbf{x}) = \mu + \epsilon(\mathbf{x}) \tag{4.28}$$

The approach from equation 4.26 – 4.28 has become quite popular under the name DACE, an abbreviation of 'Design and Analysis of Computer Experiments' which is the famous paper by Sacks, Welch, Mitchell and Wynn (1989). The main goal is now to find the values of the $2 + 2d$ parameters $(\mu, \sigma^2, \theta_1, \ldots, \theta_d, p_1, \ldots, p_d)$ of DACE using the maximum likelihood function for non-independent variables:

$$L(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma|}} e^{\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})^T\right]} \tag{4.29}$$

Here Σ is denoting the covariance matrix and thus, by substituting $\mathbf{R}$ as the $n \times n$ matrix of r_{ij} with $i, j = 1, \ldots n$ and $\mathbf{1}$ as a n-vector of ones, the likelihood can be calculated using:

$$L\left(\mu, \sigma^2, \boldsymbol{\theta}, \mathbf{p}\right) = \frac{1}{(2\pi)^{n/2}\left(\sigma^2\right)^{n/2}\sqrt{|\mathbf{R}|}} e^{\left[-\frac{(\mathbf{f}-\mathbf{1}\mu)^T\mathbf{R}^{-1}(\mathbf{f}-\mathbf{1}\mu)}{2\sigma^2}\right]} \tag{4.30}$$

When $\boldsymbol{\theta}$ and $\mathbf{p}$ are known the estimators of μ and σ^2, $\hat{\mu}$ and $\hat{\sigma}^2$, can be estimated using:

$$\hat{\mu} = \frac{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{f}}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}} \tag{4.31}$$

$$\hat{\sigma}^2 = \frac{\left(\mathbf{f} - \mathbf{1}\hat{\mu}\right)^T \mathbf{R}^{-1}\left(\mathbf{f} - \mathbf{1}\hat{\mu}\right)}{n} \tag{4.32}$$

In order to obtain the expected value $\tilde{f}(\mathbf{x})$ of a new point $\mathbf{x}$ we need the vector $\mathbf{r}^T = (r_{x1}, \ldots, r_{xn})$, after which:

$$\tilde{f}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T\mathbf{R}^{-1}\left(\mathbf{f} - \mathbf{1}\hat{\mu}\right) \tag{4.33}$$

Finally due to the model used, the variance at a known point is zero, whilst the variance at a point far away from all other data is estimated to be $\hat{\sigma}^2$. In fact the error can be predicted at all points using:

$$s^2(\mathbf{x}) = \hat{\sigma}^2\left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{\left(1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{r}\right)^2}{\mathbf{r}^T\mathbf{R}^{-1}\mathbf{1}}\right] \tag{4.34}$$

The derivation of equation 4.26 – 4.34 can be found in Sacks et al. (1989).

The Kriging method described above (using only one constant μ which is unknown) is generally known as ordinary Kriging. Literature however does also consider two other approaches, simple Kriging, where there is one constant μ which is known, and universal Kriging, where there is a polynomial function included. Cressie (1990) concluded that the use of the ordinary Kriging method and predicting μ by the best linear unbiased estimator $\hat{\mu}$ is to be preferred over simple Kriging, because it is difficult to select a good value of μ up front. Journel and E. (1989) compared the results of using ordinary and universal Kriging and concluded that the use of higher order terms did not improve the results. It therefore seems that the ordinary Kriging method, as described above, is the most suitable method in nearly every case.

Probably one of the most difficult problems in Kriging is the estimation of the parameters θ_h and p_h (for μ and σ^2 we already found reasonable estimators).[7]

[7]Note that we can substitute the values for $\hat{\mu}$ and $\hat{\sigma}^2$ from equations 4.31 and 4.32 into equation 4.30. In that case we end up with an equation that is only dependent on $\boldsymbol{\theta}$ and $\mathbf{p}$.

This is a problem that is also recognised by the developers of DACE (Sacks et al., 1989), who suggest the use of maximum likelihood estimation (MLE). The problem with this solution method however is that it is also a complex problem (not convex) and thus cannot be solved to optimality in polynomial time. Instead one has to resolve to gradient based methods, such as SA or GA, to solve this problem, without knowing whether a global optimum has been reached (Büche et al., 2005).

Dietrich and Osborne (1991) found an analytical solution to a similar problem, with the limitation that the covariance function has the form:

$$r_{ij} = \theta_0 a\left(\|i-j\|\right) + \theta_1 b\left(\|i-j|, l\right) \tag{4.35}$$

Of course this severely limits the flexibility that was considered to be one of the assets of Kriging and is therefore less suitable for more complex problems.

Therefore often an assumption is made on the value of p_h. When $p_h = 1$ the correlation (see equation 4.27) is reduced to an exponential function (equation 4.36), whereas $p_h = 2$ gives a Gaussian function (equation 4.37).

$$e^{-\sum_{h=1}^{d} \theta_h \|x_h^{(i)} - x_h^{(j)}\|} \tag{4.36}$$

$$e^{-\sum_{h=1}^{d} \theta_h \|x_h^{(i)} - x_h^{(j)}\|^2} \tag{4.37}$$

This leaves only one set of undefined parameters $\boldsymbol{\theta}$ (in the case of universal Kriging also a set $\boldsymbol{\beta}$) which is a problem that can be solved much more conveniently. Martin and Simpson (2005), who use universal Kriging, mention two approaches that could be used, Maximum Likelihood Estimation (MLE) and Cross Validation (CV).

When applying MLE Martin and Simpson (2005) do not maximise the likelihood function that was given earlier (see equation 4.30) but the natural logarithm of this problem. The extra term is because they applied universal Kriging, in which $\mathbf{P}$ is used to describe the coefficients of the higher order polynomes:

$$\ln(\boldsymbol{\beta}, \sigma^2\boldsymbol{\theta}) = -\frac{n}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{2}\ln\left(|\mathbf{R}|\right) - \frac{1}{2\sigma^2}\left(\mathbf{f} - \mathbf{P}\boldsymbol{\beta}\right)^T \mathbf{R}^{-1}\left(\mathbf{f} - \mathbf{P}\boldsymbol{\beta}\right) \tag{4.38}$$

In that case the optimal values of $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ can be determined using:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{P}^T\mathbf{R}^{-1}\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{R}^{-1}\mathbf{f} \tag{4.39}$$

$$\hat{\sigma}^2 = \frac{1}{n}\left(\mathbf{f} - \mathbf{P}\hat{\boldsymbol{\beta}}\right)^T \mathbf{R}^{-1}\left(\mathbf{f} - \mathbf{P}\hat{\boldsymbol{\beta}}\right) \tag{4.40}$$

The problem can then be solved by iteratively determining the values of $\hat{\boldsymbol{\beta}}$, $\hat{\sigma}^2$, substituting these in equation 4.38 and then maximising that problem for $\boldsymbol{\theta}$. After some iterations convergence is reached (i.e. the maximum likelihood does not change much) and a solution has been found. It is important to note

that when matrix $\mathbf{R}$ is singular, the determinant $|\mathbf{R}| = 0$ at which point it is impossible to determine the logarithm of the determinant.[8]

CV works slightly different, it assumes that all parameters are constant $(\sigma^2, \boldsymbol{\beta}, \boldsymbol{\theta})$ and the error of these values is determined by creating n Kriging models, which predict the value of point i based on the $n-1$ other solutions. Now assume a diagonal matrix $\mathbf{Q}$ where the values are the inverse of the diagonal of $\mathbf{R}^{-1}$ (note: this implies a square matrix). Furthermore assume $g = \mathbf{R}^{-1}\mathbf{f}$ and $w = \mathbf{R}^{-1}\mathbf{P}$, the total error can then be described by:

$$e = \mathbf{Q}\left(g - w\boldsymbol{\beta}\right) \tag{4.41}$$

And the best estimate $\hat{\boldsymbol{\beta}}$ can be determined using:

$$\hat{\boldsymbol{\beta}} = \left(w^T\mathbf{Q}^2 w\right)^{-1} w^T\mathbf{Q}^2 g \tag{4.42}$$

Again the solution can be determined by iteratively determining $\hat{\boldsymbol{\beta}}$ and substituting these values in equation 4.41, which can then be minimised for $\boldsymbol{\theta}$.

Finally Jeong, Obayashi and Yamamoto (2006) use a better defined iterative approach (see Mardia and Marshall (1984) for a full derivation) in which they try to maximise the shorter likelihood function:

$$\ln\left(\mu, \sigma^2, \boldsymbol{\theta}\right) = -\frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|\mathbf{R}|) \tag{4.43}$$

Then the vector $\boldsymbol{\theta}$ can be updated using:

$$\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{\text{old}} + \mathbf{B}^{+}\frac{\partial \ln}{\partial \boldsymbol{\theta}} \tag{4.44}$$

Given that *tr* indicates the sum over the diagonal, the partial derivative and $\mathbf{B}$ can be determined using:

$$\frac{\partial \ln}{\partial \theta_k} = -\frac{1}{2}\operatorname{tr}\left(\mathbf{R}^{+}\frac{\partial \mathbf{R}}{\partial \theta_k}\right) - \frac{1}{2\sigma^2}\left(\mathbf{f} - \mathbf{1}\mu\right)^T\mathbf{R}^{+}\frac{\partial \mathbf{R}}{\partial \theta_k}\mathbf{R}^{+}\left(\mathbf{f} - \mathbf{1}\mu\right) \tag{4.45}$$

$$\mathbf{B}_{ij} = \frac{1}{2}\operatorname{tr}\left(\mathbf{R}^{+}\frac{\partial \mathbf{R}}{\partial \theta_i}\mathbf{R}^{+}\frac{\partial \mathbf{R}}{\partial \theta_j}\right) \tag{4.46}$$

Mardia and Marshall (1984) suggest that by using a Levenberg-Marquardt parameter δ such that $\mathbf{B} + \delta \cdot \operatorname{diag}(\mathbf{B})$ the solution becomes even more robust.

All three approaches require quite a lot of work to be done to find the $\boldsymbol{\theta}$-vector. Both CV as well as the method proposed by Jeong et al. (2006) and

[8] We therefore modified the algorithm that searches the new 'optimal' values of $\boldsymbol{\theta}$ in such a way that, when $|\mathbf{R}| = 0$ we do not use the true $\ln(|\mathbf{R}|)$ but instead we use $\lim_{|\mathbf{R}|\to 0}\ln(|\mathbf{R}|) = -\infty$. Although this ensures that the algorithm does not 'crash' it can cause the algorithm to be unsure in which direction to look for improvements at which point the algorithm is aborted in a neat way (i.e. the algorithm continues with the next operation).

Mardia and Marshall (1984) seem to require a lot of work (either by determining n Kriging models or by determining the derivatives of $\mathbf{R}$) whereas the MLE approach is quite straightforward. Furthermore Martin and Simpson (2005) conclude that the MLE model performs better when there are a large number of observations, something that is true in this research. It therefore seems best to use the MLE approach, and use a numerical optimiser to determine the values of $\boldsymbol{\theta}$ in each iteration. We thus choose for the MLE approach and use the numerical optimiser *fmincon* using the interior-point algorithm from the Matlab® Optimization Toolbox™(The Mathworks Inc., 2010). Main benefit of the interior-point algorithm is that it only requires a start value for the variables (in our case $\boldsymbol{\theta}$ and $\mathbf{p}$), whereas other algorithms also require the Jacobian and Hessian matrix (the vector of first-order derivatives and matrix with second-order derivatives). Furthermore it is interesting to note that Lophaven, Nielsen and Søndergaard (2002) developed a Kriging Toolbox for Matlab®, which contains the DACE approximation technique.

Another issue is the determination of the prediction error s^2, Den Hartog, Kleijnen and Siem (2006) suggest that the method as discussed in D. R. Jones et al. (1998) underestimates the error, because one uses the same data set to fit the model and predict $\hat{\sigma}^2$. One should therefore use bootstrapping (leave-one-out) in order to determine the Mean Squared Error (MSE) of the model, which can be used as a better predictor for $\hat{\sigma}^2$. However often it is more important that one realises that the error is underestimated than knowing the exact predicted error at a certain point. Especially since it requires a lot of computational effort, such an approach should only be used when there is a real need for good error predictors.

Kriging can of course be used to pre-evaluate solutions that are considered by the GA, however there is also a more intelligent approach which tries to determine the next point that should be evaluated using Expected Improvement (EI). D. R. Jones et al. (1998) were amongst the first to combine this method with DACE, although limiting themselves to single objective problems. Simply stated, the improvement I at a certain point is dependent on the distribution of $\hat{f}$ and the minimum value that has been observed so far f_{min}. Then the EI can be determined using:

$$E\left[I(\mathbf{x})\right] = \begin{cases} \left(f_{min} - \hat{f}(\mathbf{x})\right) \Phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s}\right) + s\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s}\right) & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases} \tag{4.47}$$

Note that $\Phi(\cdot)$ and $\phi(\cdot)$ represent the standard normal cumulative distribution function (cdf) and standard normal probability density function (pdf). Sasena, Papalambros and Goovaerts (2002) improved this approach by adding an attribute g which emphasises the uncertainty, thus creating a more global

search. In this case the EI can be determined using:

$$E\left[I^g(\mathbf{x})\right] = s^g \sum_{k=0}^{g} (-1)^k \left(\frac{g!}{k!(g-k)!}\right) \left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s}\right)^{g-k} T_k \tag{4.48}$$

Where:

$$T_k = -\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s}\right) \left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s}\right)^{k-1} + (k-1)T_{k-2} \tag{4.49}$$

$$T_0 = \Phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s}\right) \tag{4.50}$$

$$T_1 = -\phi\left(\frac{f_{min} - \hat{f}(\mathbf{x})}{s}\right) \tag{4.51}$$

Because high values of g will leave the algorithm to only search globally, Sasena et al. propose a cooling schedule in which g slowly decreases from 20 to 0, where in the latter case only solutions that have expected values below the current minimum value are considered. Emmerich et al. (2006) expand the method to a multiobjective case, where they focus on the Probability of Improvement (PoI) instead of the EI. However in this case it is quite difficult to determine 'improvement' and they therefore stick to calculating the integral of the pdf over the valid solution space, i.e.:

$$PoI(\mathbf{x}) = \int_{\mathbf{x} \in V} \phi_{\mathbf{x}}(\mathbf{y}) \, \mathrm{d}\mathbf{y} \tag{4.52}$$

$$V := \{\mathbf{y} | \mathbf{y} \text{ is non-dominated}\} \tag{4.53}$$

Although the idea of finding the point at which EI is maximised is a good one, it seems to be quite difficult to find such a point when dealing with a multiobjective problem. An interesting alteration of the methods discussed above would be to determine a combined expected improvement, i.e. determining the quadratic sum over the EIs of all objectives. The use of determining the value of EI, or in fact any method to determine an interesting point, can for instance be found in Kleijnen (2009) and Kleijnen, Van Beers and Van Nieuwenhuyse (2010) who propose the sequential build of the set of points that are used to determine the Kriging model. This means that at each iteration those points are added that contribute the most to the model (either by finding a good solution, or by evaluating areas that are unknown). Main drawback of creating a new Kriging model for each iteration is shown in the work by Gano, Renaud, Martin and Simpson (2006) who specifically use a low-fidelity model in order to save computational time that would otherwise be lost by fitting a Kriging model. Dellino, Lino, Meloni and Rizzo (2009) proposes the use of adaptive updating schedules, where the Kriging model is only updated when

the error between the prediction and the true exact value became too large. However in this case there is a risk, since it might very well be that the Kriging model rejects several good solutions before the algorithm found out that the Kriging model was wrong.

Although the approach was first developed for the mining industry, Kriging has now also become popular in other sciences such as meteorology and statistics (Cressie, 1990). The wide range of applications of Kriging has expanded to structural and mechanical engineering, where the use of Kriging models becomes more and more common practice (see e.g. El-Beltagy, Nair & Keane, 1999; Mehnen, Michelitsch, Lasarczyk & Bartz-Beielstein, 2007; Dellino et al., 2009). The power of Kriging lies, according to Simpson et al. (2001), with the fact that the method is extremely flexible and can easily be applied to problems with up to 50 parameters. Emmerich et al. (2006) find that Kriging (especially in multiobjective situations) forces algorithms to perform evaluations in less explored regions, thereby preventing the algorithms from being misled by poor predictions. J. Knowles (2006) concludes that a Kriging assisted GA generally outperforms NSGA-II. However there are also some drawbacks to Kriging, for instance R. Jin, Chen and Simpson (2001) finds that Kriging is very sensitive to the sample that is provided, more than for instance RBF. Simpson et al. (2001) find that the complexity of Kriging, it is not an easy method to grasp, makes it less suitable in some occasions.

Kriging however can be applied in a number of different ways. Already mentioned is the use of Kriging in determining the EI in order to find new, possibly optimal, points (see e.g. D. R. Jones et al., 1998; Sasena et al., 2002; Emmerich et al., 2006). Both Ratle (2001) and J. Knowles (2006) use Kriging in combination with a GA. They let the Kriging model evaluate the solutions that are provided by the GA and evaluate only the best solutions using the exact model. After a couple of generations the Kriging model is updated using the points that have been evaluated by the exact model in the past generations. Li et al. (2009) give even more power to the Kriging model, if the error (as estimated by the Kriging model) is small the value that is calculated by the Kriging model is used, only when the predicted error is large, the exact model is used to evaluate the solution.

The use of Kriging shows that the model is capable of modelling complex problems, however when the number of objectives, parameters or solutions becomes large, solving the Kriging model itself becomes a computational burden.

4.5 Quality Measures

In order to select the best approximation method, one needs a clear definition of ‘best’. However selecting the best method is not that easy since we should

weigh approximation accuracy and computational effort. Furthermore it is difficult to define approximation accuracy. We therefore limit ourselves to finding good approximation methods, without trying to find the best one.

In literature we find three (quite common) definitions of approximation accuracy (see for instance: Lim, Ong, Jin & Sendhoff, 2007). The first is probably the simplest definition and determines the average (absolute) error over all predictions. This Mean Absolute Error (MAE) can be determined using:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - \tilde{f}_i| \tag{4.54}$$

The second definition is probably the best-known method (Y. Jin, Hüsken & Sendhoff, 2003) and determines the average squared error, thereby giving more weight to predictions that are further away from the true value. Giving more weight to larger errors is in line with the common thought that minor errors do not really matter, whereas major errors are so much the worse. This Root Mean Square Error (RMSE) can be determined using:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(f_i - \tilde{f}_i\right)^2} \tag{4.55}$$

Note that both of these definitions focus solely on how well the true values are approximated. However Y. Jin et al. (2003) propose the use of methods that focus on predicting the behaviour of problems rather than solutions. Therefore Y. Jin et al. as well as Lim et al. (2007) suggest the use of the sample correlation coefficient r, which describes the relation between two data sets, in this case the exact values and the approximates values. The value of r is restricted to the range $[-1, 1]$, where -1 means that the predictions and true values take the opposite directions. Logically 1 means that the predictions and true values take the same direction and thus indicate that the behaviour of the predictions is similar to the behaviour of the true values. This sample correlation coefficient can be determined using:

$$r = \frac{n\sum_{i=1}^{n} f_i\tilde{f}_i - \sum_{i=1}^{n} f_i \sum_{i=1}^{n} \tilde{f}_i}{\sqrt{n\sum_{i=1}^{n} f_i^2 - (\sum_{i=1}^{n} f_i)^2}\,\sqrt{n\sum_{i=1}^{n} \tilde{f}_i^2 - (\sum_{i=1}^{n} \tilde{f}_i)^2}} \tag{4.56}$$

In fact Y. Jin et al. (2003) go even further and propose a method that goes back to determining false positives and false negatives (see Table 4.3 for the definition of false positives and false negatives). This method is, in contrast to the three previous quality measures, a multiobjective quality measure. It assumes that one is able to determine, using the approximated objective values, which of the children is going to be part of the next parent set (i.e. the parent

set that is the result of the non-dominated sorting algorithm) and as such are not dominated by other children or parents. If the expected set of dominating children is indeed similar to the true set of dominating children (to be determined using the exact objective values) the approximation method appears to provide objective values that lead to the right decisions (either accepting or rejecting children to be part of the set of non-dominated solutions).

		Approximation	
		dominated	non-dominated
Exact	dominated	OK	false negative
	non-dominated	false positive	OK

Table 4.3: Possible Outcomes Domination Quality Measure

Based on the concept introduced by Y. Jin et al. (2003) we developed our own method to determine how good an approach is. Generally speaking one can identify four different possible outcomes (see Table 4.3), which are dependent on two indicators. The first is whether or not a solution is part of the non-dominated set (when evaluated exactly), the second is whether the approximation method indicated the solution to be part of the non-dominated set.

From Table 4.3 it becomes clear that there are in fact three results, a number of solutions that is predicted correctly, a number of false positives and a number of false negatives. From an optimisation point of view, false negatives are worse than false positives, since in the former case one disposes good solutions, whereas in the latter case one only spends computational time in vain. We could therefore decide to sort the approximation methods based on false negatives, false positives and correct predictions, however it seems more convenient to create a single predictor for the quality of the approximation method. We therefore constructed a quality measure as described in equation 4.57, where lower values indicate better solutions. This quality measure is based on two concepts: a) we try to minimise the number of false predictions; and b) we try to get as close as possible to the correct number of rejected solutions.[9] Given that minimising the number of false predictions comes down to minimising the sum of false positives and false negatives and getting close to the correct number of rejected solutions is equal to minimising the difference between the false positives and false negatives[10] the quality measure can be defined as follows:

$$\vartheta = \text{false positives}^2 + \text{false negatives}^2 \tag{4.57}$$

[9]Note: these two concepts do not imply that false negatives are worse than false positives.

[10]Note: the latter part requires us to quadratise both parts of the quality measure, because a negative difference is as bad as a positive difference.

These four quality measures can clearly be categorised into two classes. The first category consists of quality measures that determine how well an approximation method estimates the value of a single objective function. Clearly RMSE and MAE are members of this category (both focus on the distance between the approximated and exact objective value), but r can also be used to determine how well a certain method approximated the behaviour of the data. In order to find the best approximation methods from an objective value point of view, we first selected for each objective function the best solution, i.e. the one with the lowest RMSE. This lowest RMSE was used as a normalisation constant and for each approximation method the summed deviation from the lowest RMSE was determined. Using RMSE_o as RMSE for objective o of a certain approximation method and RMSE_o^{min} as minimum RMSE of objective o over all the approximation methods, the summed deviation (RMSE_Σ) was determined using:

$$\text{RMSE}_\Sigma = \sum_o \left(\frac{\text{RMSE}_o}{\text{RMSE}_o^{min}} \right) \tag{4.58}$$

Besides RMSE and MAE also r is a good indicator to find approximation methods that closely mimic the behaviour of the objective functions. We therefore determine for each approximation method the average value of r over the three objectives, noted by $\widehat{r}$.

The second category consists of measures that do not focus on how well a single objective value is approximated, but focus on whether or not the right decision is made. In this specific case this means whether or not a solution would be correctly categorised as dominating (or non-dominating) when using the approximated values. Clearly ϑ is a measure that determines whether the right decisions are taken.

4.6 Methodology

From the explanation in sections 4.2 – 4.4 it becomes clear that each method has a number of variables (methods that can be used, parameter settings, initial values) that could potentially influence the results (and the quality) of the approximations. It therefore seems appropriate to first investigate the dependency of the approximation methods on the variable values. This is why for each of the variables (per approximation method) a set of possible values (or settings) is selected. The selected possible values per variable can be found in Tables 4.4 – 4.6. This results in a total of 148 approximation method variants.

In order to prevent the results to be biased due to a single unfortunate dataset, we decided to average the results over ten datasets. Each dataset consists of a part that is used by the approximation methods to learn and a part of which the objective values should be approximated. The former is called *learning data* and consists of the control values and the exact objective

type	maximum power
cubic	2
full	3

Table 4.4: Variable Values for Response Surface Method

type	distance function	c-value	γ-value
RBF	linear	Tanaka et al. (2007)	1
eRBF	cubic	Büche et al. (2005)	5
	thin plate spline	100	10
	multiquadratic	10	
	Gaussian	1	
	inverse multiquadratic	0.1	
		0.01	

Table 4.5: Variable Values for Radial Basis Functions

objective function	optimise p-vector	initial ϕ	
Martin and Simpson (2005)	yes	$1E^{-9}$	$5E^{-5}$
Mardia and Marshall (1984)	no	$1E^{-8}$	$1E^{-5}$
		$5E^{-7}$	$1E^{-4}$
		$1E^{-7}$	$1E^{-3}$
		$5E^{-6}$	$1E^{0}$
		$1E^{-6}$	$1E^{3}$

Table 4.6: Variable Values for Kriging/DACE

values, the latter consists solely of control values and is called *input data*. The corresponding exact objective values will be referred to as *exact output*, in contrast to the *approximated output*, which is the result of the approximation methods.

The performance of each of these approximation methods is measured using two different test networks. The first test network, that is described in section 3.1, was used in the research of Wismans et al. (2009, 2010). Furthermore we make use of a database of exact objective value evaluations (and their corresponding measure settings) that was the result of the same study. The values of the controls ($\mathbf{x}$) are based on the characteristic parameters as described in section 2.3. In order to reduce the risk of selecting a method that only performs well on this specific network, we also developed a second test network (with different properties), which is described in section 3.2. For this test network we generated 2000 random solutions which were evaluated

exactly and stored in a database.

From the databases we selected the first 100 solutions as learning data and the next 100 solutions were selected as input data. An approximated output was calculated using each approximation method and for each solution in the input data. Using the exact output of this solution, for each approximation method the quality measure values were calculated. The 200 solutions (100 input data, 100 output data) were removed from the databases in order to prevent overlapping data. After 10 iterations the results were averaged, approximation methods that resulted in invalid quality measure values (e.g. Not a Number[11] or infinity) are considered to be unsuitable, since they do not provide a (feasible) solution in every instance.

We first applied the approximation methods to Test Network I and drew conclusions about the behaviour of the different approximation methods on this test network. Next we employed the approximation methods to Test Network II and checked whether the previous conclusions still hold. Because the fourth quality measure (ϑ; equation 4.57) is a multiobjective quality measure, we also wanted to investigate how this quality measure behaves for different numbers of objective functions. We therefore applied this quality measure also to a situation in which we only consider travel time (TTT; equation 1.1) and climate (CO_2; equation 1.2) and to a situation in which a fourth objective function (road safety) was included.

Because of the categorisation that has been made in section 4.5 we can define two groups of 'best' solutions. The first group consists of the approximation methods that approximate the objective values best, the second group consists of approximation methods that are good in predicting whether or not a solution is part of the non-dominating set.

In order to find the best approximation methods from an objective value point of view, we use $RMSE_\Sigma$ (see equation 4.58) and select (and rank) the best 10 approximation methods per test network. The approximation methods that were amongst the ten best solutions of both test networks were selected as 'best' solutions and sorted based on the sum of the rank they received for each individual test network. Since we also determined that $\hat{r}$ is a good indicator for the quality of the predictions, the best 10 solutions based on $\hat{r}$ per test network were selected and ranked. The approximation methods that were amongst the ten best solutions of each test network were selected as 'best' solutions and sorted based on the sum of the rank they received for each test network.

The best approximation methods from a decision point of view should be selected based on ϑ for two to four objectives. Therefore the approximation methods are ranked based on ϑ for two, three and four objective functions

[11] Not a Number (NaN) is the result of undefined arithmetic operations such as subtracting infinities, multiplication of infinity by zero, division of zero by zero, division of infinity by infinity, etc.

for each test network. For each test network the ten best approximation methods are selected based on the sum of ranks for two three and four objective functions. The 'best' approximation methods from a decision point of view are those approximation methods that are in the top ten of each test network and are sorted based on the sum of the rank they received for each test network.

Furthermore it is interesting to take a closer look at the errors (s^2) that are estimated by the Kriging/DACE approach. As explained in section 4.4, Kriging/DACE is capable of estimating errors (s^2) for each objective value for each individual solution. This would offer (for instance) the possibility to select only those solutions for exact evaluation that have a high uncertainty (i.e. high expected error) according to Kriging/DACE. We therefore will investigate the error behaviour of the Kriging/DACE methods that score 'best' on approximating objective values.

What is important is that the errors that are provided by Kriging/DACE are reliable, in that sense that predicted errors that are larger should indeed correspond to solutions that have larger errors. We will therefore (for each objective) sort all the solutions based on the predicted error and plot the true error (i.e. the squared error which can be calculated using the approximation and the exact objective value) against it. In order to reduce the effect of incidental errors we calculate the average error over 50 solutions, resulting in 20 points of data. If the Kriging/DACE error is indeed a good predictor one would expect positive correlation. Using standard regression analysis we will analyse whether there is indeed a positive correlation between the errors predicted by Kriging/DACE and the true errors.

4.7 Results

Because we have examined in total 148 different variants of approximation methods (4 RSM, 24 RBF, 72 eRBF and 48 Kriging/DACE), publishing detailed information on all these variants would hardly contribute to a clear decision. We therefore decided to publish the results in a different way. We therefore start by discussing the results for each of the approximation methods on both test networks in a general way, i.e. we discuss remarkable results and try to discover trends as to which variants appear to provide consistently good results. Furthermore we give an indication of the calculation times that were necessary, since one of the purposes of this research is to accelerate the search for the set of Pareto optimal solutions. Again, main goal of these two subsections (*Test Network I* and *Test Network II*) is to provide a global overview of the results.

Furthermore we briefly investigated the sensitivity of the variants of approximation methods towards the number of objective functions that is used in the problem (see: *Comparing 2 – 4 Objectives*). Again this subsection merely

gives a general overview of the results and does not attempt to be an in depth research into the effects.

The subsection *Best Performing Variants* provides for both of the two categories of 'best solutions' (as defined in sections 4.5 and 4.6) the best scoring variants of the approximation methods. In order to provide a clear overview of the 'best solutions' the tables do not only include the five best scoring variants for a specific quality measure, but also include variants that performed best for other quality measures. Although this subsection does not draw a definitive conclusion about which approximation method variant should be selected, it does define a subselection from which the final variant should be picked.

Finally we will discuss the quality of the Kriging/DACE errors in *Error Estimations*. Main goal of this subsection is to determine which (if any at all) of the Kriging/DACE variants is capable of producing reliable error estimates, in the sense that the true errors and error estimates should be positively correlated. The conclusions drawn in this subsection are considered when the definitive conclusion about the approximation methods is drawn in section 4.8.

Test Network I

For RSM we see that the calculation time varies between less than 1 second for a cubic squared model to nearly 12 minutes for a full cubed model. The calculation time of this latter approach might become an issue with realistic networks with many DTM measures. Furthermore we see that a cubic squared model outperforms all other RSM approximation methods at each quality measure. This is for instance represented by the correlation coefficient (r) which is on average 0.66 for RSM cubic squared, whereas the r for RSM full cubed is on average only 0.06.[12]

For RBF we see calculation times are consistently less than 1 second, which makes it a quite fast approach. The first conclusion that can be drawn is that a Gaussian distance function gives results that are so far off that the approximation method never rejects a proposal. Only for very small values of c (either using the method of Tanaka et al. (2007) or a constant value of 0.01) the approximation method gives valid results (i.e. real numbers), although they still are not very good (r values around zero and sometimes even negative).

For an inverse multiquadratic distance function exactly the opposite is true, since large values of c improve the results. The optimal value of c (from a RMSE point of view) is somewhere between the value of Büche et al. (2005), which is about 20, and 100. The constant c-value of 100 does however give better results on the ϑ quality measure and the correlation coefficient r (which is on average 0.62).

[12]For some objective functions the value of r is even negative, indicating that the model is completely unable to predict the results.

For a multiquadratic distance function again the best results can be obtained with large c-values although the optimal values vary between 10 and 100. If we therefore consider the ϑ quality measure it seems that the constant value of 100 does provide the best results.

Something remarkable is that the very simple linear distance function gives reasonable results, i.e. results that are better than most other RBF approximation methods. However the best multiquadratic and inverse multiquadratic distance functions do outperform the linear distance function in respect to RMSE, ϑ as well as r.

The calculations for the extended RBF (eRBF) take about 75 seconds with very little variation between the different methods. The most remarkable conclusion is that there is (nearly) no difference between the results of the approximation methods that use linear, multiquadratic, Gaussian or inverse multiquadratic distance functions (the latter only with c-values larger than 0.1) and that this conclusion holds for all studied values of γ.

Looking more closely at the results for the inverse multiquadratic distance function, we found that smaller values of γ improved the results (something that holds for all distance functions) and that smaller values of c also gave much better approximations.

However of all eRBF approximation methods the one based on a cubic distance function gave the best results when considering RMSE (although cubic is outperformed by inverse multiquadratic when considering the newly added c-values) and r (which is on average 0.67). For some reason the cubic distance function is unable to give a good approximation for the noise objective when $\gamma = 1$, for all other γ-values cubic seems to do perfectly fine.

The calculation times for DACE vary between less than 1 second and 6 minutes, unfortunately both the very fast as well as the very slow approximation methods do not provide reliable results; they result in either a constant objective value or provide a RMSE that is infinity.

The results show clearly that not optimising the value of p gives better or at least equal results when compared to the same approximation methods that do optimise p. Furthermore we see that there is very little difference between using the method of Martin and Simpson (2005) or Mardia and Marshall (1984) when determining the maximum likelihood, which could be expected, because the difference exists solely of a constant value.

Unfortunately many of the start values of ϕ result in invalid approximated output for one or more objectives. However ϕ start values of $1E^{-6}$ and $1E^{-7}$ give (when not optimising p) good approximations (for some objective functions even the best), with r values over 0.60. However the ϕ start value of $1E^{-8}$ gives the best results for ϑ and is much faster than the other two start values. This better result however comes at a cost, where a start value of $1E^{-6}$ rejects 57% of the solutions, a start value of $1E^{-8}$ only rejects

48%. This means that the latter would require nearly 21% more exact DTA evaluations.

Test Network II

The calculation of the approximated output for RSM took less than 1 second for the cubic squared and nearly 30 minutes for full cubed. The cubic squared still gives the best approximations based on RMSE and r (r values are on average 0.85), however a cubic cubed method seems to give slightly better results for the ϑ with a r value of 0.83.

Calculation of an RBF approximation still takes less than 1 second, which indicates that the method is not extremely nervous to the number of characteristic distances. The conclusions that were drawn about the Gaussian distance function still hold. However for the inverse multiquadratic distance function now a constant c-value of 100 provides consistently the best results, both for RMSE as well as ϑ with r values that are on average over 0.70. For the multiquadratic distance function a constant c-value of 100 provides the best results for both RMSE as well as ϑ except for the RMSE of CO_2 were the c-value provided by Büche et al. (2005) gives a better results. Again a linear distance function gives reasonable results, although it has on average a r value of only 0.18 and is being outperformed by the better multiquadratic and inverse multiquadratic distance functions.

The calculations for an eRBF approximation took about 104 seconds for Test Network II, with very little variation between the different methods.

Again we see that the linear, multiquadratic, Gaussian and inverse multiquadratic (for c-values larger than 0.1) give similar results with average r values of 0.69. Again do the results improve when the value of γ decreases. For the higher values of γ the cubic distance function outperforms the inverse multiquadratic distance function with c-values based on Tanaka et al. (2007), however for $\gamma = 1$ the result is exactly the opposite. Again this suggested that the results might be improved if we use lower c-values.

For Test Network II the calculation times for DACE range from less than a second to just over 5 minutes. However since these extreme values usually occur when the approximation method is unable to give valid results, these calculation times cannot be compared directly to the ones of Test Network I.

Again the approximation methods that do not optimise p outperform the methods that do optimise p and again there is very little difference between the likelihood function of Martin and Simpson (2005) or Mardia and Marshall (1984). However in this case a ϕ start value of $1E^{-7}$ clearly is best, both from the perspective of RMSE as well as ϑ and has a correlation coefficient r of 0.89.

Comparing 2 – 4 Objectives

When comparing the results for two to four objectives for both test networks, we see that more or less the same approximation methods perform best. However it is quite remarkable to notice that the eRBF approximation methods perform sometimes quite well (at least some of the eRBF approximation methods), but that the same methods do sometimes perform worse than average. It is however reassuring to see that there are a couple of approximation methods that are always in the top ten in both test networks.

The values of the quality measures sometimes vary strongly when changing the number of objective functions. There is however not a clear trend (either increasing or decreasing), which indicates that the quality is not so much dependent on the number of objective functions, but more on the difficulty of the individual objective functions.

Best Performing Variants

If we focus on those approximation methods that are good at approximating the objective values, we see that especially Kriging/DACE methods are capable of good results. Whether it is on Test Network I or II, Kriging/DACE methods with ϕ start values of $1E^{-7}$ or $1E^{-8}$ perform very good, independent of the likelihood function (Mardia and Marshall (1984) or Martin and Simpson (2005)) that is being used. Table 4.7 gives the RMSE and MAE for each objective for each test network.[13] Also the RMSE_Σ (over the situations 2 – 4 objective functions) is given.

When looking at the approximation methods that score best on the correlation coefficient (r) we see that not only Kriging/DACE (and a rare RBF function) are capable of approximating the behaviour, also the simple RSM approach (cubic squared) as well as two (relatively simple) eRBF functions appear to be amongst the best. Table 4.8 gives the r values per objective and the average $\hat{r}$ over all the objectives.

When looking at the approximation methods that are good at predicting the final decision, we see that not only Kriging/DACE performs very well but also RSM is able to produce good (and in some occasions the best) results. This is especially interesting, because RSM is the approximation method that is easiest to understand and very fast. It is quite remarkable that there is not a single RBF approach that is able to provide consistently good results. Table 4.9 gives the value of ϑ for 2 – 4 objective functions.

Furthermore we provide an overview of the calculation times of each of the nine approximation method variants that are considered to be the 'best' based

[13] The approximation variants above the middle line are the best performing variants for this quality measure, the variants under the middle line are approximation variants that perform best on other quality measures and are added for completeness.

	TTT		CO_2		Noise		$RMSE_\Sigma$
	RMSE	MAE	RMSE	MAE	RMSE	MAE	
DACE, Mardia, no, $\phi = 1E^{-7}$	188.17	145.10	1401665	1092351	0.08	0.06	3.04
DACE, Martin, no, $\phi = 1E^{-7}$	197.79	152.94	1386682	1085045	0.08	0.06	3.05
RBF, inverse multiquadratic, $c = 100$	228.39	178.68	1634048	1290823	0.09	0.07	3.51
DACE, Mardia, no, $\phi = 1E^{-8}$	227.72	176.08	1865659	1489833	0.09	0.07	3.75
DACE, Martin, no, $\phi = 1E^{-8}$	227.60	176.84	1865659	1489833	0.09	0.08	3.75
eRBF, $\gamma = 5$, cubic	239.66	187.06	1728236	1379024	0.19	0.14	4.92
RSM, cubic, squared	234.10	174.72	1819115	1073642	0.23	0.09	5.44
eRBF, $\gamma = 1$, cubic	206.32	160.13	1780328	1405631	0.24	0.18	5.47
RSM, cubic, cubed	410.04	224.68	3155573	1340989	0.71	0.17	13.39

(a) Test Network I

	TTT		CO_2		Noise		$RMSE_\Sigma$
	RMSE	MAE	RMSE	MAE	RMSE	MAE	
DACE, Mardia, no, $\phi = 1E^{-7}$	2452840	1949302	2105537	1623153	0.10	0.08	3.13
DACE, Martin, no, $\phi = 1E^{-7}$	2455398	1952094	2125148	1635014	0.10	0.08	3.13
DACE, Mardia, no, $\phi = 1E^{-8}$	2694163	2127465	2274658	1761309	0.11	0.09	3.41
DACE, Martin, no, $\phi = 1E^{-8}$	2682653	2122464	2293354	1779051	0.12	0.09	3.51
RBF, inverse multiquadratic, $c = 100$	2811575	2249812	3839874	3039226	0.19	0.15	4.91
RSM, cubic, squared	2182751	1711620	2745277	2160307	0.13	0.10	3.51
RSM, cubic, cubed	2428597	1920298	2975960	2381833	0.14	0.11	3.83
eRBF, $\gamma = 5$, cubic	2695499	2153628	3745959	2993796	0.17	0.14	4.70
eRBF, $\gamma = 1$, cubic	2980571	2375309	3621462	2870824	0.19	0.15	4.87

(b) Test Network II

Table 4.7: Best Methods for Approximating Objective Values ($RMSE_\Sigma$)

	TTT	CO_2	Noise	$\widehat{r}$
eRBF, $\gamma = 1$, cubic	0.74	0.82	0.44	0.67
RSM, cubic, squared	0.68	0.84	0.47	0.66
DACE, Mardia, no, $\phi = 1E^{-7}$	0.75	0.81	0.43	0.66
DACE, Martin, no, $\phi = 1E^{-7}$	0.72	0.81	0.43	0.65
eRBF, $\gamma = 5$, cubic	0.69	0.83	0.37	0.63
RBF, inverse multiquadratic, $c = 100$	0.66	0.81	0.39	0.62
DACE, Martin, no, $\phi = 1E^{-8}$	0.69	0.77	0.38	0.62
DACE, Mardia, no, $\phi = 1E^{-8}$	0.69	0.77	0.36	0.61
RSM, cubic, cubed	0.55	0.76	0.31	0.54

(a) Test Network I

	TTT	CO_2	Noise	$\widehat{r}$
DACE, Mardia, no, $\phi = 1E^{-7}$	0.90	0.82	0.94	0.89
DACE, Martin, no, $\phi = 1E^{-7}$	0.90	0.81	0.94	0.89
RSM, cubic, squared	0.92	0.71	0.92	0.85
eRBF, $\gamma = 5$, cubic	0.89	0.58	0.86	0.77
eRBF, $\gamma = 1$, cubic	0.86	0.58	0.84	0.76
DACE, Martin, no, $\phi = 1E^{-8}$	0.89	0.80	0.93	0.87
DACE, Mardia, no, $\phi = 1E^{-8}$	0.88	0.80	0.94	0.87
RSM, cubic, cubed	0.90	0.68	0.91	0.83
RBF, inverse multiquadratic, $c = 100$	0.87	0.53	0.83	0.74

(b) Test Network II

Table 4.8: Best Methods for Approximating Objective Values ($\hat{r}$).

	ϑ			$\sum \vartheta$
	2 Obj. Fun.	3 Obj. Fun.	4 Obj. Fun.	
RSM, cubic, squared	200	648	841	1690
DACE, Martin, no, $\phi = 1E^{-7}$	216	769	813	1798
DACE, Mardia, no, $\phi = 1E^{-7}$	214	803	823	1840
DACE, Martin, no, $\phi = 1E^{-8}$	276	719	853	1848
RSM, cubic, cubed	206	681	963	1851
DACE, Mardia, no, $\phi = 1E^{-8}$	279	724	929	1932
RBF, inverse multiquadratic, $c = 100$	319	874	1046	2239
eRBF, $\gamma = 1$, cubic	203	1019	1063	2285
eRBF, $\gamma = 5$, cubic	247	925	1139	2311

(a) Test Network I

	ϑ			$\sum \vartheta$
	2 Obj. Fun.	3 Obj. Fun.	4 Obj. Fun.	
DACE, Martin, no, $\phi = 1E^{-7}$	504	281	615	1400
DACE, Mardia, no, $\phi = 1E^{-8}$	415	335	659	1409
DACE, Mardia, no, $\phi = 1E^{-7}$	524	287	638	1449
DACE, Martin, no, $\phi = 1E^{-8}$	454	346	650	1450
RSM, cubic, squared	552	450	623	1625
RSM, cubic, cubed	644	408	741	1794
eRBF, $\gamma = 1$, cubic	637	463	668	1768
RBF, inverse multiquadratic, $c = 100$	606	510	683	1799
eRBF, $\gamma = 5$, cubic	634	484	805	1923

(b) Test Network II

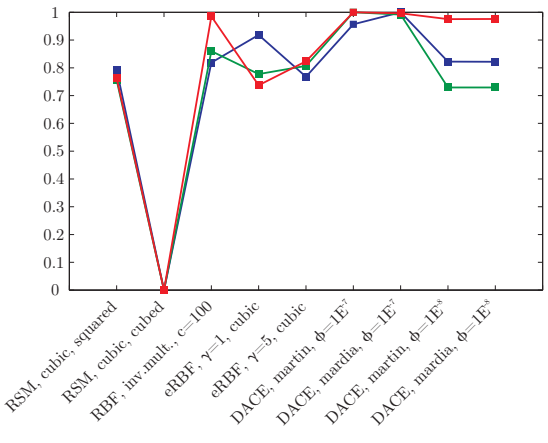
Table 4.9: Best Methods for Prediction Decisions (ϑ)

Test Network I calculation time (s)		Test Network II calculation time (s)
< 1	RSM, cubic, squared	< 1
< 1	RSM, cubic, cubed	< 1
< 1	RBF, inverse multiquadratic, $c = 100$	< 1
121	eRBF, $\gamma = 1$, cubic	104
124	eRBF, $\gamma = 5$, cubic	104
512	DACE, Martin, no, $\phi = 1E^{-7}$	169
541	DACE, Mardia, no, $\phi = 1E^{-7}$	173
169	DACE, Martin, no, $\phi = 1E^{-8}$	133
170	DACE, Mardia, no, $\phi = 1E^{-8}$	137

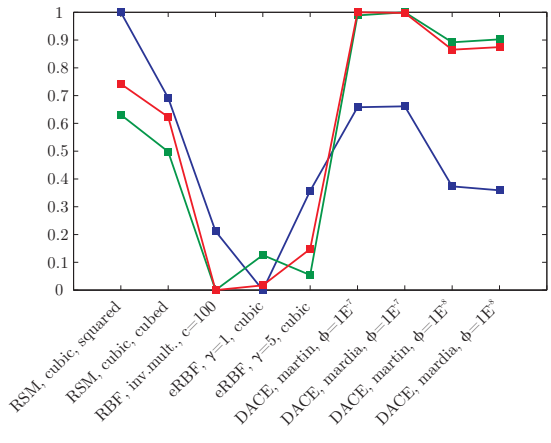
Table 4.10: Calculation times for Approximation Method Variants

on the previous analysis. Although it is not one of the stringent criteria, it gives an indication of the computational effort of each of the variants. The computations for Test Network I and Test Network II were not performed on the same hardware and can therefore not be compared. It is however possible to compare the results within a test network.

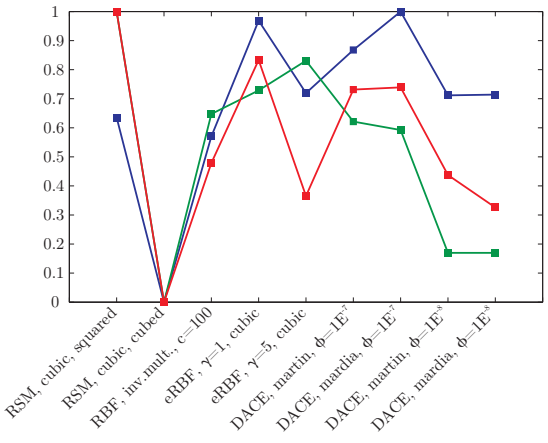
For convenience Figure 4.4 provides an overview of all nine methods that are amongst the 'best' for one or more criteria. To ensure a clear overview, we gave the best performing approach (of the nine approaches) score 1, the worst performing approach score 0 and scaled all the other approximation variants



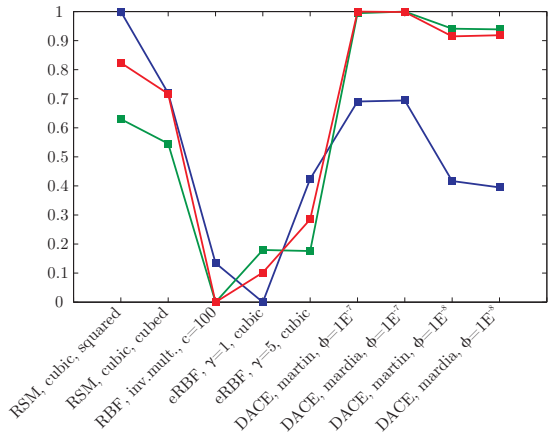
(a) RMSE for Test Network I



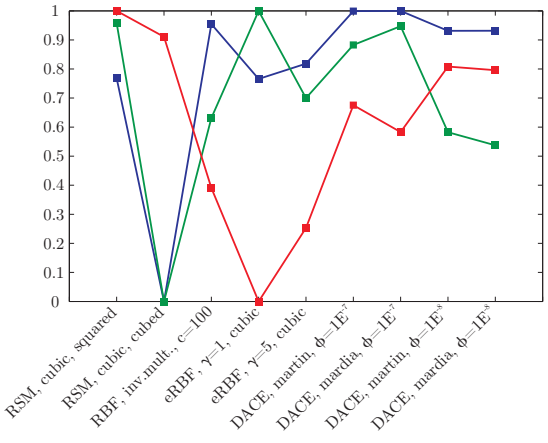
(b) RMSE for Test Network II



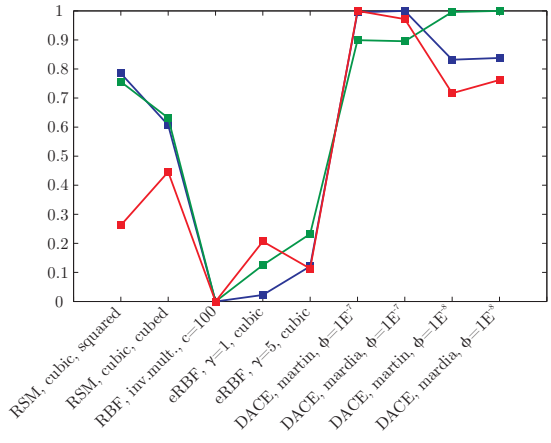
(c) r for Test Network I



(d) r for Test Network II

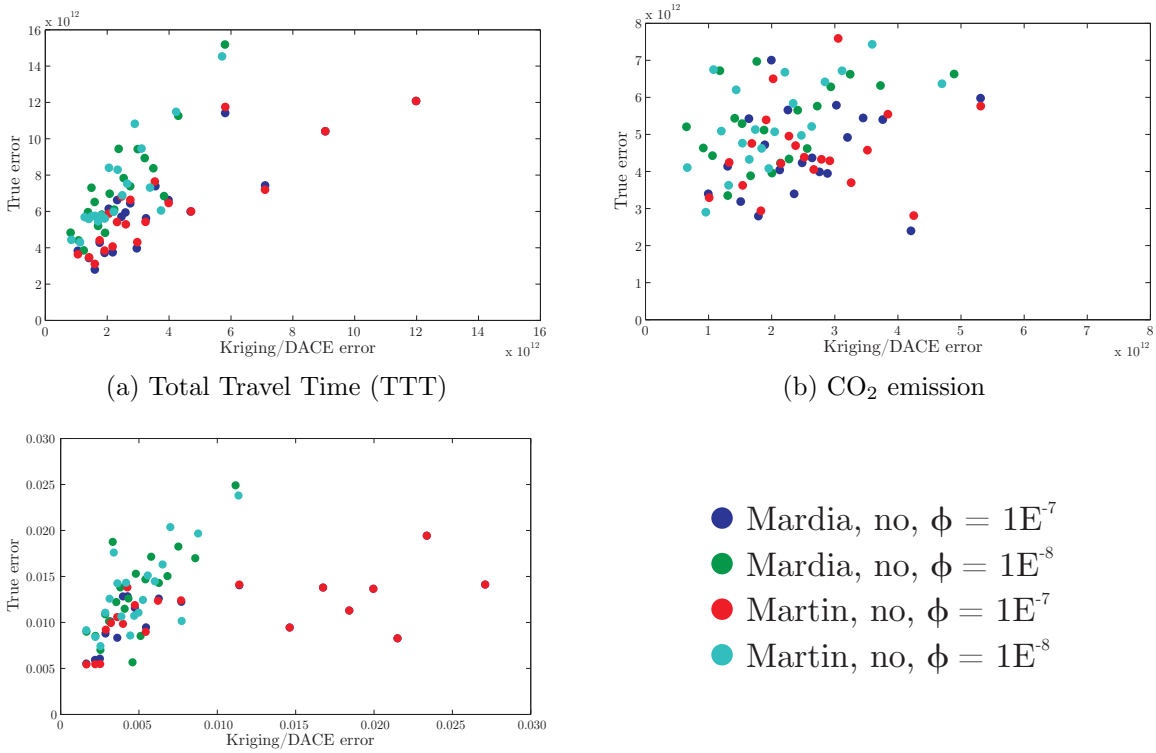


(e) RMSE_Σ, $\hat{r}$ and $\sum \vartheta$ for Test Network I



(f) RMSE_Σ, $\hat{r}$ and $\sum \vartheta$ for Test Network II

Figure 4.4: Overview of Best Scoring Approximation Variants



(a) Total Travel Time (TTT)

(b) CO_2 emission

(c) Noise

Figure 4.5: True vs Kriging/DACE Errors

linearly in the range (0,1).

It is interesting to find that both Kriging/DACE approaches with ϕ start value $1E^{-7}$ seem to provide results that score consistently over 0.5, whereas many of the other approaches score far below this value. Furthermore we find that the most simple approach (RSM with cubic squared terms) performs quite well, in fact it only scores below the 0.5 in one occasion (for the $\sum \vartheta$ criterion on Test Network II). What is also interesting to see is that solutions that perform very poorly on RMSE and r (e.g. RSM cubic cubed for Test Network I) can score very good on $\sum \vartheta$ (see Figures 4.4a, 4.4c and 4.4e).

Error Estimations

The Kriging/DACE approximation methods that should be investigated are the methods that use the likelihood function of either Mardia and Marshall (1984) or Martin and Simpson (2005), with a start value of ϕ of $1E^{-7}$ or $1E^{-8}$ which do not optimise the values of p (i.e. the approximation methods that showed to have consistently the lowest RMSE). The graphical result of the analysis that is described in section 4.6 can be found in Figure 4.5.

	TTT		CO_2		Noise	
	a	R^2	a	R^2	a	R^2
Mardia, no, $\phi = 1E^{-7}$	1.36	0.19	1.55	0.00	0.76	0.00
Mardia, no, $\phi = 1E^{-8}$	2.79	0.57	2.12	0.00	2.50	0.26
Martin, no, $\phi = 1E^{-7}$	1.36	0.18	1.55	0.00	0.76	0.00
Martin, no, $\phi = 1E^{-8}$	2.81	0.51	2.25	0.00	2.45	0.11

Table 4.11: Values of a and R^2 for Kriging/DACE methods

	TTT			CO_2			Noise		
	b $(\times 10^{12})$	a	R^2	b $(\times 10^{12})$	a	R^2	b $(\times 10^{-3})$	a	R^2
Mardia, no, $\phi = 1E^{-7}$	3.19	0.80	0.75	3.82	0.27	0.06	8.59	0.24	0.34
Mardia, no, $\phi = 1E^{-8}$	2.69	1.91	0.78	4.26	0.48	0.21	5.94	1.50	0.57
Martin, no, $\phi = 1E^{-7}$	3.21	0.80	0.74	3.96	0.24	0.04	8.50	0.25	0.34
Martin, no, $\phi = 1E^{-8}$	2.75	1.88	0.74	3.78	0.74	0.36	6.67	1.35	0.55

Table 4.12: Values of a, b and R^2 for Kriging/DACE methods

From Figure 4.5 it seems that Kriging/DACE is able to produce quite reliable errors, in fact for some objectives (TTT, noise) it seems that Kriging/DACE produces follow more a less a straight line through the origin. Note that the angle of the 'straight line' is different for the different approximation methods. In fact, it seems that the angle (and as such the results of the approximation methods) is strongly dependent on the value of ϕ, since the results for both the method using the likelihood function of Mardia and Marshall (1984) and the method using the likelihood function of Martin and Simpson (2005) is more or less equal.

If we investigate these results more closely, by trying to find a straight line through the points (a function of the form $y = ax$) we find the values for a as shown in Table 4.11. This table also gives the coefficient of determination, R^2, which is a common statistical test that gives the fraction of the value of y that can be explained by x. Note that the values of R^2 are sometimes zero, which is an indication that the model used ($y = ax$) is unsuitable for this problem. Furthermore it is interesting to see that the predicted error is generally speaking smaller than the true error ($a > 1$).

Because the $y = ax$ model is unable to give good estimates for CO_2 and noise, we decided to also check another model ($y = ax + b$) which would probably suite the results better. The results of fitting this model can be found in Table 4.12. Although now all of the values of R^2 are valid, some of them still are not quite good. Generally speaking it appears that start values of $\phi = 1E^{-8}$ are better able to approximate the errors than start values of $1E^{-7}$. That being said, it appears that $\phi = 1E{-}8$ consistently underestimates the errors for TTT and noise ($a > 1$) whereas all other errors are overestimated ($a < 1$). At the same time the likelihood function by Mardia and Marshall (1984) seems to outperform the one by Martin and Simpson (2005) although the differences are quite small and for CO_2 the results are exactly the opposite.

4.8 Conclusions

This chapter started by investigating three different approximation techniques, RSM, RBF and Kriging/DACE. At the end of this chapter we can conclude, based on the selected quality measures and the results from the two test networks that there are few approximation techniques that truly look promising.

It is difficult to state that one approximation variant is 'better' than another variant, simply because the research we did is too limited for such a firm verdict. In the introduction of this chapter we stated that we would select two approximation variants to be used in the remainder of this research. A clear candidate would be the Kriging/DACE approximation method with the objective function of Mardia and Marshall (1984) and $\phi = 1E^{-7}$. It has provided the best results for RMSE (see Figures 4.4a and 4.4b) and proved to score above average on r (see Figures 4.4c and 4.4d). Unfortunately the results for the other criterion (making the right decision) were less convincing since it scored amongst the best for Test Network I, but more or less failed for Test Network II (see Figures 4.4e and 4.4f). One of the major benefits of selecting this approximation variant (or any Kriging/DACE method for that matter) is that it allows us to use in the remainder of this thesis techniques that require error estimates.

Another very interesting (and promising) approximation variant is the RSM approach which uses squared cubic terms. It proved to be consistently good (although not the best) technique for RMSE and r (see Figure 4.4). Unfortunately the results for making the right decisions are (as with the Kriging/DACE approach) inconsistent. It proved to be the best for Test Network I but performed quite poorly on Test Network II (see Figures 4.4e and 4.4f). Another benefit of this approach is that it is really fast and easy to understand, something that is highly appreciated when the approximation method has to be explained to principals.

It is therefore that we decide to continue with the Kriging/DACE approximation technique with the objective function of Mardia and Marshall (1984), and $\phi = 1E^{-7}$ and the RSM approach with squared cubic terms. In the next chapter we will discuss and select different ways in which these approximation methods can be integrated with Genetic Algorithms. In chapter 6 we will then integrate the results of this and the next chapter into two different Approximation Method Assisted NSGA-II algorithms.

Chapter 5

Metamodel Assisted Evolutionary Algorithms

The world the algorithm makes possible is retrograde in its nature to the world of mathematical physics.
Its fundamental theoretical objects are symbols*, and not muons, gluons, quarks, or space and time fused into a pilot knot.*
Algorithms are human artifacts. They belong to the world of memory and meaning, desire and design.

David Berlinski (1942 –)

In this chapter we will perform a literature search (section 5.1) into how metamodels have been incorporated in EAs in order to accelerate the algorithm or improve the results (these new algorithms are called Metamodel Assisted Evolutionary Algorithms; MAEAs). The results of this literature search can be used as a guideline when we have to decide how and where we would like to use metamodels in our GA.

5.1 Literature Overview

Although literature about combining metamodels and GAs is readily on hand, integrating metamodels and MOGAs is a much less explored area. Furthermore, most authors refrain from explaining how they integrated the two algorithms. In fact, although literature provides a broad range of possibilities that could be applied, literature that clearly explains how metamodels are incorporated in MOGAs (or MOEAs) is quite rare.

Karakasis and Giannakoglou (2006) describe probably the most familiar and intuitive approach, where a metamodel is used as an Inexact Pre Evaluation

(IPE). This means that each generation of N new solutions (provided by the GA) is first evaluated using this metamodel and only the best $x\%$ will be evaluated with the expensive model. In their situation they define best by applying the fitness evaluation of the GA on the estimated values for each solution, although it is also possible to find other definitions of best (e.g. selecting only those population members that are expected to be non-dominating). Because of the intuitivity and simplicity of the approach it has been quite commonly used in the past decades (see e.g. El-Beltagy et al., 1999; Karakasis et al., 2007; Praveen & Duvigneau, 2009).

Emmerich et al. (2006) use a similar approach, although they define 'best' in a different way. In their opinion it is not so much the quality of the solution with respect to reproduction, but the improvement a solution could make to the metamodel that is relevant. They therefore make use of the Probability of Improvement (PoI), that has been mentioned before (see section 4.4), to determine which solutions should be evaluated with the exact model. The work by Keane (2006) provides a very good description of how this approach could be used in a multiobjective environment. D. R. Jones et al. (1998) as well as Sasena et al. (2002) describe the PoI too, but in their opinion it is not so much the probability of improving a solution but the Expected Improvement (EI, mentioned earlier in section 4.4) that is of interest when deciding which solutions should be evaluated, thereby incorporating also the magnitude of the improvement

A different approach is followed by Yang, Yeun and Ruy (2002) who make maximum use of the speed of the metamodel. They initialise the metamodel using a set of exactly evaluated sampling points and use a MOGA and the metamodel to determine an expected Pareto optimal set. From this set they select solutions that should be evaluated exactly based on the maximin distance design criteria, i.e. they select those solutions which are farthermost from all other solutions. After exactly evaluating those solutions, they update the metamodel and estimate a new expected Pareto optimal set. If there is very little difference between the two predicted Pareto optimal sets (with or without the selected points) the algorithm ends, otherwise a new MOGA is started to predict a new Pareto optimal set. The main difference with the approach of Karakasis and Giannakoglou (2006) and Emmerich et al. (2006) is that they decide each iteration of MOGA which solutions should be evaluated with the exact model, whereas Yang et al. wait until the entire MOGA, based on the metamodel, has finished. There is something to say for letting MOGA run a few generations before selecting the solutions one would like to evaluate, since the solutions can improve significantly after a few runs. However it requires a metamodel that is without major flaws, otherwise MOGA will search in the wrong direction, whereas these flaws might have been corrected already when the metamodel was updated with each new generation.

Li et al. (2009) combine metamodel assisted fitness evaluations and the Design of Experiments (DoE) approach. This DoE approach selects new solu-

tions that should be evaluated exactly, based on the knowledge they would add to the understanding of the problem. By using the Kriging metamodel they are able to predict the error for each solution that is suggested by the MOGA, and they only perform an exact evaluation if the error at that specific point is large. For all other solutions they accept the estimated values of the metamodel. In comparison to Emmerich et al. (2006) and Karakasis and Giannakoglou (2006) they rely heavier on the metamodel, since they do accept the objective values that have been found by the metamodel in their Pareto optimal set. Besides evaluating all solutions with the exact model, Li et al. also identify a number of solutions that could be considered interesting. Therefore they perform a MOGA search on the metamodel, in order to identify those points that maximise entropy of the solution. They show that the additional point that maximises entropy can be found by maximising $|\mathbf{R}|$ under some constraints that ensure that the new solution does indeed dominate existing solutions. Although the approach seems to make sense, it does create a new complex problem that has to be solved each iteration using another MOGA. The question is whether this increase in complexity does pay off, or whether simpler approaches are also able to improve the results of the algorithm.

Finally Georgopoulou and Giannakoglou (2009) create an even more complex model, by combining the selection of specific approximation models, IPE and local search into one algorithm. For each new solution that is provided by MOGA they determine the closest known points and build an RBF model based on these points in order to estimate the objective values for each solution. They separate the outliers and select, from the remaining solutions, a couple of solutions worth an extra local search. Finally a number of solutions from each of the sets (outliers, 'ordinary solutions' and solutions derived from local search) is evaluated using the exact model. Although Georgopoulou and Giannakoglou conclude that their algorithm outperforms others, the approach seems to be very complex and requires quite a lot of computational effort. The use of local search is also suggested by Khoo and Chen (2001) and Noman and Iba (2008), unfortunately local search is of limited use when trying to find a Pareto front, because there is not a single 'best' solution that can be found by local search.

In conclusion we can say that literature provides a couple (although not too many) possibilities to accelerate MOGAs with metamodels. The algorithms range from quite straightforward to very complex, and whereas the former can be applied in virtually every situation, the latter requires problems where it is possible to apply other search algorithms to improve intermediate results. It seems however that this list is not exhaustive and that alternative approaches can be developed when considering the current NSGA-II algorithm.

5.2 Assisting NSGA-II

In this section we will investigate how and where we can accelerate NSGA-II, based on the literature review in the previous section. In NSGA-II there is in fact only one location where it makes sense to incorporate 'intelligence' and that is just before exactly evaluating the 'children' (solutions). If one would incorporate intelligence before this point (i.e. when generating new solutions) one would 'destroy' the 'heart' of the GA (which is finding new solutions based on genetic mutation of two 'parents'). Incorporating intelligence after this point seems useless because then the time consuming exact evaluations have already been performed.

If we only consider the approaches (or closely related variants) from section 5.1 we can categorise the ways in which we can incorporate intelligence using two criteria. In order to do so, we left out the approach by Yang et al. (2002), which uses only approximated objective values in the GA. The main reason for leaving this approach out is that the problems that are under consideration are very complex and we do require a regular update using exact evaluations. Similarly the approach by Georgopoulou and Giannakoglou (2009) is left out, because their approach is too complex. Especially the part in which a local search is performed is not feasible with our problems, since this local search approach is only applicable to single objective problems.

The first criterion then focusses on whether or not the approximated objective values are used in the GA. In some cases the GA continues assuming that the approximated objective values are indeed the true objective values. In that case solutions that have not been evaluated exactly can be used as parents in a future iteration. The other possibility is to only continue using exactly evaluated solutions, thus ensuring that the solutions that are used in generating new offspring are 'true'.

The second criterion is based on the data that is needed to obtain the intelligence. Some approaches only require the use of the expected value of the objective values others also require an estimate of the error. Clearly approaches that require estimates of the error severely limit the approximation methods that can be used, only Kriging/DACE is able to provide estimates of the error.

Table 5.1 shows how the approaches from section 5.1 would be categorised. It is interesting to see that most approaches require estimates of the error, whilst in chapter 4 we showed that there is only one approximation technique (Kriging/DACE) that is able to provide these estimates. Furthermore it is interesting that there is only one approach that has that much confidence in the approximation methods that it uses the values provided by these approximations in future iterations in the GA. In the next paragraphs we will discuss what the benefits and drawbacks of each of these approaches are.

		Required Estimates	
		both objective value and error	objective value only
Use approximations	yes	Design of Experiments (DoE)	
	no	Probability of Improvement (PoI) Expected Improvement (EI)	Inexact Pre Evalution (IPE)

Table 5.1: Categorisation of MAEA approaches

The Design of Experiments (DoE) approach can be considered a useful one, since it adds information to parts of the solution space where little is known about the objective value, i.e. the predicted error is quite large. DoE thereby tries to get a set of solutions that evenly spreads the knowledge over the solution space. However NSGA-II will immediately remove all knowledge that is not on the Pareto front, thereby also removing solutions that did provide additional knowledge to the 'problem'. This could be prevented when using two different sets of solutions, one to generate new children similar to the one used now (named 'parents') and one that is used to generate the approximation model (named 'fitdata'). The main difficulty with this approach is how to update the fitdata in such a way that the spread of solutions is maximised, whilst focussing on data that is around the Pareto front (parents). During a brief brainstorm two possible solutions for this problem have been found, the first is to change NSGA-II by adapting the approach of SPEA2+ (Zitzler, 2001; M. Kim et al., 2004) which does indeed contain two 'archives' and selects a specific archive as parentset based on the size of the Pareto front.[1] The second solution is by expanding the existing set of 'parents' by including for each child that is evaluated the solution from the database of previously evaluated solutions that is closest in solution space. In that way we increase the knowledge in areas around solutions that have to be evaluated (children). This however affects the time that is needed to create an approximation model, because the calculation times of all approximation methods discussed in chapter 4 are (more or less) squared proportional to the size of the fitdata.

The main benefit of the Inexact Pre Evaluation (IPE) approach is that it is intuitive, the idea of only evaluating solutions that one expects to be on the Pareto front is clear and intelligible. The main drawback however, is that if a solution is only just behind the Pareto front it is discarded, even if there is a lot of uncertainty about the exact objective values of this solution. In that sense it is a very rough and simplistic approach. This however also means

[1] One archive holds the best solutions sorted first on dominance and second on fitness sharing in the objective space, whereas the other archive uses dominance as primary and fitness sharing in the solution space as secondary sorting criterion.

Algorithm 5.1 Inexact Pre Evaluation (IPE)

1. Initialisation – 6. Variation

see the generic NSGA-II algorithm on page 13

7. IPE - Function Approximation

Fit approximation model to P_{g+1}
Estimate objective values for Q_{g+1} and store as $\hat{Q}_{g+1}$

8. IPE - Fitness Assignment

Combine $\hat{R}_{g+1} = P_{g+1} \cup \hat{Q}_{g+1}$
Determine fitness value by dominance and crowding distance

9. IPE - Selection

Select N best solutions (based on fitness) from $\hat{R}_{g+1}$ and store as A_{g+1}
Set $Q_{g+1} = A_{g+1} \cap \hat{Q}_{g+1}$

10. Function Evaluation

Determine objective values for all solutions in Q_{g+1}
Set $g = g + 1$ and continue with step 2

that a lot of solutions will not be evaluated, which reduces calculation times significantly. Furthermore the fact that only approximate objective values are necessary makes it possible to combine this approach with any of the approximation methods from chapter 4. Algorithm 5.1 shows how the IPE approach can easily be incorporated into the original NSGA-II algorithm.

The Probability of Improvement (PoI) approach, more or less covers the drawbacks of the IPE approach by taking into account the uncertainty around the approximated values. It therefore allows solutions that are just behind the Pareto front to be evaluated exactly, if the uncertainty of the objective values in this point is large enough. This however requires this approach to have estimates of the errors in objective values for each of the solutions, something that can only be provided by the Kriging/DACE approximation method.[2] Roughly speaking there are three possible approaches to select the 'best' solutions for exact evaluation. The first is by allowing only the best n solutions to be evaluated exactly, main benefit of this approach is that one can determine the saving in calculation times up front. Drawback of this approach is that it is possible that good solutions are discarded because they are not amongst the best n solutions. The second approach is selecting only solutions that have a PoI higher than $x\%$, thus only allowing solutions that

[2]It is however not quite straightforward to calculate the PoI for problems with three objective functions or more, Appendix C describes how this can be done.

Algorithm 5.2 Probability of Improvement (PoI)

1. Initialisation – 6. Variation

see the generic NSGA-II algorithm on page 13

7. PoI - Function Approximation

Fit approximation model to P_{g+1}
Estimate objective values and errors for Q_{g+1} and store as $\hat{Q}_{g+1}$

8. PoI - Probability Calculation

Determine for each solution in $\hat{Q}_{g+1}$ the probability of improving the Pareto front formed by P_{g+1}.

9. PoI - Selection

Sort the solutions in $\hat{Q}_{g+1}$ based on Probability of Improvement and select the n best solutions and/or solutions with a PoI higher than $x\%$ and store those in Q_{g+1}

10. Function Evaluation

Determine objective values for all solutions in Q_{g+1}
Set $g = g + 1$ and continue with step 2

are very likely to improve the Pareto front to be evaluated exactly. Main drawback of this approach is that is is possible that the algorithm decides to evaluate all solutions, thereby providing no computational benefits. The third approach combines the previous two and selects only the solutions with a PoI higher than $x\%$ with a maximum of n solutions. This approach ensures that only solutions that are likely to improve the Pareto front are evaluated, whilst also providing a fixed minimum reduction in computational effort. Drawback of this latter approach is that it is possible that good solutions are removed (based on the fixed number n) in one iteration, where in another iteration no solution is evaluated (based on the percentage x). Concluding we state that PoI allows us to use knowledge about the uncertainty, whilst providing a guaranteed computational effort reduction. The combination of these two aspects makes this a very interesting approach. Algorithm 5.2 shows that also this approach con conveniently be incorporated into the NSGA-II algorithm.

Finally Expected Improvement (EI) is (as mentioned earlier) an enhancement of PoI. The main advantage of EI over PoI is that it does not only consider the probability of improving the Pareto front, but also the extend to which the objective values are reduced. This means that a solution which has a low probability of improving the Pareto front significantly might be preferred over a solution that has a high probability of improving the Pareto front only

marginally. However computing the true EI for multiobjective problems is, although similar to the approach used to determine the PoI, somewhat difficult. The work by Keane (2006) provides quite convenient equations for both PoI and EI for a biobjective case, but the more general approach described in appendix C for determining the PoI for multiobjective problems cannot be applied when determining EI. This makes this approach less suitable for most realistic cases (i.e. cases with more than two objective functions).

5.3 Conclusions

Based on the discussion above, it seems that the DoE approach is less suitable for this specific application. This is mainly because NSGA-II aims at selecting those solutions that form the Pareto front in objective space, whereas a good model for applying DoE requires a good spread in solution space. In section 5.2 we briefly mentioned some possible solutions to this problem, however given the limited time available we decided not to develop these approaches and thereby decided not to select Design of Experiments as one of the metamodels.

The second approach that is rejected as metamodel, is Expected Improvement. The main reason is that it seems that, in order to be able to use the EI approach, we require an algorithm that is able to compute this value for multiobjective problems in an efficient way. Until such an algorithm has been found EI requires a custom made algorithm (custom made for a specific number of objective functions), which makes this approach unattractive. Another issue that should be taken care of before applying this approach is normalisation, otherwise it would be impossible to combine the expected improvements for individual objectives.

However IPE, as a very intuitive approach, seems able to reduce calculation times significantly. This reduction is achieved because only solutions that are considered to be part of the Pareto front (by the approximation method) are evaluated using the exact model. Another benefit is that it requires almost no additional computations, again something that is beneficiary from a computational effort point of view. This is why the Inexact Pre Evaluation is selected as one of the metamodels.

The second approach that is selected is the Probability of Improvement approach. The main reason why this approach is selected is that it extends the options of IPE, by incorporating uncertainty around the expected objective values. This enables the algorithm to evaluate solutions that are close to the Pareto front (but not part of the Pareto front) if we are uncertain of the objective value. This does however come at a cost, because the number of solutions that is exactly evaluated is likely to increase significantly.

In the next chapter we will combine the two metamodels that were selected in this chapter and the two approximation techniques that were selected in

chapter 4. A combinations of metamodels and approximation techniques will be applied to our test networks and their performance will be measured.

Chapter 6

Accelerating NSGA-II

An algorithm must be seen to be believed.

Donald E. Knuth (1938 –)

This chapter starts where chapters 4 and 5 (Approximation Techniques and Metamodel Assisted Evolutionary Algorithms) ended. In section 6.1 we combine the results from these two chapters into three Approximation Method Assisted NSGA-II algorithms (AMANs) which will be applied to the test networks from sections 3.1 and 3.2. The performance of each of these AMANs will be measured using criteria which are developed in section 6.2 and we will discuss the performance in section 6.3. Finally in section 6.4 we will select one single approach that will be applied to the more realistic case of Almelo, which is discussed in chapter 7.

6.1 Approximation Method Assisted NSGA-II

As mentioned above AMANs are a combination of an approximation method and NSGA-II. In chapter 4 we found that a Response Surface Method (RSM) with squared cubic interaction terms and Kriging/DACE with the objective function of Mardia and Marshall (1984) and a ϕ start value of $1E^{-7}$ are amongst the best approximation methods, when 'making the right decision' is the main objective (see section 4.8). Furthermore the Kriging/DACE approach also performs very good when the main objective is 'approximating objective values'.

We can now combine the two approximation methods with Inexact Pre Evaluation (IPE) and Probability of Improvement (PoI), the approaches we found to be promising in chapter 5. This results in three different AMANs,[1] of which an overview can be found in Table 6.1.

[1] In theory we could have constructed a fourth AMAN. This fourth AMAN, however, proved to be unfeasible, as we will explain later on.

		Approximation Techniques	
		Response Surface Method	Kriging/DACE
MAEA approach	Inexact Pre Evalution	IPE-RSM	IPE-DACE
	Probability of Improvement	-	PoI-DACE

Table 6.1: Overview of AMAN approaches

IPE-RSM, the AMAN from the top left corner of Table 6.1, is the combination of the most simple approaches. One approximates the objective values using cubic squared RSM and evaluates only those solutions that appear to be on the Pareto front. In this case we will use only the solutions that are part of the parent set as learning data.

IPE-DACE combines an intuitive approach in the usage of the data, with a complex approximation model. We have shown that Kriging/DACE is capable of predicting the objective values extremely well (section 4.7), note that Kriging/DACE was also successful in predicting whether the solutions were on the Pareto front. Because of the complexity of the approximation model, we will only use parent solutions as learning data. Unfortunately this approach does not make use of all the information that is generated by Kriging/DACE.

PoI-DACE does make use of the additional data, kriging errors, that are generated by the Kriging/DACE approach. In this case the probability that the solution will improve the Pareto front is determined for each of the solutions. Now we have three possible ways of selecting solutions for exact evaluation: a) one could select the best n solutions to be exactly evaluated, which ensures that the computational effort can be predicted up front; b) one could use only the solutions which have a predicted improvement of at least $x\%$; or c) one combines these two approaches. We decided to use option c for two reasons. First because we only wanted to evaluate solutions that were likely to improve the Pareto front (defined as solutions with a PoI larger or equal to 50%). Second because we assume that over time more and more solutions will meet the $x\%$ criterion,[2] we like to restrict ourselves to using only the n best solutions where n decreases over time (defined as 75 in the first iteration and linearly decreasing to 25 in the last iteration, assuming an offspring size of 100). An additional benefit is that we are sure that we will decrease computational effort (actually the number of exactly evaluated solutions) by 50%.

PoI-RSM, the approach that should be found at the bottom left corner of Table 6.1, cannot be considered a realistic AMAN. This because PoI requires the knowledge about the error in objective values for a specific point and RSM

[2]Since we start with a completely random set of solutions, it is likely that the solutions that are generated by the Genetic Algorithm move slowly towards the true Pareto front. Afterall that is the concept on which Genetic Algorithms are based.

is unable to deliver this information. Although we could device solutions to this problem, e.g. using the average realised Root Mean Squared Error (RMSE) from the previous iteration, we decided not to use this combination.

We will first apply the 'ordinary' NSGA-II to the two test networks (described in sections 3.1 and 3.2), using 100 iterations. In each iteration we will generate 100 children from 100 parents. Using the same set of start parents as the 'ordinary' NSGA-II, we will apply the three AMANs that were described above (IPE-RSM, IPE-DACE and PoI-DACE) to the same Test Networks, again 100 iterations in which 100 children are generated from 100 parents.

It is important to note that, due to the stochastic behaviour of Genetic Algorithms, we cannot compare the results with absolute certainty. In order to be truly able to compare the different AMANs we would need to perform dozens (if not hundreds) of runs, before the we can conclude (with 95% certainty) that one approach is better than another. Due to time limitations we are unable to peform such an extensive analysis, and we should therefore discuss the results with care. Furthermore it is important to note that if we would remove all stochasticity from the equation, e.g. by using only a single iteration (which can then be the same for all approaches), the Pareto front found by an AMAN (F_{AMAN}) can never outperform the Pareto front by the Genetic Algorithm (F_{GA}). This is because the set of solutions that form the Pareto front of the AMAN is a subset of the set of solutions that form the Pareto front of the Genetic Algorithm $F_{AMAN} \subseteq F_{GA}$, and thus it holds that $F_{AMAN} \preceq F_{GA}$.

After applying the different AMANs (and the 'ordinary' GA) to the two test networks, we can compare the results using the criteria described in the following section.

6.2 Performance Measures

Before we can determine which performance measures are suitable, we have to determine when we consider an AMAN to be better than another. One of the ideas of the Genetic Algorithm approach is that we try to come as close as possible to the true Pareto front. Since the location of this front is unknown we cannot measure this 'fitness' directly. We can however determine to what extend solutions are dominated by a Pareto front. We can therefore state that a Pareto front that dominates a larger part of the objective space is considered to be a better solution. Furthermore we would like the solutions to be evenly spread in the objective space, since this increases the possibilities for researchers and principals to choose a solution that they think is best for the current situation. If all solutions (except one) are very close together, we still know very little about the problem. We therefore can say that a Pareto front

in which the solutions are more evenly spread in objective space is considered to be a better solution.
Summarised we find that a Pareto front is 'better' when:

1. the volume of the objective space that is dominated by the Pareto front is larger; and

2. the solutions are more evenly spread over the Pareto.

We will refer to performance measures for the first criterion as *dominance metrics* or performance measures for dominance, and refer to performance measures for the second criterion as *diversity metrics* or performance measures for diversity.

It is interesting to note that these two criteria are also commonly used in literature. For instance Deb (2001) identifies two major classes of performance measures for multiobjective evolutionary algorithms: the first are the metrics for convergence, the second the metrics for diversity. Deb uses the convergence metrics to check if the new generation of the evolutionary algorithm has improved enough to continue with the algorithm. The criteria for convergence are however similar to the dominance criterion defined above. This enables us to use performance measures that are developed for convergence as performance measures for dominance.

The convergence metrics can be used to determine whether solutions improve over time, and as such are useful to abort a Genetic Algorithm if there is no improvement for some iterations. However, they can also be used to compare the results of two different AMAN approaches. One should however be cautious, if an AMAN approach scores better on a specific convergence (dominance) metric it is not necessarily a better solution (Okabe, Jin & Sendhoff, 2003), as we will show later on in this section. It is therefore important to use a multitude of dominance metrics in order to be able to compare two AMAN approaches.

The diversity metrics focus on how well the individual solutions are spread in the Pareto front, and can therefore be used to determine to what extend a Genetic Algorithm (or AMAN) allows global search. Here, however, one should be careful, because the dimensions of the different objective functions are usually different, which makes normalisation necessary. Finally Deb (2001) also mentions a third class, which contains performance measures that measure both convergence and diversity, they are however quite rare.

As starting point for the literature search we take the work by Grosan, Oltean and Dumitrescu (2003) and to a lesser extend the work by Tan, Lee and Khor (2001), who provide us with a good (and fairly recent) overview of dominance (convergence) and diversity metrics. For a multitude of performance measures related to the ones mentioned below we refer the interested reader to the work by Okabe et al. (2003).

Dominance Metrics

One of the most intuitive performance measures is the C-Metric (also known as Coverage of Two Sets; CTS), as discussed in Grosan et al. (2003) and Tan et al. (2001). The approach was developed by Zitzler and Thiele (1998, 1999) and improved in the PhD thesis of Zitzler (1999).

The main idea behind the C-Metric is that a Pareto front A is more interesting than Pareto front B if many solutions of B are (weakly) dominated by A (denoted by $A \succeq B$). In this case Pareto front A 'covers' Pareto front B. In order to determine to what extend A covers B we therefore count the number of solutions in B that are dominated by a solution from A and divide this by the total number of solutions in B. Or in mathematical notation:

$$C(\mathrm{A},\mathrm{B}) = \frac{|b \in B / \exists\ a \in A : a \succeq b|}{|B|} \tag{6.1}$$

Now $C(\mathrm{A},\mathrm{B})$ is the fraction of solutions in B that are covered by A, i.e. if $C(\mathrm{A},\mathrm{B}) = 1$ Pareto front B is completely within Pareto front A and thus A is much better than B. Note that $C(\mathrm{A},\mathrm{B}) + C(\mathrm{B},\mathrm{A})$ is not necessarily equal to one, which has two reasons. Firstly because solutions in which $a = b$ are included in both and secondly because Pareto fronts do not need to dominate each other at all.

The S-Metric is also developed by Zitzler and Thiele (1998, 1999) and improved in the PhD thesis of Zitzler (1999). Both Grosan et al. (2003) and Tan et al. (2001) discuss the approach, although the latter refer to it as 'Size of Space Covered'. The definition of Tan et al. gives a good indication of the concept behind the performance measure.

The S-Metric (also known as Size of Dominated Space, Size of Space Covered; SSC, Hyperarea or Hypervolume; HV) determines the hypervolume of the Pareto front, i.e. it determines the size of the area that is dominated by a solution. This however requires the definition of a utopian point that is the 'worst' possible solution in all objectives. For a maximisation problem (with objective values that are strictly positive) such a point can easily be found, the origin is a utopian point. Clearly in this case a larger value of S indicates a better solution (the size of the dominated area is larger).[3] However for a minimisation problem one would have to device a point that consists of the theoretical maximum of each objective function. Unfortunately it is impossible to find the true maxima for the objective values, which leaves us with little other solution than selecting the maximum value for each of the objectives as indicator for the utopian point. Note that (when comparing different AMAN approaches) this means selecting the maximum objective values over the combined set of AMAN approaches. Furthermore we do not use the

[3]Note that the origin can be a very unrealistic utopian point when it is (physically) impossible for the objective values to reach zero.

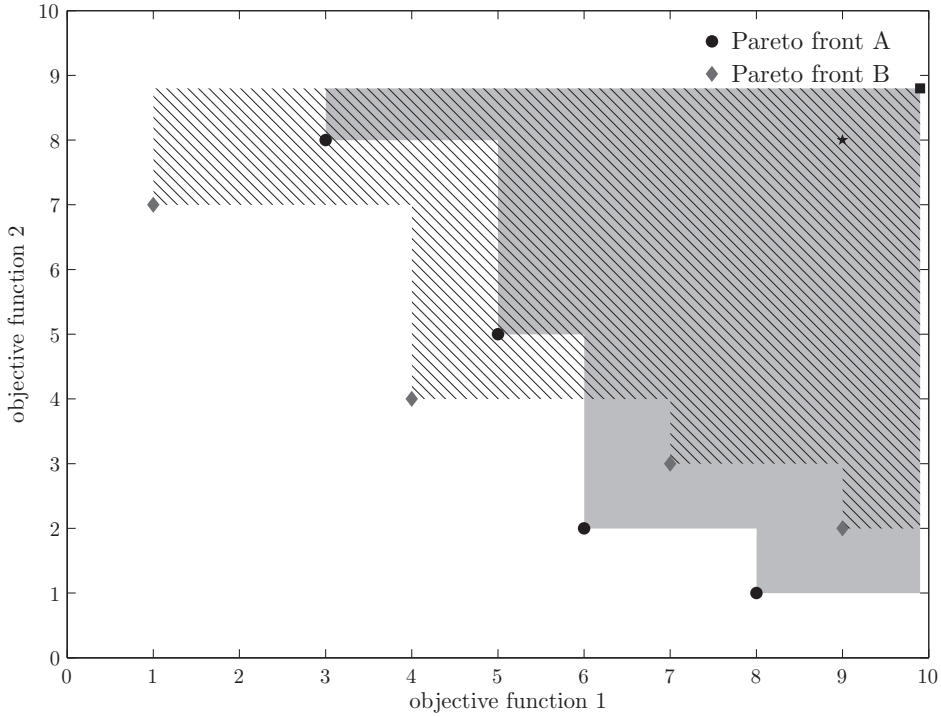


Figure 6.1: Example of the S- and D-Metric performance measures

maximum values (f_i^{max}) themselves, but $1.1 \times f_i^{max}$ denoted by $f_i^{\overline{max}}$, in order to ensure that if multiple solutions have the same maximum value in one of the objectives, the one with the lower objective value scores better on the performance measures. In order to determine the hypervolume for multiobjective problems we use the Hypervolume by Slicing Objectives (HSO) algorithm of While, Hingston, Barone and Huband (2006).
Figure 6.1 shows two Pareto fronts, front A is marked by black dots and front B is marked by grey diamonds. Furthermore it shows the theoretical maximum point, which is marked by a black star and the point that is used for calculating the hypervolume marked by a black square. Then the S-Metric for both Pareto fronts can be determined, $S(\mathrm{A}) = 22$ (solid grey area) and $S(\mathrm{B}) = 25$ (hatched area). This suggests that Pareto front B is better, because the size of the dominated area is larger.

The D-Metric (also known as Coverage Difference of Two Sets; CDTS) approach was designed because there are situations in which C-Metric is unable to decide when a Pareto front is better than another (Grosan et al., 2003; Okabe et al., 2003), for example in the case where $C(\mathrm{A}, \mathrm{B}) = C(\mathrm{B}, \mathrm{A})$ (see Figure 6.1). However in such cases it still is possible to decide which Pareto front is better, for Zitzler (1999) proposed a method using the S-Metric values that have already been obtained. The main idea is that the difference

in S-Metric between the combined Pareto fronts $S(\mathrm{A} \cup \mathrm{B})$ and the individual S-Metric of the competitor $S(\mathrm{B})$ is the unique contribution of Pareto front A. We can now define D(A,B) as the unique contribution of A, when being merged with B.

$$D(\mathrm{A}, \mathrm{B}) = S(\mathrm{A} \cup \mathrm{B}) - S(\mathrm{B}) \tag{6.2}$$

Clearly the Pareto front which unique contribution D is larger, is a better Pareto front. If we go back to the example from Figure 6.1 we find that $S(\mathrm{A} \cup \mathrm{B}) = 30$ (combination of the grey and hatched area) and thus we can determine that $D(\mathrm{A}, \mathrm{B}) = 5$ and $D(\mathrm{B}, \mathrm{A}) = 8$. This indicates that the unique contribution from B is larger and as such B is a better Pareto front. Note that $C(\mathrm{A}, \mathrm{B}) = 0.50$ and $C(\mathrm{B}, \mathrm{A}) = 0.50$ which is exactly the reason why we need the D-Metric in addition to the C-Metric.

Based on Zitzler (1999), Grosan et al. (2003) suggests that instead of the absolute value of D, it is better to use the relative value of D, denoted by D_{rel}, which can be calculated using:

$$\begin{aligned} D_{rel}(\mathrm{A}, \mathrm{B}) &= \frac{D(\mathrm{A}, \mathrm{B})}{V} \\ \text{where } V &= \prod_{i=1}^{o} \left(f_i^{\overline{max}} - f_i^{min} \right) \end{aligned} \tag{6.3}$$

Here o is the number of objective functions and $f_i^{\overline{max}}$ and f_i^{min} represent the (adapted) maximum and minimum value for objective function i over $A \cup B$ respectively. In other words, V is the smallest cube possible that contains all solutions in both A and B. The reason why the use of D_{rel} might be preferred over the use of D is that it puts the improvement of one Pareto front over another into perspective: is a solution that adds less than 1% to the dominated set really 'better' than the other solution? In our problem from Figure 6.1 we find that $V = (9-1)(8-1) = 56.$ and thus $D_{rel}(\mathrm{A}, \mathrm{B}) = 0.089$ and $D_{rel}(\mathrm{B}, \mathrm{A}) = 0.143$, of course this does not change the conclusion drawn in the paragraph about D-Metric.

It is however necessary to make some comments on the S- and D-Metric performance criteria. This is because both are sensitive to the range of the different objective functions, something that is best explained using an example. Assume a maximisation problem in which we have a known point (1,1), furthermore we found that the maximum value for objective one is 2 and for objective two is 4. If we would now find a solution that optimises our problem in objective one we find point (2,1) with a value for the S-Metric of 2. If we would optimise the same problem for objective two, we find point (1,4) with a value for the S-Metric of 4. This seems somewhat strange, because we cannot state that solution (1,4) is indeed twice better than solution (2,1), in fact it is more likely that both solutions are equally good since they optimise the problem for one objective only.

It is important to note that the order of magnitude of the different objective functions is not important. If in the previous example the values for objective one would be 1000 and 2000, we would find $S_{2000,1} = 2000$ and $S_{1000,4} = 4000$ and the conclusion would still be the same.

A solution to this problem could be found by normalising the objective values, thus ensuring that all objective values are always on the interval $[0, 1]$, where the solution with the minimum objective value is normalised to 0 and the solution with the maximum objective value is normalised to 1. This normalisation will be indicated by the superscript N and could be performed by:

$$x_i^N = \frac{x_i - f_i^{min}}{f_i^{\overline{max}} - f_i^{min}} \tag{6.4}$$

Here x_i^N is the normalised objective value for objective i for the original objective value x_i. $f_i^{\overline{max}}$ and f_i^{min} are the (adapted) maximum and minimum for objective i over all solutions respectively. Going back to the example of Figure 6.1, this would result in $S^N(\mathrm{A}) = 0.3929$, $S^N(\mathrm{B}) = 0.4464$, $S^N(\mathrm{A} \cup \mathrm{B}) = 0.5357$ and thus $D^N(\mathrm{A}, \mathrm{B}) = 0.0893$ and $D^N(\mathrm{B}, \mathrm{A}) = 0.1428$. Note that these figures (for D-Metric) do strongly resemble the results for D_{rel} when not applying normalisation, this is due to the fact that the range of both objective functions is similar. When this normalisation is applied, the value of D_{rel} becomes irrelevant, because $V = 1$ and thus $D = D_{rel}$.

Diversity Metrics

The first diversity metric discussed here is the spacing metric Δ that was introduced by Deb, Agarwal, Pratap and Meyarivan (2000) and is mentioned in both Grosan et al. (2003) as well as Okabe et al. (2003). The Δ measures the distances between consecutive points on the Pareto front and aims to get a distribution that is as even as possible. Deb et al. (2000) defines Δ_{orig} therefore according to:

$$\Delta_{orig} = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}} \tag{6.5}$$

Here d_f and d_l represent the distance from the known extreme solutions to the edges of the Pareto front under consideration. N is the number of solutions in the Pareto front, d_i is the distance from solution i to solution $i + 1$ and $\bar{d}$ is the average over all d_i. However in our situation the true extreme solutions are often unknown, it is therefore that Grosan et al. (2003) decided to leave out d_f and d_l, thereby reducing the Δ_{orig} to Δ:

$$\Delta = \frac{\sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{\sum_{i=1}^{N-1} d_i} \tag{6.6}$$

The major drawback of the approach by Deb et al. (2000) is that it can only be applied to biobjective problems (Grosan et al., 2003; Okabe et al., 2003), because in multiobjective problems it is not possible to determine which solutions are consecutive (for an example see Figure 6.4b).

There is however another approach, and that is the application of Δ' which was introduced by Schott (1995), and is mentioned quite often in literature (Grosan et al., 2003; Okabe et al., 2003; Tan, Khor & Lee, 2005). Schott introduced the approach for a bi-objective problem, but it can easily be extended to the multi-objective case. The original equation by Schott (1995) is:

$$f_{spacing} = s^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(d_i - \bar{d}\right)^2 \tag{6.7}$$

$$\text{where } d_i = \min_{j,j\neq i}\left\{\left|f_i^1 - f_j^1\right| + \left|f_i^2 - f_j^2\right|\right\} \tag{6.8}$$

Here N is the total number of points in the Pareto front, f_i^o is the value of objective function o for point i and $\bar{d}$ is the average value of d_i overall points in the Pareto front. For clarity: d_i is the 'Manhattan' distance between i and the nearest other point. Both Grosan et al. (2003) and Okabe et al. (2003) decided not to use the s^2 value but simply s which makes it more comparable to Δ. Now define O as the number of objective functions, then the more general definition is:

$$\Delta' = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(d_i - \bar{d}\right)^2} \tag{6.9}$$

$$\text{where } d_i = \min_{j,j\neq i}\sum_{o=1}^{O}\left|f_i^o - f_j^o\right| \tag{6.10}$$

Unfortunately Δ' also has a major drawback, which is that it tends to cluster points on the Pareto front (Okabe et al., 2003). This problem is easiest understood using an example. In Figure 6.2 we see two sets of Pareto fronts, A (black dots and squares) and B (grey diamonds), of a bi-objective optimisation problem. Clearly Pareto front B is better (more evenly) spread than Pareto front A. This is represented by the performance measure Δ, since $\Delta(\text{A}) = 0.686$ and $\Delta(\text{B}) = 0.075$. Note that $\Delta = 0$ as well as $\Delta' = 0$ indicates a solution where the distance between all points is exactly equal. However if we use the Δ' of Schott (1995) we find that $\Delta'(\text{A}) = 1.095$ and $\Delta'(\text{B}) = 0.471$. Now it seems that (although there still is quite a big difference) that the two Pareto fronts are more comparable. This is caused by the fact that the large gap between the third and fourth point in Pareto front A is completely ignored by Δ'.

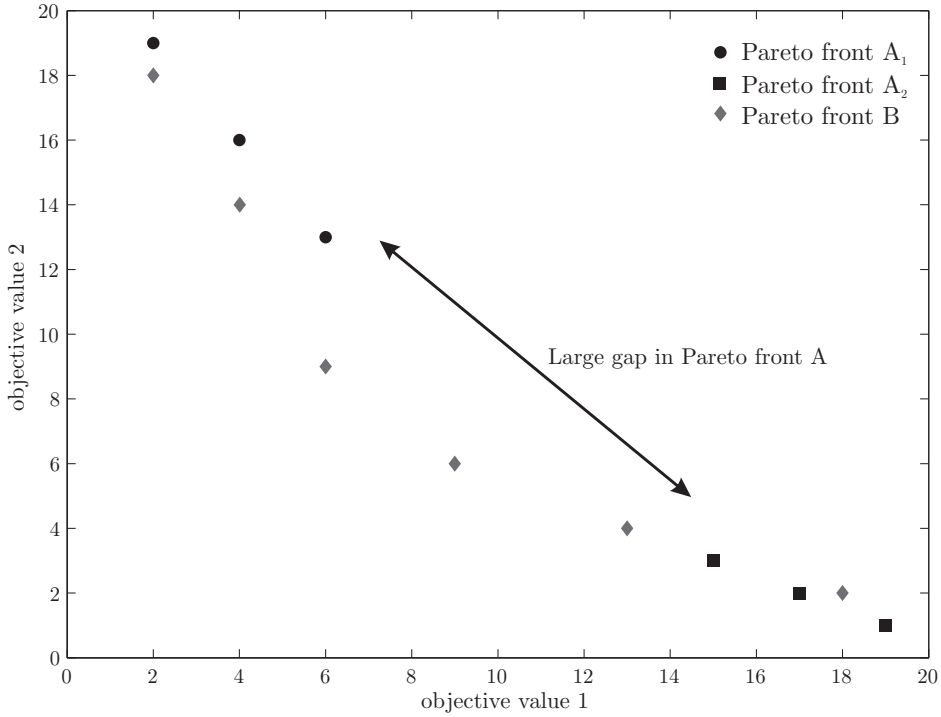


Figure 6.2: Example of the Δ and Δ' performance measure

Wismans, Van Berkum and Bliemer (2011) notice that if (in the example of Figure 6.2) Pareto front A would be split in two parts (the three black dots left of the gap are called A_1 and the three black squares on the right are called A_2) both would score perfect on Δ and Δ'. However when looking at the entire set of solutions (the combination of A_1, A_2 and B) we clearly see that the spread in reality is not so good, in fact, only local solutions have been found. Wismans et al. (2011) therefore suggest adding the solutions that contain the extreme values (that is, the minimum or maximum value for an objective function known amongst all solutions) to each Pareto front, thereby measuring the spread over the entire front. In Figure 6.2 we would therefore include the solutions $(2, 19)$ and $(19, 1)$ in all the Pareto fronts. If we use ∇ and ∇' to represent Δ and Δ', only now also including the extreme solutions, we find that: $\nabla(A_1) = 0.762$, $\nabla(A_2) = 0.991$ and $\nabla(B) = 0.400$. Furthermore $\nabla'(A_1) = 8.660$, $\nabla'(A_2) = 11.258$ and $\nabla'(B) = 2.278$. Not only does this show how wrong Δ and Δ' can be when the extreme values of Pareto fronts are different, it also illustrates how 'wrong' the approach of Schott (1995) can be when there is a large gap in a Pareto front.

Another issue that comes along with the approach of Wismans et al. (2011) can best be illustrated using an example. If we consider Pareto front A_1 and Pareto front A_1', which is Pareto front A_1 plus the extreme solution $(19, 1)$,

from Figure 6.2 we find that $\nabla'(A_1) = \nabla'(A'_1)$. The performance measure ∇' is thus unable to differentiate between these two solutions, which can be considered a drawback. The approach is, however, better than the original Δ', which would prefer the solution that did not include the extreme solution, because $\Delta'(A_1) < \Delta'(A'_1)$.

Furthermore it is interesting to note that Δ' is an absolute performance measure (it is only dependent on the Pareto front under consideration) whereas the ∇' performance measure is relative, i.e. the performance of a specific Pareto front is dependent of the other Pareto fronts which are under consideration. It should also be noted that the difference between Δ' and ∇' is only marginal if their are many points on the Pareto front.

Finally we have a problem that is similar to the problem we had with the S- and D-Metric approaches. With the S- and D-Metric approaches we found that the range of the objective functions would cloud the performance measures, in this case however the order of magnitude of the different objective values could substantially influence the results. This problem is caused by the fact that noise always has objective values that are somewhere around 70, whereas the objective values for TTT and CO_2 can easily be in the millions, billions, or even more. The difference between two solutions in noise would therefore contribute only marginally (if anything at all) to the performance in diversity. The solution to this problem is however similar to the solution we deviced for the problem with the S- and D-Metric. We therefore decided that, in order to reward AMANs with an even spread in objective space, we also should apply the normalisation technique from the previous subsection to ∇ and ∇'. Without repeating the procedure for normalisation again (see equation 6.4) we can define ∇^N and $\nabla^{N\prime}$.

Normalising ∇' is extremely useful and can sometimes even reverse the conclusions, something we can best explain using an example. Consider a problem with three known solutions (1,1000), (2,200) and (10,100), i.e. we have a problem where one objective function has objective values that are hundredfold the objective values of the other objective function. In this non-normalised case we find that $d_1 = 801$ and $d_2 = d_3 = 108$, resulting in $\bar{d} = 339$ and finally $\nabla' \approx 400$. The conclusion would thus be that the solutions are extremely unevenly spread over the Pareto front. However if we would ignore the magnitude of the different objective functions, we find that, just by looking at the data, the solutions appear to be extremely well spread over the Pareto front. This can easily be verified using the normalised objective values (0, 0.8), (0.1, 0.1) and (0.8, 0). In that case we find that $d_1 = d_2 = d_3 = \bar{d} = 0.9$ and thus $\nabla^{N\prime} = 0$, which indicates a perfect spread over the Pareto front.

Other Performance Measures

Literature however also suggests a couple of performance measures that do not measure the quality of the answer, but the quality of the algorithm instead.

Tan et al. (2001) suggests the use of the Algorithmic Effort (AE) measure, defined as:

$$AE = \frac{T_{run}}{N_{eval}} \tag{6.11}$$

Where T_{run} is a fixed amount of time and N_{eval} is the number of function evaluations the algorithm performs. Of course a higher number of evaluations in the same amount of time would yield a 'better' algorithm. Since we do not (explicitly) limit the amount of time an AMAN may use, we measure the effort of the algorithm by determining the time the AMAN needed to finish over the number of solutions that was exactly evaluated. As such it represents the average time needed per exact evaluation. We therefore define AE' as:

$$AE' = \frac{T_{calc}}{N_{eval}} \tag{6.12}$$

However it is also interesting to measure how many solutions are rejected by the AMAN based on the approximation method. We therefore define the Fraction of Accepted Solutions (FAS) as:

$$FAS = \frac{N_{eval}}{N_{proposed}} \tag{6.13}$$

Here N_{eval} is the number of solutions that is exactly evaluated, whereas $N_{proposed}$ is the total number of solutions that is generated by the Genetic Algorithm (i.e. the number of iterations multiplied by the number of children that is generated in each iteration).

Furthermore it is interesting to combine the results of FAS and AE' into a single performance measure that gives a normalised (all AMANs are divided by the same value $N_{proposed}$) indication of the total computational effort of an AMAN:

$$FAS \cdot AE' = \frac{N_{eval}}{N_{proposed}} \cdot \frac{T_{calc}}{N_{eval}} = \frac{T_{calc}}{N_{proposed}} \tag{6.14}$$

Another more general performance measure is the Ratio of Non-dominated Individuals (RNI), mentioned by (amongst others) Tan et al. (2001) and M. Kim et al. (2004). This performance measure determines the fraction of solutions in the total set of evaluated solutions that form the Pareto front. We therefore define:

$$RNI = \frac{N_{Paretofront}}{N_{eval}} \tag{6.15}$$

In the next subsection we will select different measures from the dominance, diversity and other performance measures, which will later be used to assess the results of the different AMANs.

Selecting Performance Measures

Unfortunately many of the performance measures assume that a true Pareto front is known, and are based on determining the distance between the known and true Pareto front (see e.g. Tan et al., 2001; Grosan et al., 2003; Okabe et al., 2003; Tan et al., 2005). We therefore are limited to selecting performance measures from the subsections above.

From the dominance metrics we first of all select C-Metric (equation 6.1). Although we have shown that the C-Metric alone might not be enough to determine the 'best' solution, it does provide valuable insight into how a solution set contributes to the combined Pareto front.

We have shown that the value of the S-Metric in itself is not of much use and we therefore decided to use only the resulting performance measure, the D-Metric. Because of the large range differences that might be found in the different objective function values, we reject the original D-Metric approach and resolve to the normalisation technique that was proposed. We therefore decided to use the D^N derived from equation 6.2, using the normalisation function described in equation 6.4.

From the divergence metrics it seems that the original performance measures Δ and Δ' are biased when the extreme solutions differ from Pareto front to Pareto front. Especially in our case (considering discrete parameters and many solutions) it seems unrealistic to assume that the extreme solutions of two Pareto fronts are equal. Furthermore we have to consider the magnitude of the objective functions and it therefore only seems fair to use the normalised data to determine these performance measures. It is therefore that we use the performance measure $\nabla^{N\prime}$, based on equation 6.10 including the known global extreme solutions and applying the normalisation technique from equation 6.4.

From the performance measures for the algorithms, we propose to use the AE' (see equation 6.12) to measure the efficiency (how long does a single exact evaluation take) of the different algorithms.

Furthermore the FAS (equation 6.13) gives (in combination with the AE') a good indication of the savings in calculation times, since it shows how often an exact evaluation is unnecessary. For completeness we will also include the FAS $\cdot$ AE' (equation 6.14) as indication for the total calculation time that the algorithm has used.

Finally RNI (equation 6.15) shows the fraction of the exactly evaluated solutions that form the Pareto front. Clearly a solution with a high RNI is much more desirable than a solution with a low RNI. The former has contributed much more expensive computational time to the Pareto front than the latter.

All this results in 7 performance measures in three categories: algorithmic

performance (AE', FAS, FAS · AE' and RNI), dominance (C and D^N) and diversity ($\nabla^{N'}$). When analysing the performance of the AMANs the focus however will be on the latter three performance measures because they are derived from the two performance criteria that were defined on page 94.

6.3 Results

When comparing Genetic Algorithms, literature usually only compares the n parents that were selected in the last iteration (see e.g. J. D. Knowles & Corne, 2000b; Corne et al., 2001; Deb et al., 2002; M. Kim et al., 2004). In this case however it seems more appropriate to compare the subsets of all exactly evaluated solutions that form the Pareto front. After all each solution in the Pareto front adds valuable information and, given the spread in solution space, plotting more solutions gives a better overview of the 'true' Pareto front. In order to make comparisons with other literature possible, we decided to plot the performance of both the Pareto front as well as the 100 parents from the last iteration.

It should be noted (again) that the results presented in this section are obtained in a single run (i.e. each AMAN is ran only once), therefore minor differences might simply be due to the randomness that is inherent to Genetic Algorithms. In order to reduce the effect of the initial solution (i.e. the 100 parents that are used to generate offspring in the first iteration) we used the same set of start solutions for all the AMANs and the GA.

We found that plotting all combinations of the AMANs and GA for the C-Metric and D-Metric resulted in figures that were counterintuitive or incomprehensible. We therefore decided not to plot $C(\mathrm{A}, \mathrm{B})$ for all combinations of A and B from the AMANs, instead we decided to compare the three AMANs with the GA, by plotting $C(\mathrm{AMAN}, \mathrm{GA}) - C(\mathrm{GA}, \mathrm{AMAN})$, i.e. we subtract the negative effect of the GA dominating solutions from the AMAN from the positive effect of the AMAN dominating solutions from the GA. Clearly a higher score on this performance measure, indicates a better AMAN approach.[4] Similarly we decided not to plot $D(\mathrm{A}, \mathrm{B})$ for all combinations of the GA and AMANs, but only $D(\mathrm{AMAN}, \mathrm{GA}) - D(\mathrm{GA}, \mathrm{AMAN})$, i.e. we we subtract the negative effect of the area that is only dominated by the GA and not by the AMAN from the positive effect of the area that is only dominated by the AMAN and not by the GA. Again, a higher value on this performance measure indicates a better solution. Note that the values of $D(\mathrm{AMAN}, \mathrm{GA}) - D(\mathrm{GA}, \mathrm{AMAN})$ can indeed be used to compare the different

[4] Note that in reality (i.e. when all stochasticity would be excluded) the value of $C(\mathrm{GA}, \mathrm{AMAN})$ would always be 1. This is due to the fact that when there is no stochasticity $F_{AMAN} \subseteq F_{GA}$ and thus $F_{GA} \succeq F_{AMAN}$ (where F denotes a Pareto front).

AMANs, since:

$$\begin{aligned} D(\text{AMAN}, \text{GA}) - D(\text{GA}, \text{AMAN}) = \\ \{S(\text{AMAN} \cup \text{GA}) - S(\text{GA})\} - \{S(\text{AMAN} \cup \text{GA}) - S(\text{AMAN})\} \\ = S(\text{AMAN}) - S(\text{GA}) \end{aligned} \tag{6.16}$$

We also decided not to plot the Rate of Non-dominated Individuals (RNI) but the exact opposite, 1−RNI, instead. The main reason is that we could now combine it into one figure with FAS, where for both performance measures a lower value indicates a better performing AMAN.

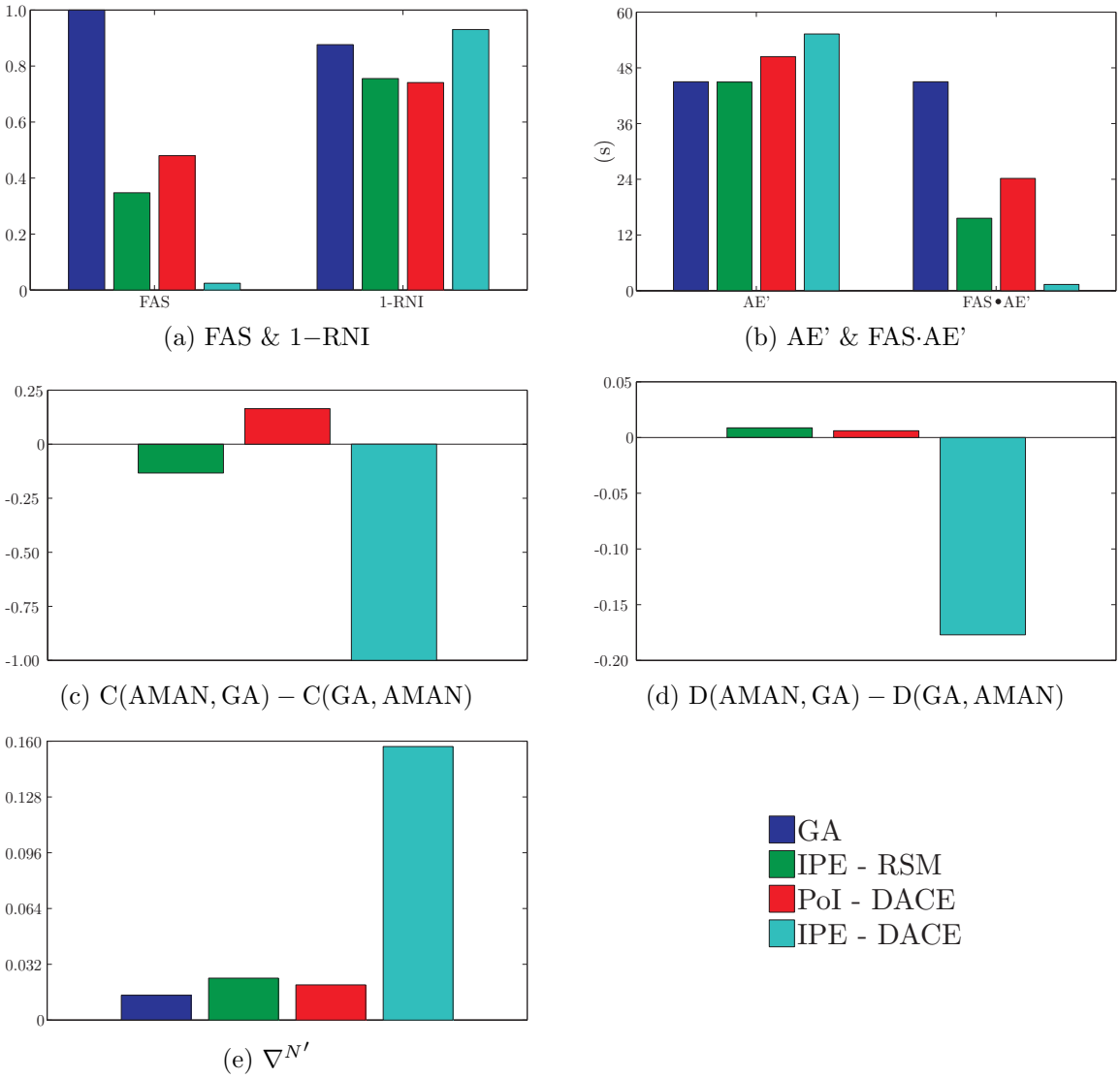
Test Network I

If we start by looking at the results shown in Figures 6.3a and 6.3b (FAS, 1-RNI, AE' and FAS·AE'), we find that the IPE-DACE approach has evaluated less than 5% of the solutions and as a result is a very fast approach. This could indicate two completely different things. Either the algorithm is by far the best approach that can be found or the algorithm encountered a problem and only produces approximations that are invalid, such as infinite objective values. We need the results for the C- and D-metric to determine which of the conclusions is right.

The results for the two other approaches (IPE-RSM and PoI-DACE) can hardly be called surprising. IPE-RSM appears to be very fast by evaluating only 35% of the proposed solutions whilst the computational effort per exact evaluation (AE') is only marginally larger than the computational effort of the original GA. PoI-DACE, however, evaluates only half of the solutions, which is consistent with the 'evaluate a maximum of n solutions' constraint that we applied to this approach. It does however also suggests that the effect of the 'evaluate only the best x%' constraint is marginal. Given the fact that PoI-DACE evaluates 15% more solutions than IPE-RSM and the fact that the average computational effort (AE') is slightly larger, we obviously find that the total computational effort (FAS · AE') of PoI-DACE is much larger than the one of IPE-RSM. It is also interesting to see that the results in RNI for both approaches are more or less similar and better than the result of the original GA.

For both the C- and D-metric (Figures 6.3c and 6.3d) we see that IPE-DACE performs extremely poor and is completely dominated by the original GA. We will discuss one of the possible reasons for these results later on. For the other two approaches (IPE-RSM and PoI-DACE) we find that the results for $D(\text{AMAN}, \text{GA}) - D(\text{GA}, \text{AMAN})$ are similar, whilst PoI-DACE appears to be marginally better on $C(\text{AMAN}, \text{GA}) - C(\text{GA}, \text{AMAN})$, which makes it difficult to decide which approach is better.

Finally we find $\nabla^{N'}$ in Figure 6.3e which shows that the spread of IPE-DACE is also quite bad. This is most likely caused by the extreme values



(a) FAS & 1−RNI

(b) AE' & FAS·AE'

(c) C(AMAN, GA) − C(GA, AMAN)

(d) D(AMAN, GA) − D(GA, AMAN)

(e) $\nabla^{N'}$

Figure 6.3: Performance Measures for Test Network I

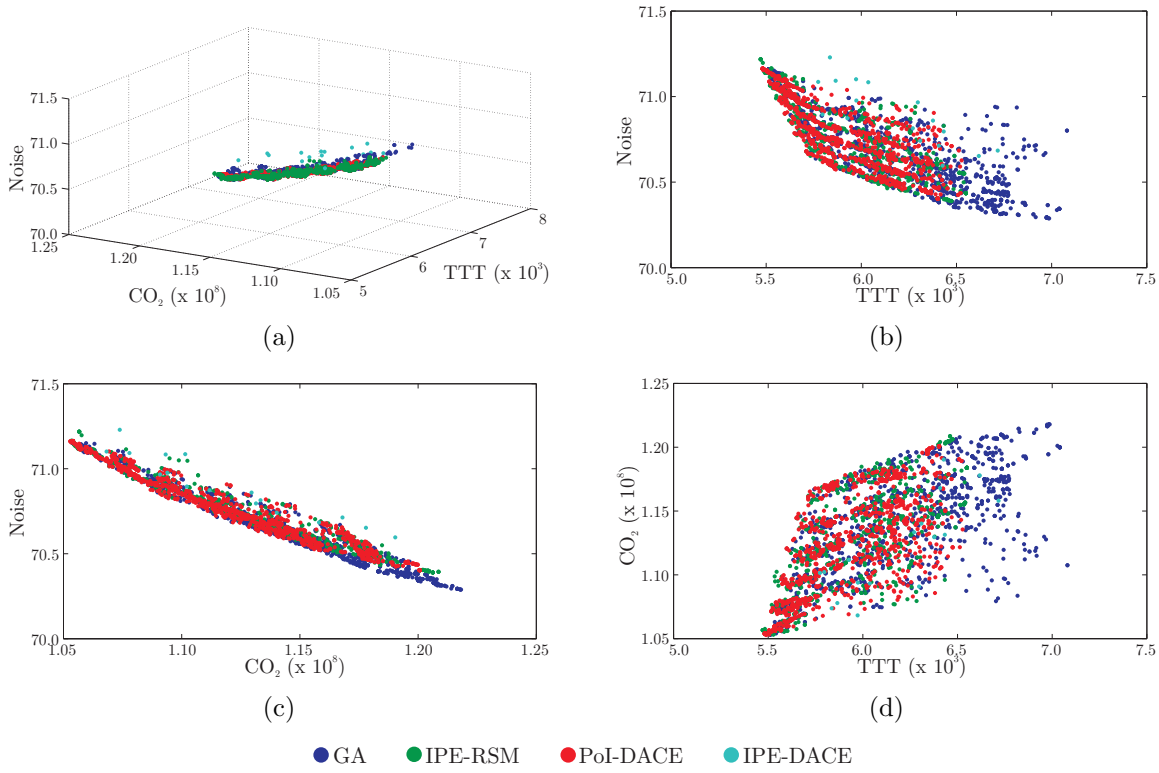


Figure 6.4: Pareto front for Test Network I

(none of which were originally part of IPE-DACE) and the fact that the Pareto front of IPE-DACE consists of only 17 solutions. We also see that the spread in GA is slightly better than the spread in IPE-RSM and PoI-DACE, but those differences are quite small. It should however be noted that due to the large Pareto fronts of IPE-RSM and PoI-DACE (both consists of over 1,000 solutions) small differences should be regarded significant.

That these small differences are indeed significant can easily be seen if one looks at the 3D plot and 2 dimensional projections of the Pareto front (Figure 6.4). It is immediately clear that the original Genetic Algorithm investigates solutions in an area that is more or less ignored by the AMANs (in fact by all the AMANs). It is however also interesting to see that this area is an area which scores poorly on both TTT and CO_2. This might suggest that the AMANs converge faster to 'better' solutions, however it also suggests that the AMANs have difficulties finding good solutions for the Noise objective. Unfortunately it is impossible to draw definite conclusions on these topics, simply because one example does not provide enough evidence.

Furthermore it is interesting to see that there clearly is a normative DTM measure, which causes the 'strokes' in the Pareto front (clearly visible in Fig-

ure 6.4b and 6.4d). Given the number of strokes it is likely that an ATS is the normative DTM measure in this test network. It is this behaviour that is (from a traffic engineering perspective) very interesting, because it indicates that the performance on the different objective functions is strongly dependent on a single DTM measure.

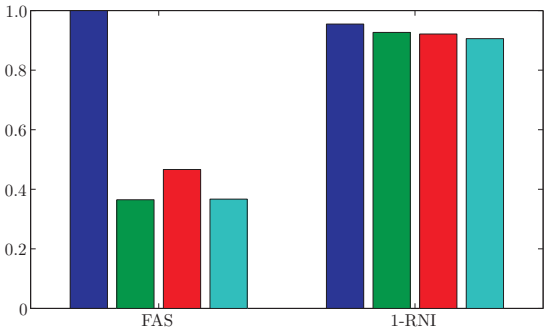
It is difficult to determine for sure why IPE-DACE performs so poor on the C- and D-Metric. However based on debugging data that was saved whilst running the IPE-DACE AMAN on Test Network I, we are able to give a founded explanation. It appears that after a few iterations the solution (which is matrix $\mathbf{R}$) of the maximum likelihood function (see equation 4.30) becomes nearly singular, causing the determinant of $\mathbf{R}$ to become 0. Although this in itself is not a problem, it causes the objective function of the optimisation function of Mardia and Marshall (1984) to become negative infinity since $\ln(0) = -\infty$, at which point the optimisation algorithm does not know in which direction to look for an improvement.

The most likely reason why a matrix $\mathbf{R}$ becomes singular is that two rows (or two columns) become equal. This might occur if a solution i is very far away in solution space from all parents j, indicated by a large kriging distance $c\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)$ (see equation 4.26) and thus a very low correlation r_{ij} (see equation 4.27). A solution that is far away from all other solutions is unlikely, but might occur if for instance all parents have the same setting for a couple of DTM measures. Any solution that does have another setting for these DTM measures might be a solution that is far away from all parents.

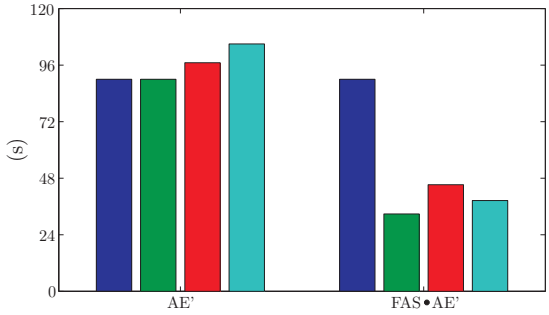
As said before, when the matrix $\mathbf{R}$ becomes singular, the optimisation algorithm is unable to determine for certain in which direction an improvement might be found. This causes the optimisation algorithm to provide answers that are unreliable, and tests in the earlier stages of this research have shown that expected objective values of (near) infinity are not uncommon. The conjecture is that during the first iterations (in which the position of the Pareto front is uncertain) many potential good solutions are removed, because their approximated position is behind the known Pareto front. This causes the set of parent solutions to become a set in which many solutions are considered to be far away from the parents, which causes unreliable predictions. Due to these unreliable (overestimated) objective values very few solutions are exactly evaluated. When using the Probability of Improvement approach this effect is marginalised, because uncertain solutions also resort in high uncertainties (see equations 4.33 and 4.34), which ensures that these solutions are evaluated.

Test Network II

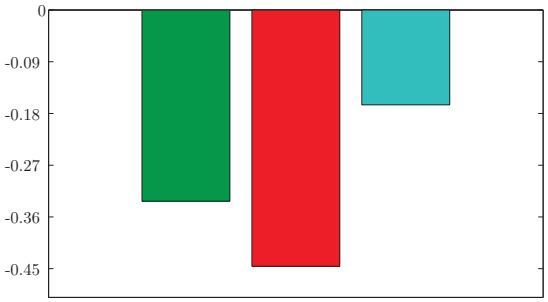
In Test Network II we see again that PoI-DACE evaluates just under 50% of the solutions using the exact model (see FAS in Figure 6.5a), partly due to the fact that only in the first couple of iterations solutions are rejected based



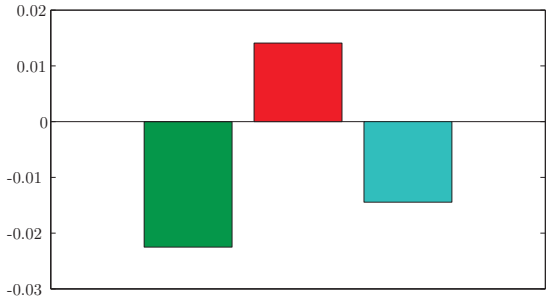
(a) FAS & 1−RNI



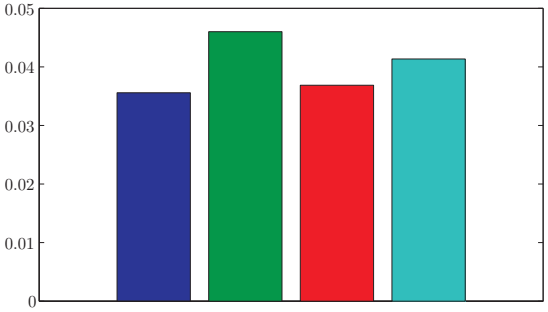
(b) AE' & FAS·AE'



(c) C(AMAN, GA) − C(GA, AMAN)



(d) D(AMAN, GA) − D(GA, AMAN)



(e) $\nabla^{N'}$

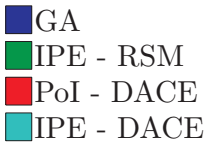


Figure 6.5: Performance Measures for Test Network II

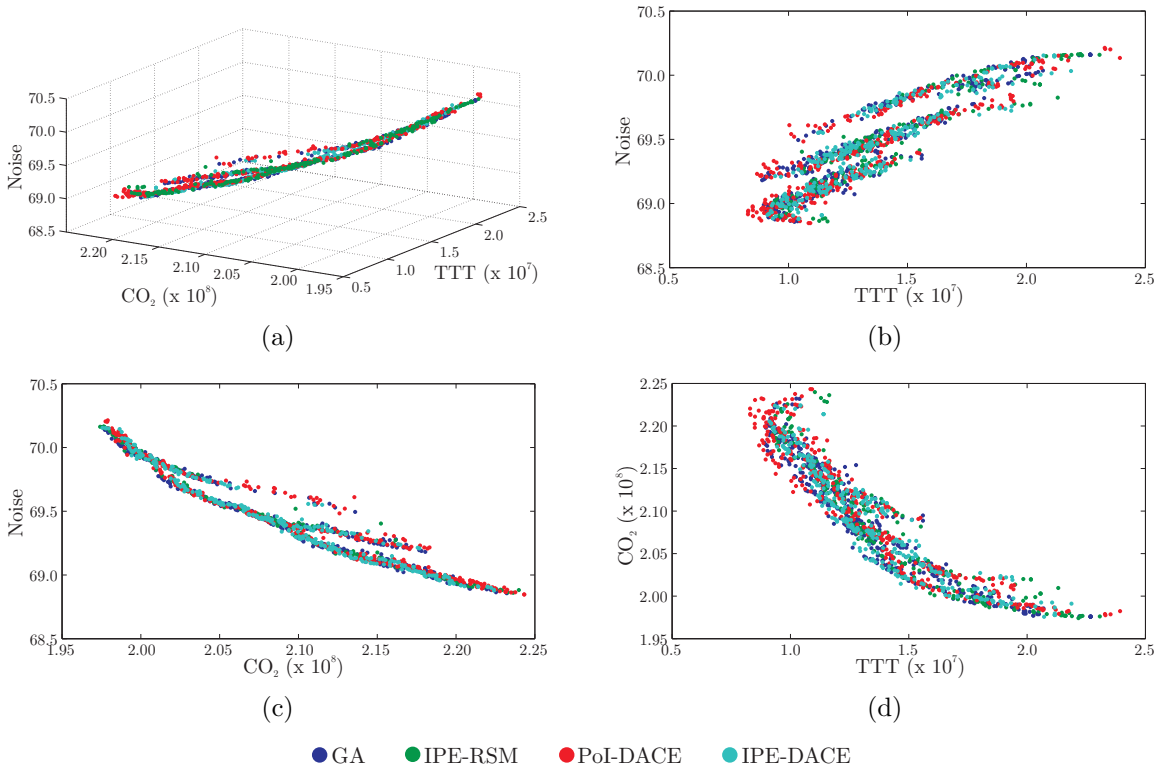


Figure 6.6: Pareto front for Test Network II

on the 'evaluate the best x% solutions' criterion. Furthermore IPE-RSM and IPE-DACE perform more or less equally well by evaluting around 35% of the solutions. Clearly in this test network, IPE-DACE is able to provide a reliable Pareto front.

The fraction of the exactly evaluated solutions that form the Pareto front (RNI) is for all approaches more or less similar. Furthermore we see that the computational effort (AE', see Figure 6.5b) is indeed as expected. Due to the low values of FAS for both IPE-RSM as well as IPE-DACE the resulting total computational effort (FAS $\cdot$ AE') is similar and slightly lower than the computational effort for the PoI-DACE approach. Yet the PoI-DACE approach is still around 50% faster than the original GA.

For the C- and D-metric (Figures 6.5d and 6.5d) we find somewhat contradictorily results. Whilst PoI-DACE is outperformed by the other two AMANs on $C(\text{AMAN}, \text{GA}) - C(\text{GA}, \text{AMAN})$, it outperforms them on $D(\text{AMAN}, \text{GA}) - D(\text{GA}, \text{AMAN})$, which is considered to be the most interesting performance measure This will make it difficult to select the 'best' or probably 'most promising' approach. It is also interesting to find that, where in the first test network the GA was sometimes outperformed by the AMANs, in this case the GA always outperforms the AMANs.

Finally we find that according to $\nabla^{N'}$ in Figure 6.5e, PoI-DACE provides us with results that are evenly spread over the objective space. In fact, PoI-DACE provides results that are comparable to the original GA. It is interesting to see that again IPE-RSM is not the best performing AMAN. Note that the differences between PoI-DACE and GA might very well be significant, as explained in the discussion about the results of Test Network I.

From Figure 6.6 it becomes immediately clear that in this case all AMANs and GA are more evenly spread over the objective space. It should however be noted that (just from sight) it appears that PoI-DACE is more often on the edges of the objective space than other AMANs or the GA.

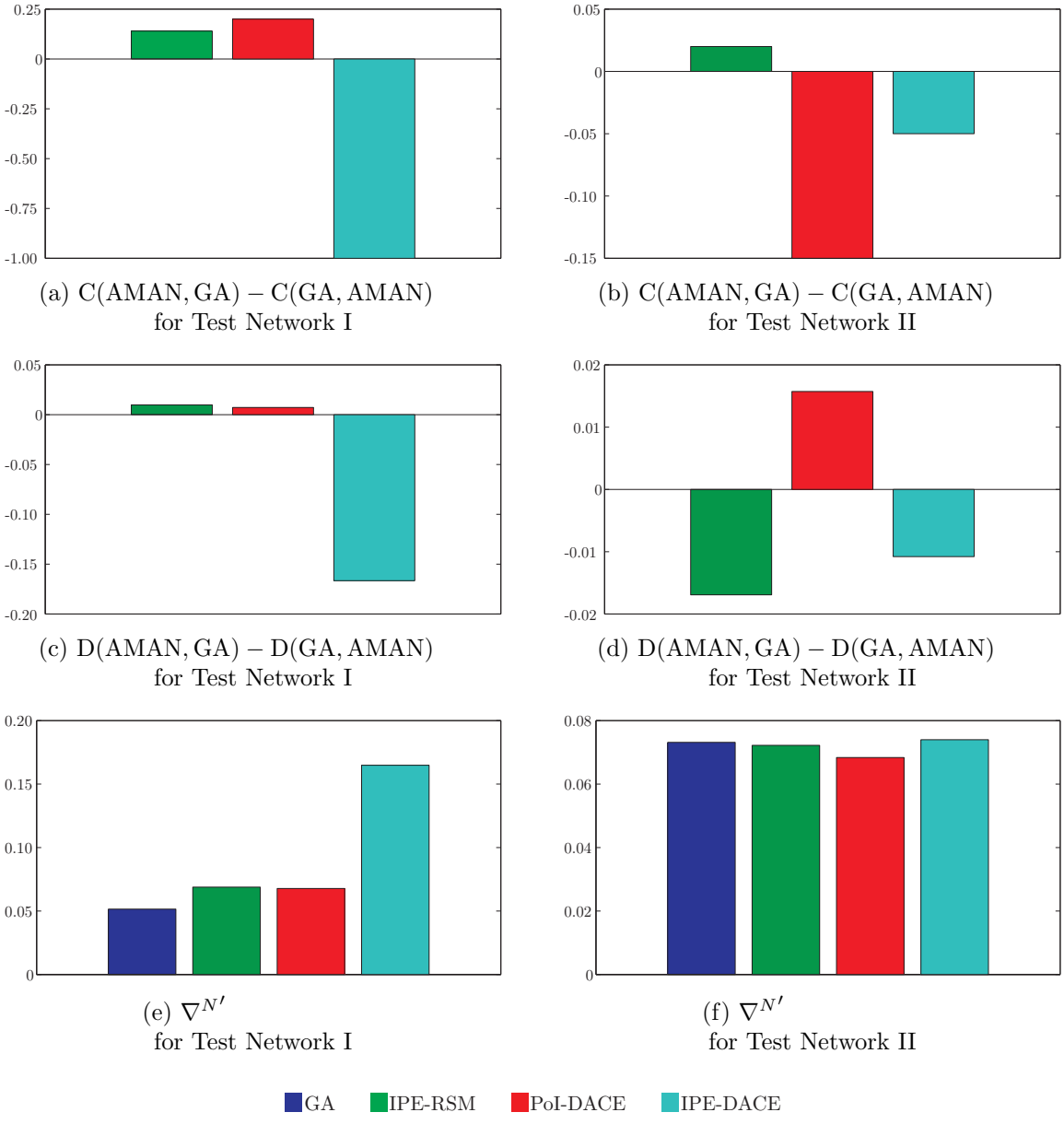
It is also interesting to see that in this test network we again can identify a normative DTM measure. In this case however the DTM measure is probably one with three settings, which can be derived from the three 'strokes' which are clearly visible in Figure 6.6b, 6.6c and 6.6d. This suggests that the Variable Speed Sign is the normative DTM measure.

100 Final Parents

For the 100 final parents we only have to determine the C-Metric, D-Metric and $\nabla^{N'}$ performance measures. The other performance measures (FAS, RNI and AE') are only relevant when comparing the entire process of the AMANs or GA.

For Test Network I (Figure 6.7a, 6.7c and 6.7e) we find more or less the same results as when we use the entire Pareto front (Figure 6.3). Note however that in this case the value for $C(\text{AMAN}, \text{GA}) - C(\text{GA}, \text{AMAN})$ for IPE-RSM is positive, whereas it was negative when using the entire Pareto front. It is interesting to see that the results for the $\nabla^{N'}$ performance measure, which heavily depends on the number of solutions, are more or less the same. What is also interesting to find, is that IPE-DACE scores more or less similar on all the performance measures, whilst the Pareto front consists of only 17 solutions. This means that, in order to obtain these results, we also had to use solutions that are in the second or third front (i.e. solutions that are dominated by solutions in the Pareto and second front respectively).

For Test Network II (Figure 6.7b, 6.7d and 6.7f) the conclusions do not change much as well. For the $C(\text{AMAN}, \text{GA}) - C(\text{GA}, \text{AMAN})$ performance measure it is interesting to see that IPE-RSM now has a positive value, whereas it was negative when the entire Pareto front was considered. This suggests that IPE-RSM finds a cluster of solutions that were dominated by the original GA. Due to the crowding distance algorithm, many of these solutions are removed when applying the non-dominated sorting algorithm (the crowding distance algorithm is part of this algorithm), which could cause the better score on this performance measure. For the $D(\text{AMAN}, \text{GA}) - D(\text{GA}, \text{AMAN})$ performance measure little has changed. Finally we find for the $\nabla^{N'}$ perform-



(a) C(AMAN, GA) − C(GA, AMAN) for Test Network I

(b) C(AMAN, GA) − C(GA, AMAN) for Test Network II

(c) D(AMAN, GA) − D(GA, AMAN) for Test Network I

(d) D(AMAN, GA) − D(GA, AMAN) for Test Network II

(e) $\nabla^{N'}$ for Test Network I

(f) $\nabla^{N'}$ for Test Network II

Figure 6.7: Performance Measures for 100 Final Parents

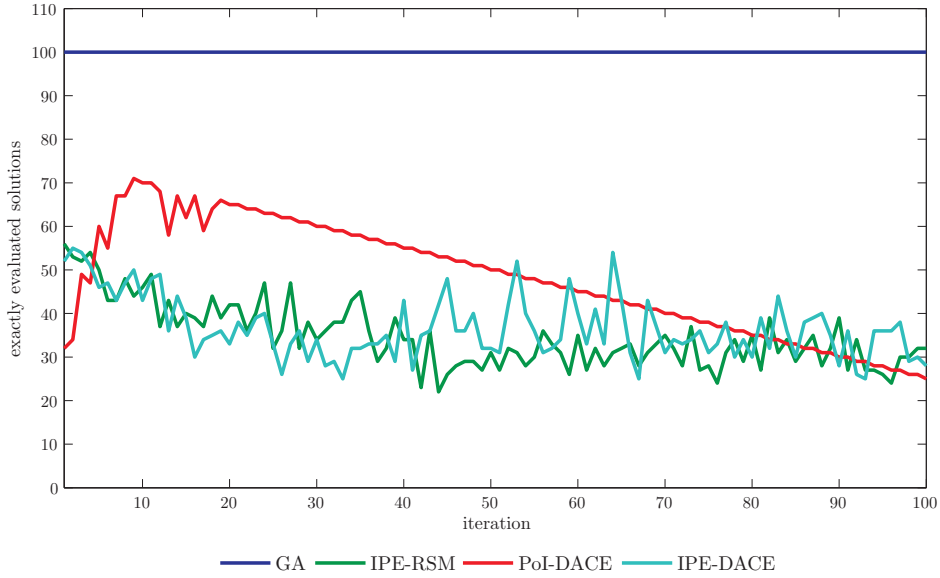


Figure 6.8: Convergence of exactly evaluated solutions

ance measure that GA now performs worst, whereas it scored somewhat in the middle when using the entire Pareto front. This might be caused (again) by the crowding distance algorithm, which selects solutions that are furthest away. In this case this could have caused massive gaps between some of the solutions, thus resulting in a slightly higher $\nabla^{N'}$.

Convergence

Besides the scores on the performance measures it is also interesting to discuss the converging behaviour of the AMANs. Figure 6.8 therefore displays the number of exact function evaluations for each of the iterations in the NSGA-II algorithm. Obviously the original GA (NSGA-II) evaluates 100 solutions in every iteration. What is furthermore interesting to see is that both IPE-based AMANs seem to have an initial phase (iteration 1 – 20) in which the number of solutions they exactly evaluate slowly decreases from 55 to 30 – 40. After this initial phase, both evaluate somewhere between 25 –40 solutions for the remaining iterations, of course with the occasional peek up and down.

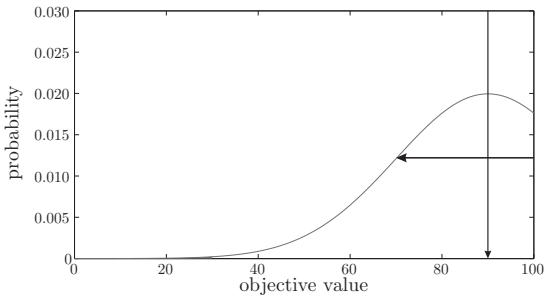
However the truly interesting part is the convergence of the PoI-DACE approach. In the first 20 iterations the selected minimal probability of improvement of 50% appears to do its job, since only a limited number of solutions is exactly evaluated. The most likely explanation for this behaviour is that many of the proposed solutions are quite far behind the Pareto front and thus the probability of them improving the Pareto front is only marginal. In the remainder of the algorithm (iteration 20 – 100) the number of exactly evalu-

ated solutions follows the 'evaluate a maximum of n solutions', which proves the use of this criterion. It does however also indicate that all 8000 solutions that are proposed in the last 80 iterations have a probability of improvement that is larger than 50%.

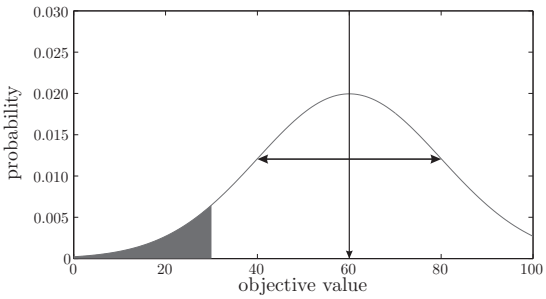
It is this conclusion that needs some contemplation. The Kriging/DACE approximation technique estimates two variables, the expected objective value $E(x)$ and the corresponding kriging error $s(x)$. In order to explain the expected behaviour we will use the example shown in Figure 6.9. In Figure 6.9a we see a possible solution with $E(x) = 90$ and $s(x) = 20$ and a Pareto front which (in this one dimensional example) is located at 30. The probability of this solution improving the Pareto front is marginal, $P(x \leq PF) = 0.0013$. Over time we would expect that the proposed solutions would shift closer to the Pareto front, as is the case in Figure 6.9b. Obviously the probability of improvement increases significantly, which is in accordance with the behaviour that is observed in our test network during the first 20 iterations (see Figure 6.8). However at the same time we would like the kriging error $s(x)$ to decrease, thereby improving the reliability of the expected objective values. An example of this behaviour is shown in Figure 6.9c, this of course causes the probability of improvement to decrease. We do however not perceive this behaviour (reducing $s(x)$ enough to reduce the Probability of Improvement) when looking at the convergence of the PoI-DACE approach in Figure 6.8.

The reason why the convergence of PoI-DACE does not show this behaviour can be found in the roots of the Kriging/DACE approximation technique. The main idea behind Kriging/DACE is that solutions that are close together in solution space are correlated. This concept is used because the known values of the 'learning data' are needed to determine the objective values of the proposed solutions. However reducing the kriging errors requires learning data that is close (in solution space) to the proposed solutions. The current algorithm selects solutions that are non-dominated and far away from other known solutions in objective space as learning data. These solutions are not by definition evenly spread in solution space. Summarised, Kriging/DACE needs learning data that is evenly spread in solution space, whereas the NSGA-II algorithm (which is the heart of the AMANs) selects data that is evenly spread amongst the Pareto front in objective space.

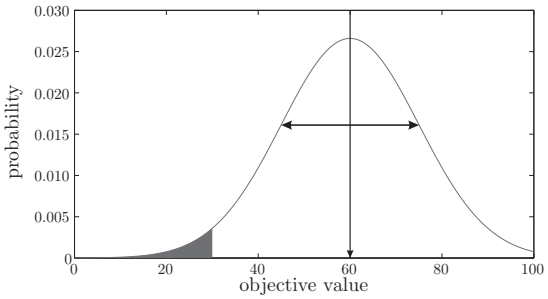
Although it is obvious that the current approach is suboptimal, we will continue to use this approach in this thesis due to time limitations. We do however provide two possible solutions for this problem. The first is to use the concept introduced in SPEA2+ (Zitzler, 2001; M. Kim et al., 2004) which is to use two 'archives' instead of one. Translated to the NSGA-II approach we use, this means that we will still use the non-dominated sorting algorithm, but now use two different crowding distance algorithms. The first is using crowding distance in objective space (as we do now) in order to obtain a set of 'best' parent solutions, the second is using crowding distance in solution space, thereby selecting solutions that are part of the Pareto front, but are scattered



(a)



(b)



(c)

figure	$E(x)$	$s(x)$	$P(x \leq PF)$
a	90	20	0.0013
b	60	20	0.0668
c	60	15	0.0228

(d) parameter values

Figure 6.9: Example of expected behaviour of Probability of Improvement in combination with Kriging/DACE

in solution space. This latter 'archive' can then be used as learning data for the Kriging/DACE approximation function. The second possible solution to the problem is not only using the parents as learning data, but also include for all instances of offspring the known solutions from the database that is closest to it in solution space. This would increase the learning data to the number of parents plus the number of children at most. Due to the fact that we now include for each child a solution that is close to it, we can reduce the kriging error for this child. We did not test either of these solutions, due to the aforementioned time limitations.

6.4 Conclusions

Choosing one AMAN from the three proposed AMANs (IPE-RSM, PoI-DACE and IPE-DACE) is not a very simple task. When looking at their performance on both test networks (Figure 6.3, 6.5 and 6.7) we see that sometimes A outperforms B and sometimes the opposite holds.

The first decision made is to reject the IPE-DACE approach. In Test Network I it became clear that the combination of IPE and DACE can give very unreliable results. Although this might just be the case with this specific

set of start solutions (due to time limitations we were unable to test this approach on multiple instances) it seems too much of a gamble to select an approach which has already failed on a test network.

When comparing IPE-RSM and PoI-DACE, we find that the results are not conclusive. For FAS we find that IPE-RSM outperforms PoI-DACE (see Figure 6.3a and 6.5a), and the same holds for AE' and FAS·AE' (see Figure 6.3b and 6.5b). However PoI-DACE performs better on the $1-$RNI performance measure (see Figure 6.3a and 6.5a).

For C-Metric, as a quite ambiguous performance measure, on Test Network I PoI-DACE is to be preferred (see Figure 6.3c and 6.7a), whereas on Test Network II exactly the opposite (IPE-RSM is to be preferred) holds (see Figure 6.5c and 6.7b).

Looking at the D-Metric performance measure, which is considered to be a good indicator, we find that on Test Network I IPE-RSM and PoI-DACE perform more or less equal (see Figure 6.3d and 6.7c). However on Test Network II PoI-DACE outperforms IPE-RSM significantly (see Figure 6.5d and 6.7d).

Finally we find that looking at the spacing in objective space, PoI-DACE is unanimously chosen as the best performing approach (see Figure 6.3e, 6.5e, 6.7e and 6.7f). Although the differences between PoI-DACE and IPE-RSM appear to be small, we also showed that the meaning of these differences can be very significant when looking at the resulting Pareto fronts.

Combining all these conclusions we might say that from an algorithmic performance perspective IPE-RSM might be preferred if speed is the main objective. When effectiveness is the most interesting PoI-DACE is to be preferred since it has a better performance on $1-$RNI.

However in order to select the definitive approach we have to return to the two criteria that were set to select the different performance measures (see page 94). We stated that the better AMAN is the one that a) maximises the volume of the objective space that is dominated by the Pareto front; and b) maximises the spread over the Pareto front. From the D-Metric we can easily deduct that PoI-DACE outperforms IPE-RSM (or scores more or less equal) on the first criterion. Furthermore we find that from a diversity point of view (criterion two) PoI-DACE is to be preferred, since it outperforms IPE-RSM on both test networks.

The main goal of this thesis is *'accelerating the search for the Pareto optimal set found by multiobjective genetic algorithms for multiobjective network design problems, in which externalities are the objectives and DTM measures the decision variables, using function approximations'*. Because we want to find a comparable Pareto front, we decide to select PoI-DACE which scores best on the dominance and diversity metrics, which measure the quality of the resulting Pareto front. Furthermore it should be noted that whilst IPE-RSM

outperforms PoI-DACE on (some) algorithmic performance measures, PoI-DACE still is able to achieve these results with a calculation time reduction of nearly 50%.

In the next chapter we will describe a more realistic test network and apply both the original Genetic Algorithm and the proposed AMAN which combines the Kriging/DACE approximation method with the Probability of Improvement approach, in order to see whether the same results can be achieved on a (much more) complicated network.

Chapter 7

Testcase Almelo

Results! Why, man, I have gotten a lot of results.
I know several thousand things that won't work.

Thomas A. Edison (1847 – 1931)

The last research question we proposed in section 1.7 focusses on the performance of the selected AMAN when it is applied to a realistic ('real-life') network. In this chapter we will apply the selected AMAN (PoI-DACE) to the case of Almelo. In section 7.1 we will describe the case environment and the DTM measures that are present in this network. Section 7.2 will briefly discuss the methodology.

In section 7.3 the results of both the original NSGA-II algorithm and the AMAN are presented. Besides providing the results for the different performance measures (the ones defined in section 6.2) we will also try to assess the results from a 'traffic and transport engineering' point of view (section 7.4). Finally in section 7.5 we will decide whether the proposed AMAN is indeed capable of presenting a solution that is qualitatively comparable to the solution of NSGA-II.

7.1 Network Description

Almelo is a city with just over 70,000 inhabitants in the eastern part of the Netherlands. Almelo is located close to the A1 motorway (Amsterdam - Hengelo - Germany) and the A35 motorway which continues as N35 (provincial road) in the direction of Zwolle. The N36 on the western side of Almelo functions as a ring road around Almelo towards the 'hinterland'. As such Almelo functions as a crossroad connecting regional roads like the N36, N349 and the N743 to the A35 motorway (Possel, 2009). Figure 7.1 gives an overview of the surroundings of Almelo.

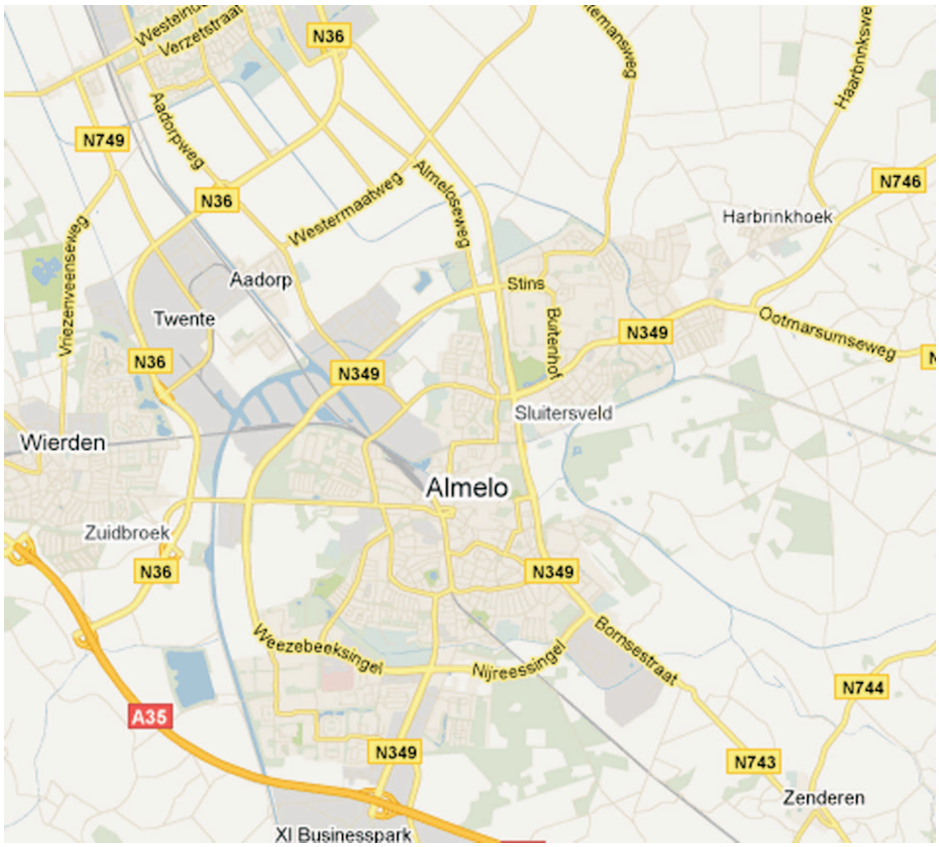


Figure 7.1: Overview of Almelo Area © 2010 Google

For this research we use a road network that is similar to the one used by Possel (2009), but is slightly larger. Where Possel uses the N36 up to Wierden, we expanded the road network so it also includes the on- and of ramps of the N36 near Vriezenveen. Figure 7.2 shows a schematic representation of the road network that is being used. The road network is a simplified representation of the real situation, many of the minor roads are not considered in this test case. Furthermore some minor changes were made to the road network and the corresponding Origin-Destination matrix (OD-matrix). This was done to reduce problems on the links that connect the centroids to the road network and to create results (objective values) that are comparable to the results of the 'verkeersmilieukaart'.[1]

From previous analysis it has become clear that the major traffic problems in the city of Almelo are (during morning peek hours) caused by inbound traffic. One of the major difficulties with these traffic problems is that many

[1]The 'verkeersmilieukaart' is a graphical representation of the environmental effects (e.g. CO_2 emissions, noise, air quality etc.) of traffic, which is developed by Goudappel Coffeng BV, www.verkeersmilieukaart.nl.

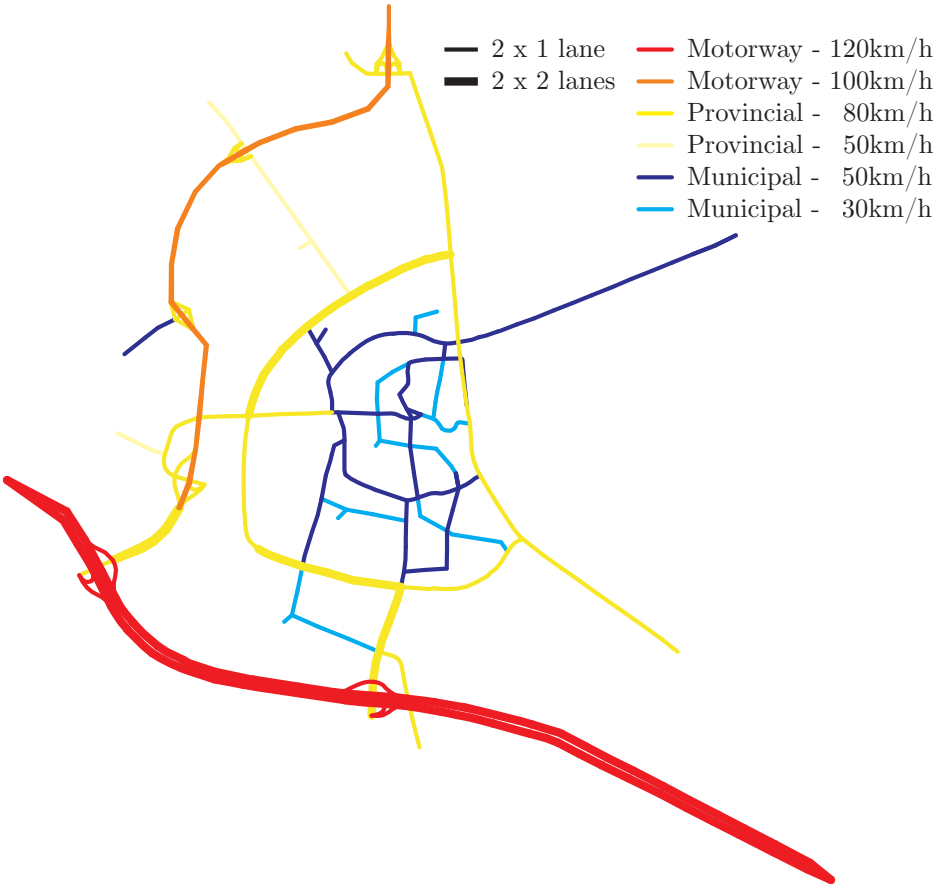


Figure 7.2: Road Network of Almelo

commuters do not have a lot of real alternatives. Of course one can take a alternative route around the block, but that does not change the situation much. We therefore have to resort to intervening at the point were people enter the city centre, i.e. on the ring roads. It is therefore that we decided to implement seven controlled Automated Traffic Control Systems (ATSs) on the ring roads of Almelo. These ATSs are able to direct traffic by giving priority to traffic that either continues on the ring road or turns into the city centre. Furthermore we decided to implement two Variable Speed Signs (VSSs) on the motorway A35, which might cause people to choose for the other off ramp, thereby shifting the traffic to another part of the ring roads. Figure 7.3 shows the seven controlled ATSs, indicated by the circles numbered 1–7, and the two sets of links were the VSSs are applied, represented using the black (continuous/discontinuous) lines numbered 8 and 9.

In Appendix D the outflow capacity for the different settings for each of the ATSs is shown, as well as the car speed limits for each of the VSSs.

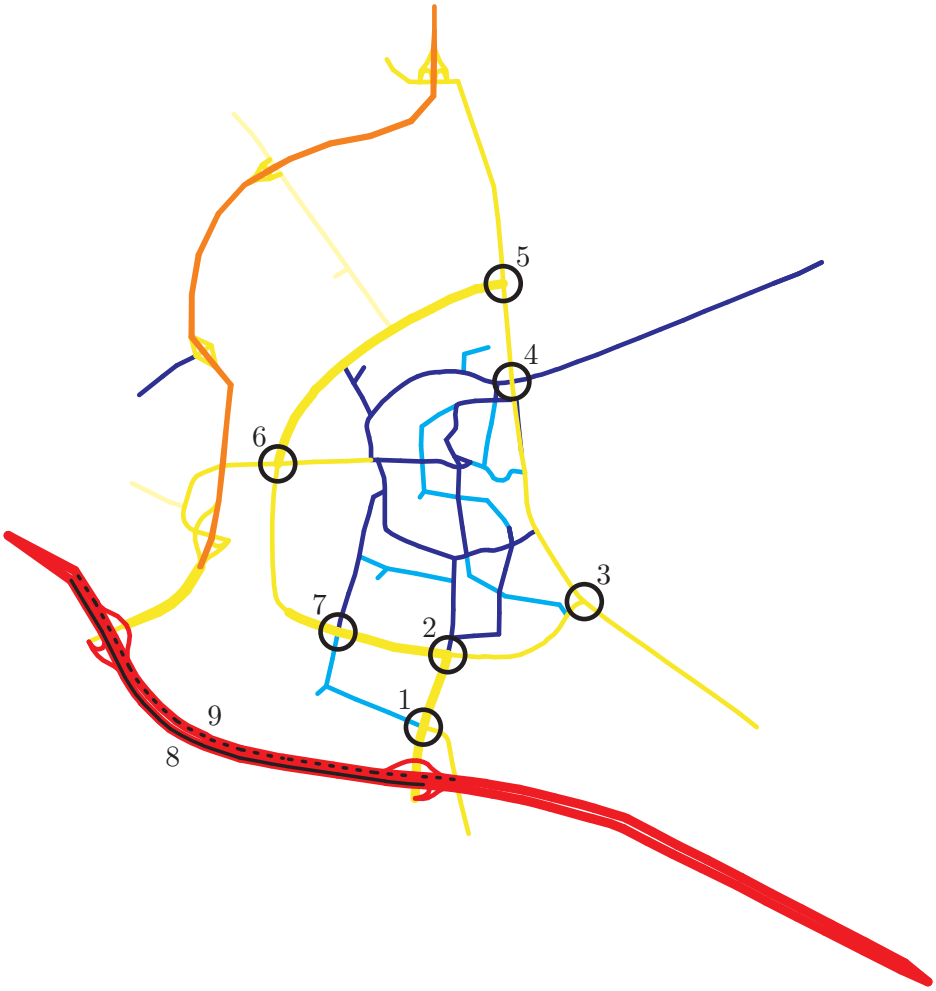


Figure 7.3: DTM measures on the Almelo network

7.2 Methodology

For this analysis we more or less follow the approach described in section 6.1. We therefore start by applying the 'ordinary' NSGA-II algorithm to the Almelo network using a set of 100 start solutions. We perform 50 iterations in which we generate 100 children from 100 parents and we repeat the whole procedure for the AMAN which uses the PoI-DACE approach. Note that we use the same set of start solutions for both approaches.

The cooling scheme for the PoI-DACE approach is based on the 'evaluate the best n solutions' approach discussed before (see section 5.2). We start by exactly evaluating 75 solutions in the first iteration and linearly decrease the number of exactly evaluated solutions to 25 in the last iteration, thereby halving the number of exact evaluations.

In the previous chapter we developed alternative approaches to show the results for the C- and D-Metric (see section 6.3). The main reason for these alternative presentations was that it allowed us to easily compare the differences between the different AMANs. However in this case we do not need to compare different AMANs, but we compare two approaches and check whether the results they produce are comparable. We therefore decided to maintain the original performance measures $C(\text{GA}, \text{AMAN})$, $C(\text{AMAN}, \text{GA})$, $D^N(\text{GA}, \text{AMAN})$ and $D^N(\text{AMAN}, \text{GA})$.

Similarly to what we did in the previous chapter we plot $1 - RNI$ instead of RNI for convenience (in Figure 7.4a lower values are now 'better').

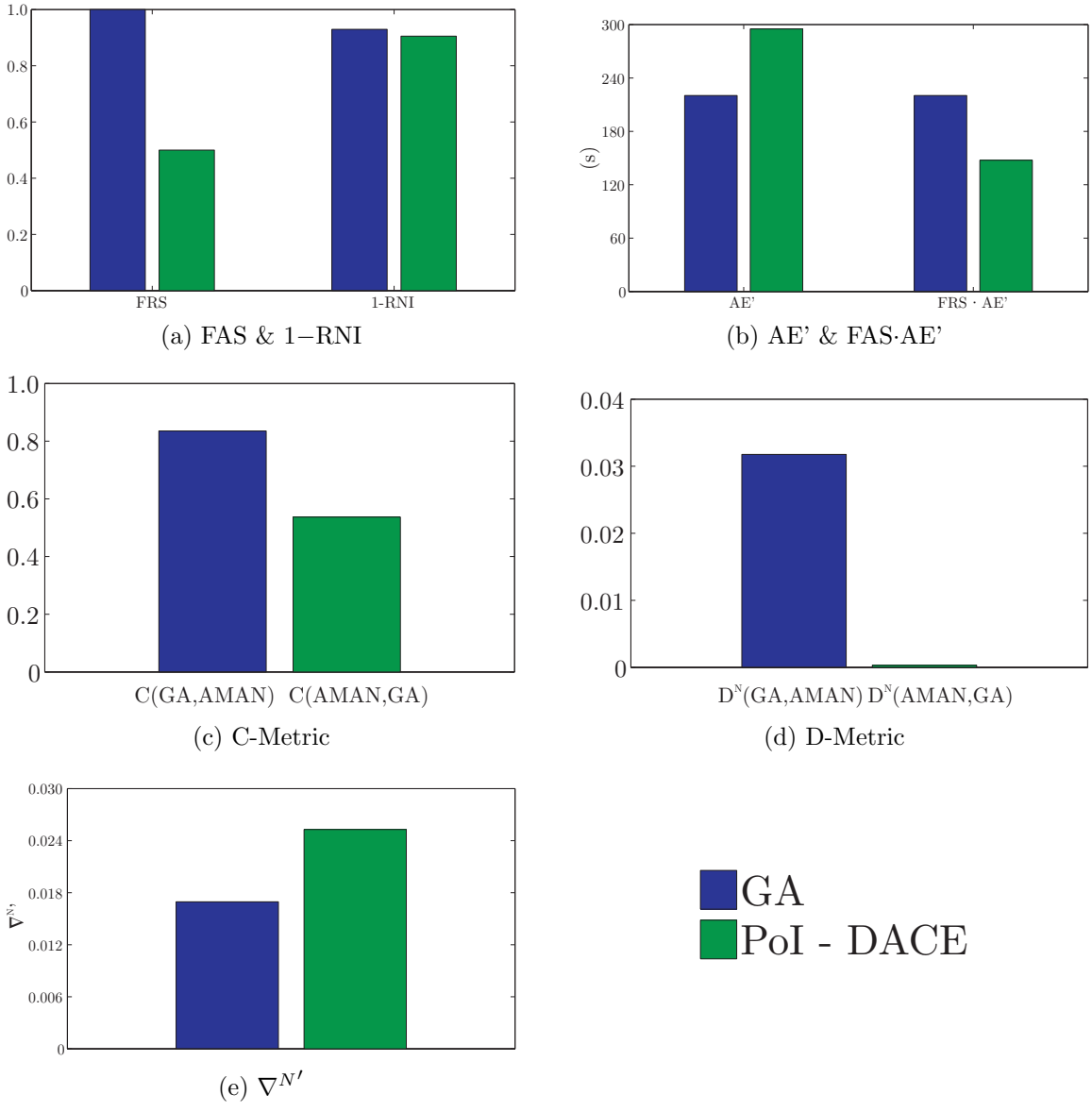
7.3 Results

The results for the FAS, RNI, AE' and FAS $\cdot$ AE' performance measures can hardly be called surprising. In Figure 7.4a we see that the Genetic Algorithm does indeed evaluate all solutions whereas the PoI-DACE approach only evaluates half of them. Furthermore we find that the AMAN outperforms the GA on RNI, which was expected based on the results found on Test Network I and II (Figures 6.3a and 6.5a). We also see that the calculation times per exact evaluation (AE') increased by nearly 35% whilst the total computational effort (FAS $\cdot$ AE') decreased by nearly 35% (Figure 7.4b). This shows that the AMAN is indeed capable of accelerating the search for the Pareto front. The question now is, however, how 'good' the Pareto front is that is found by the AMAN when compared to the Pareto front found by the original GA.

For the C-Metric (Figure 7.4c) we see that the Pareto front found by the Genetic Algorithm dominates the front found by the AMAN significantly, over 80% of the solutions found by the AMAN are (weakly) dominated. What is however perhaps even more interesting is that the Pareto front found by the AMAN dominates over 50% of the solutions found by the GA. A part of this could be explained by the fact that there are solutions that are present in both Pareto fronts, however this is not the case.[2] It is therefore that we have to conclude that this result is entirely due to the stochasticity that is inevitably part of the Genetic Algorithms. Perhaps even more importantly, this also means that if we correct $C(\text{GA}, \text{AMAN})$ it is not unlikely that only 25-30% is being dominated by the GA, the other 50% might very well be due to stochasticity. It is therefore important to put these results into perspective and further research, i.e. an exhaustive study in which dozens if not hundreds of runs are performed, is necessary to be certain about the expected value of $C(\text{GA}, \text{AMAN})$.[3]

[2] It would have been quite a coincidence if we would have found two solutions that are exactly the same, given the size of the solution space $\Theta = \left(9^7 \cdot 3^2\right)^6 = 6.3 \cdot 10^{45}$.

[3] Note that it is possible to do such an extensive study on a much smaller problem, e.g. the test networks used in this research.



(a) FAS & 1−RNI

(b) AE' & FAS·AE'

(c) C-Metric

(d) D-Metric

(e) $\nabla^{N'}$

Figure 7.4: Performance Measures for the Almelo Network

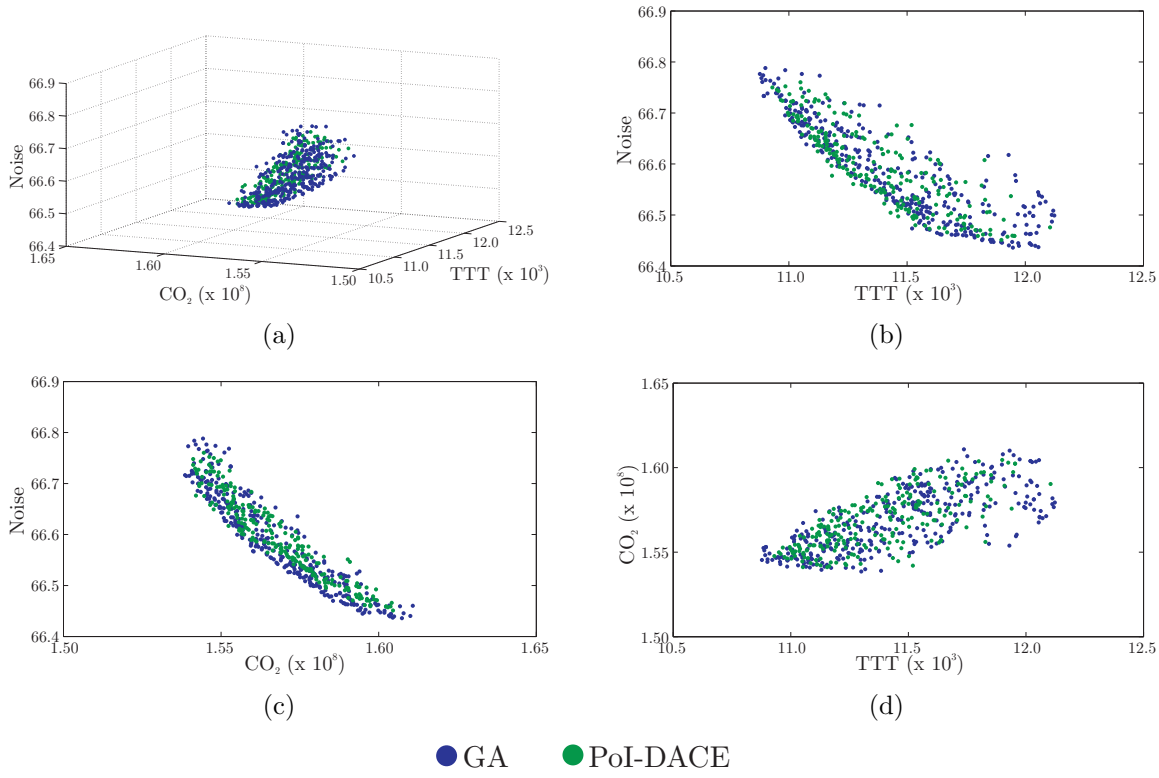


Figure 7.5: Pareto Front for the Almelo Network

For the D-Metric (actually D^N-Metric) which is shown in Figure 7.4d, we find that the unique contribution of the GA is about 3% of the total normalised volume. Compared to the unique contribution of the AMAN, which is only 0.03% and thus 100 times smaller, this seems quite a significant contribution. It is however necessary to put this conclusion in perspective. If we would judge the unique contribution of the GA on its own merits, it is difficult to state that the solution found by the Genetic Algorithm is indeed 'better'. A unique contribution of 3% is not tremendous and if we include the issue of stocasticity again, we cannot conclude that the solution found by the GA is statistically 'better' than the solution found by the AMAN.

Finally for the $\nabla^{N'}$ (Figure 7.4e) we find that the diversity of the AMAN is more even, although again the differences are not quite large. However in this case we can state that we have seen the same result earlier, i.e. on both test networks we saw that PoI-DACE outperforms the original GA on diversity (see Figure 6.3e and 6.5e). It is however important to note that also in these two instances the differences were marginal and as such we can hardly conclude that the AMAN outperforms the GA by definition.

Looking at the plots of the Pareto fronts (Figure 7.5) is probably just as

interesting as determining the performance statistics. We see that the two Pareto fronts seem to come quite close and there are no clear outliers. What is furthermore interesting is that in contrast to the two test networks where we found clear 'strokes' of solutions which indicated a normative DTM measure, we are now unable to distinguish such a DTM measure.

We can however identify a cloud of solutions that are found by the Genetic Algorithm, but appear not to be part of the Pareto front found by the AMAN. This becomes clear in Figure 7.5c where the solutions found by the GA form a clear line that is closer to the origin than the solutions found by the AMAN. These solutions can also be identified in Figures 7.5b and 7.5d although that is more difficult. What is also interesting is that in the projection of CO_2 versus Noise (Figure 7.5c) we clearly see a thick line which describes the relation, whereas for the other two projections are much more cloudy. This suggests that there is a strong relation between CO_2 and Noise, which is more or less independent from TTT.

In conclusion we can state that, based on visual inspection, the Pareto front found by the Genetic Algorithm appears to be slightly 'better' than the one found by the AMAN. It is impossible to determine whether the difference is due to the algorithm used or whether the stochasticity is the main contributor to this (rather small) difference.

7.4 Traffic & Transportation Effects

From a traffic and transportation engineering point of view it is interesting to see whether we can draw some general conclusions about the solutions that form the Pareto front. If we e.g. find that specific solutions are never part of the Pareto front, this could help us understand the network and allow us to develop a generic control policy.

We therefore generated for each individual DTM measure a set of three boxplots, one for each objective function. As input for the boxplots we used the solutions from the Pareto fronts of the GA and AMAN that form the combined Pareto front (358 solutions in total). We grouped the data based on the setting of the DTM measure and the time period. A single solution therefore is part of six bars, one per time period. Figure 7.6 contains an example boxplot for ATS 1 for the TTT objective, appendix E contains the results for all the DTM measures and all the objectives. Based on these boxplots we might be able to identify relations between objectives and the setting of a specific DTM measure. Note that we also need the control settings for the different DTM measures (appendix D) before we can draw conclusions about the effects of specific settings.

For Automated Traffic Control Signal 1 (ATS 1) we find that settings g and h are rarely part of the Pareto front and setting i is never applied.

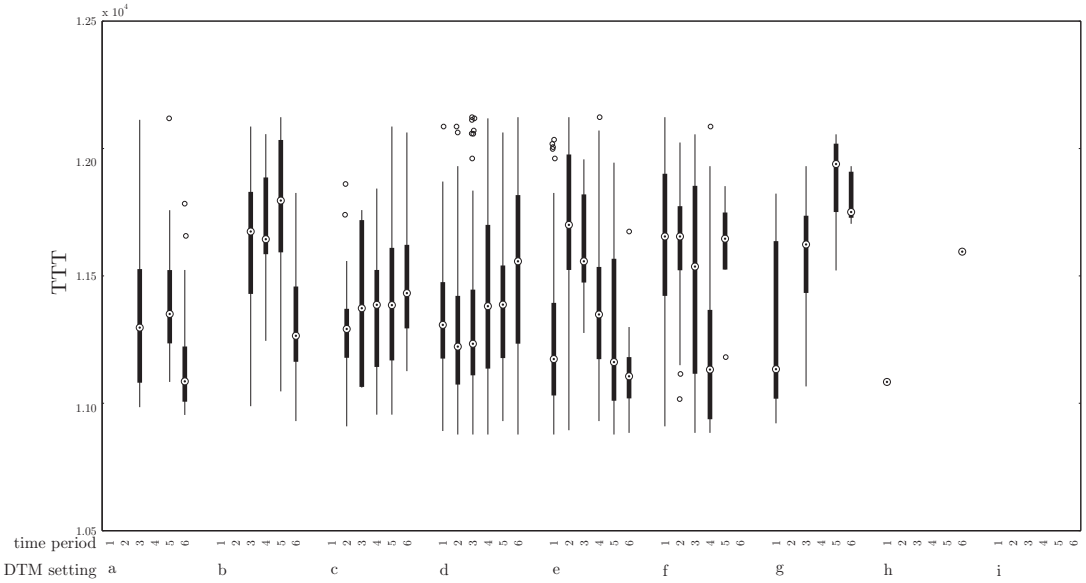


Figure 7.6: Example Boxplot of the Effects of DTM measures ($ATS\ 1-TTT$)

This seems to suggest that settings with a higher outflow capacity for the North-South direction are to be preferred over solutions with a high outflow capacity in the East-West direction. Although the total travel time (TTT) appears to be unaffected by the DTM measure, we also see a positive relation between the CO_2 emissions and 'higher DTM settings', i.e. solutions with a higher outflow capacity reserved for the East-West direction. As expected the relationship between Noise and 'higher DTM settings' is exactly opposite, a noise reduction can be achieved by assigning a larger part of the capacity to the East-West direction. Furthermore it is interesting to note that setting d (in which about $^2/_3$ of the capacity is assigned to the North-South direction) is applied in nearly 46% of the cases, which suggests that this is a robust solution.

For ATS 2 we see that all settings are applied quite often, although setting a is applied in less than 5% of the cases and might therefore be considered a less preferred solution. Furthermore it is difficult to find a clear relationship between specific settings and the results for any of the three objective functions.

For ATS 3 we find that the two 'extreme' settings (a and i) are rarely applied (2.3% and <1% respectively). What is however interesting is that we find a small positive relationship between TTT and 'higher DTM settings', which correspond with solutions that give high priority to traffic from the North. For the CO_2 this relationship is even stronger and obviously the Noise objective has a negative relationship with 'higher DTM settings'. The combination of these results seems to suggest that if one decides to prioritise traffic from the

North, thereby subordinating traffic from the East and South, traffic has to queue (hence larger TTT) which reduces Noise but increases CO_2 emissions. Because TTT and CO_2 emissions on one side and Noise on the other side show opposing behaviour, it is difficult (if not impossible) to state that one solution is 'better' than another. It does provide, however, an instrument that lets principals clearly choose between TTT and CO_2 at one end and Noise on the other.

For ATS 4 it becomes immediately clear that choosing for a solution that strongly limits the outflow from the North and South direction (i.e. less than 600 veh/hr) is not to be preferred, setting g is hardly chosen (<1% of the solutions) and settings h and i are never selected in the Pareto front. Although the spread over the other settings is much more equal, it appears that a solution in which the North-South direction is assigned 1/3 and the East-West direction the remaining 2/3 of the capacity appears to be the most used solution. It is however difficult to find a clear relationship between the settings and the objectives, the most likely explanation is that there are quite a lot of alternatives, that are similar with respect to the objective functions.

For ATS 5 we find results that collaborate with the results from ATS 4. From ATS 4 we can deduce that there is not much traffic in the North-South direction (although there is some traffic, hence the fact that the 'higher DTM settings' are rarely used), and for ATS 5 we find that assigning a large part of the capacity for traffic from the southern direction is not a wise solution (setting a and b are selected once and twice respectively, whereas setting c and d are selected in less than 1.4% and 3.7% of the solutions respectively). All other settings are selected in at least 14% of the solutions and are therefore considered more or less equal. Again it is hard to find a relationship between this DTM measure and the three objectives.

For ATS 6 we find again that solutions that severely limit the traffic flow in North-South direction are hardly selected (setting i once, setting h in only 1% of the solutions). Furthermore we find that giving too much capacity to this direction appears also not to be a solution that is frequently found in the Pareto front, although this happens more often (about 5% of the solutions contain setting a). The other settings are selected between 12.5% and 22.5% of the cases, and are therefore considered to be equal in the sense that they all might contribute to the Pareto front. Solutions in which about 1/3 of the capacity is assigned to the North-South direction and the remaining 2/3 to the East-West direction appear to be commonly found in the Pareto front.

For the last Automated Traffic Control Signal, ATS 7, we find that neglecting the North-South direction appears to be undesirable, setting i is never selected and setting h in less than 4% of the solutions. Although all other solutions are frequently part of the Pareto front, it appears that solutions in which all four directions are assigned a more or less equal portion of the capacity (but a slightly larger portion for the North-South direction, such as setting c and to a lesser extend b) are amongst the 'best', given that over 40%

of the solutions contain setting b or c.

For the first Variable Speed Sign (VSS 8) we see that the behaviour is much more whimsical than the behaviour of the ATSs. The most likely explanation is that this is due to the fact that the VSS has only three settings. From statistical analysis we find that the, in the Netherlands common, speed limit of 120 km/hr is only part of 13% of the solutions, i.e. in most solutions it appears to be best to reduce the speed limit. Furthermore we find a clear negative relation between TTT and the speed limit (which is rather obvious). However we also see a clear negative relation between the speed limit and CO_2 emissions and a positive relation between the speed limit and Noise.

For the VSS 9, the results might be even more interesting. We find that in less than 1% of the solutions the ordinary speed limit of 120 km/hr is maintained. In fact we find that 42% of the solutions contain a 100 km/hr speed limit whereas the majority of the solutions (57%) uses a speed limit of only 80 km/hr. Furthermore we find the same relations between speed and the objective functions as with VSS 8.

Summarised we can state that we have found two classes of conclusions. First we find that some settings are hardly applied in the solutions that form the Pareto front. It would be interesting to see how the Pareto front changes when we reject the solutions that do use these settings, although in reality we probably would have found comparable solutions (i.e. solutions that use a setting that is allowed and is close to the setting we rejected). If we can conclude that the Pareto front does not change that much, we might be able to use these conclusions to determine an 'optimal' set of DTM control settings. We therefore created three projections (Figure 7.7) in which we show the solutions of the combined Pareto optimal sets (blue open circles) and the solutions that remain when we leave out settings which appear in less than 1% of the solutions (called rare settings, represented by green dots). Since the latter is a subset of the former, each green dot has a blue open circle around it. We assumed that these rare settings would produce solutions that were on the edges of the Pareto front, however based on Figure 7.7 we can state that these solutions are the 'best' solutions based on the TTT and CO_2 objectives, given that many of the solutions are in the bottom left corner. If we also include the Noise objective into the equation, the solutions that are rejected appear to be less critical to the Pareto front. However in contrast to what we expected, we cannot state that the contribution of these solutions is marginal. Furthermore the number of rejected solutions is not as large as expected, this is most likely due to the fact that extreme solutions on one DTM measure also require extreme solutions on another DTM measure in order to create a traffic flow that is acceptable (i.e. that is part of the Pareto front).

The second class of conclusions we can draw, consist of the relation between settings (in fact outflow capacities) and the objective functions. In fact we can identify two ATSs and two VSSs for which we can identify a clear relationship

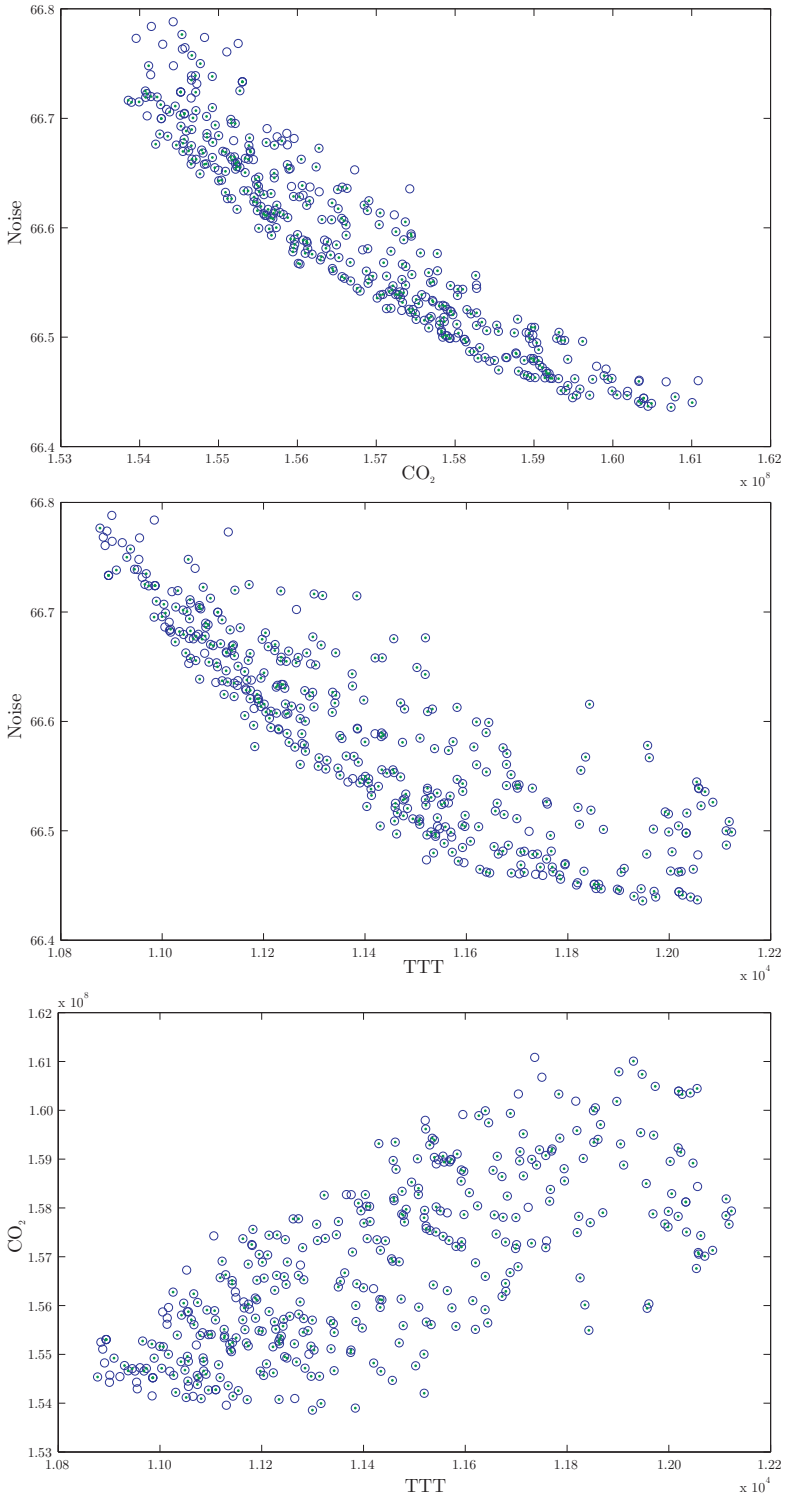


Figure 7.7: Projections of Pareto front with and without rare settings

between the settings and the objective functions. Generally speaking we can say that assigning more capacity to the East-West directions on ATS 1 leads to higher CO_2 emissions and thus reduce noise pollution. Similarly we can conclude that assigning more capacity to the northbound traffic on ATS 3 will increase travel time and CO_2 emissions, whilst reducing the noise pollution. For both ATSs the conclusions are relatively simple. reducing the speed limit will increase travel time and CO_2 emissions but (again and obviously) reduce noise pollution.

A second analysis that was assumed to be of interest was comparing the settings of different DTM measures that are likely to affect each other. We therefore defined three sets of ATSs which we assumed to be strongly interacting. We assumed that the setting of ATS 1 might influence the settings for ATS 2 and ATS 7. Furthermore we assumed that ATS 4 and ATS 5 might be strongly correlated in their 'optimal' settings. However after an extensive data analysis we did not find any clear relationships between those DTM measures. This can suggest that there are no clear relationships, however it is more likely that we have insufficient data to find them. We expect that letting the GA or AMAN run longer (thus finding solutions that are closer to the true Pareto front) might improve the results of this analysis, by creating sharper differences.

In this section we have tried to give an indication of the possibilities of analysing the Pareto front. We can conclude that analysing the Pareto front can lead to valuable information about the generic behaviour of a road network. Unfortunately we are unable to perform a full analysis, partly due to time limitations, but mainly because it falls outside the scope of this research. We do however think that a thorough analysis of a Pareto front (which might include running a specific solution again on a macroscopic model, in order to fully understand what is happening) is a useful exercise and could improve the understanding of traffic on a specific road network.

7.5 Conclusions

In the introduction of this chapter we stated that *'we will decide whether the proposed AMAN is indeed capable of presenting a solution that is qualitatively comparable to the solution of NSGA-II'*. Obviously the AMAN outperforms the Genetic Algorithm on the 'other' performance measures. We do however find no hard evidence that the Pareto front found by the GA fully covers the Pareto front found by the AMAN (C-Metric, Figure 7.4c) in fact we find that the unique contribution of the GA with respect to the AMAN (D-Metric, Figure 7.4d) is only marginal and it is not possible to say that the GA is really 'better' than the AMAN. Furthermore we find that the spread on the Pareto

front found by the AMAN is slightly better than the spread found by the GA ($\nabla^{N\prime}$, Figure 7.4e).

Considering the fact that we have little evidence that shows that the Pareto front found by the GA outperforms the Pareto front found by the AMAN, we assume that the AMAN is able to provide similar results. It should however be noted that further study on a simple network is needed (using dozens or hundreds of runs) to check whether we can statistically show that the results of the AMAN are comparable to the results of the GA.

Chapter 8

Conclusions

A conclusion is the place where you got tired thinking.

Martin H. Fisher (1879 – 1962)

After six months of literature study, testing, programming and data analysis, we are finally able to present the results. We will start this chapter by presenting our final conclusion, which is based on the answers we found to our research questions. Finally we will identify a number of areas in which further research might lead to improved results or a better applicable Approximation Model Assisted NSGA-II (AMAN) algorithm.

8.1 Accelerating the Search for Optimal Dynamic Traffic Management

In the previous four chapters we have tried to find answers to the research questions that were formulated in chapter 1. We therefore start our conclusions by providing a brief answer to each of the research questions, after which we can conclude whether we have achieved our research goal.

Approximation Techniques

The first research question, *'How can the objective values of the bilevel NDP be approximated?'*, focussed on how we can approximate objective values and whether those approximated objective values are reliable. From three main approximation techniques (Response Surface Method; RSM, Radial Basis Function; RBF, Kriging/DACE) we derived 148 different variants using different parameters or objective functions. After applying these variants to two test networks we found that there are a couple of approximation variants that look promising, i.e. they provide reliable estimates for both test networks. Especially the Kriging/DACE approximation technique appeared to provide

reliable results, independent of the objective function (Mardia and Marshall (1984) or Martin and Simpson (2005)). We selected a single variant, Kriging/DACE using the objective function of Mardia and Marshall (1984) and $\phi = 1E^{-7}$ as start value for the optimisation procedure, although we have insufficient evidence that this is indeed the single one best variant. Furthermore we found that the simplest RSM approximation variant that we studied (RSM cubic squared) provided the best results on one test network. Because of the simplicity and intuitiveness of this approach, we decided also to include this approximation method in the remainder of this research.

In conclusion we can state that our research has shown that there are indeed a few approximation techniques that are capable of estimating the objective values of the bilevel NDP, although the differences between different variants can be significant. More information about the methodology and the complete results can be found in chapter 4.

Accelerating Genetic Algorithms

'In what way can the genetic algorithm be accelerated?' is our second research question, which aims at trying to understand how approximated objective values can be used in accelerating Genetic Algorithms. Literature provides us with many different approaches, although most appear not to be suitable for our problem, e.g. because they are too complicated or require even further optimisation. We therefore selected two intuitive approaches, Inexact Pre Evaluation (IPE) and Probability of Improvement (PoI). The former requires only approximated objective values and only evaluates those solutions that are expected to be part of the new Pareto front. The latter approach requires, besides the estimated objective values, estimates of the error of the objective values and determines for each solution the probability that it improves the Pareto front, thereby enabling us to evaluate only solutions with a PoI of at least $x\%$ or the n best solutions.

We can therefore conclude that the easiest way to accelerate a Genetic Algorithm is by reducing the number of solutions that is exactly evaluated. Our research showed that we can apply either a deterministic (IPE) or a stochastic (PoI) approach. More information about those two approaches and other solutions can be found in chapter 5.

Improving the Search for the Pareto Optimal Set

Next we combined the results of the previous two research questions, in order to answer the third research question *'Which improvement methods do indeed improve the search for the Pareto optimal set?'* Combining the two approximation variants and the two acceleration approaches resulted in three Approximation Method Assisted NSGA-II (AMAN) algorithms, one AMAN was rejected because the approximation method could not deliver all information

that is required by the acceleration approach. The three AMANs were applied to the two test networks used before, and their performance was measured using a multitude of performance measures which were derived from literature. Considering that an AMAN can never outperform the original NSGA-II, we can conclude that the stochasticity that is part of the Genetic Algorithm has clouded our results significantly. Unfortunately we do not have sufficient time to perform the dozens (or perhaps even better, hundreds) of runs that are required to make a firm statement about the quality of the results that have been found by the AMAN. We can therefore only draw preliminary conclusions, i.e. the conclusions can be disputed.

The results, however, do suggest that two of the selected AMANs (IPE-RSM and PoI-DACE) provide results that are similar to the results of the original NSGA-II algorithm. It is clear that both approaches accelerate the search for this Pareto front, since the computational effort is reduced with 50 to 60% (PoI-DACE and IPE-RSM respectively). Based on the results we concluded that PoI-DACE was the most promising approach, which is why we selected this approach to be tested on a more realistic road network.

We conclude that the developed AMANs are probably capable of providing a Pareto optimal set (Pareto front) that is comparable to the Pareto front found by the original NSGA-II algorithm, whilst reducing computational effort by at least 50%. We therefore feel confident to state that it is likely that AMANs are capable of improving the search for the Pareto optimal set, although further research is needed before we draw definite conclusions. Detailed results and further analysis can be found in chapter 6.

Testcase Almelo

Finally we tried to find the answer to *'How do these methods cope with realistic networks?'*, which is our last research question. In order to answer this question, we applied the selected AMAN (PoI-DACE) to the slightly adapted network of Almelo. Using the same performance measures that were used for answering the previous research question, we were able to draw some conclusions about the performance of the AMAN compared to the NSGA-II algorithm. We find that PoI-DACE reduces computational effort by around 30-40%, which is somewhat lower than the predicted 50%. This is mainly due to the fact that the Almelo network has much more variables, which increase the calculation times for the optimisation procedure that is part of Kriging/DACE. Although the results of the performance measures seem to suggest that that the original NSGA-II algorithm outperforms the AMAN, we can hardly draw any firm conclusions because of the significant effect of stochasticity. This, again, shows how important it is to do a much more extensive study. From the projections of the Pareto fronts it seems that the NSGA-II algorithm is able to find a Pareto optimal set that forms a layer

around the Pareto optimal set that is found by the AMAN, although the performance measures contradict this conclusion.

It seems that a firm, univocal conclusion cannot be drawn for this research question. We evaluated the results of only a single run (which already took nearly three weeks of calculation time) and the influence of uncertainty (stochasticity) is simply to large. The results however do look promising and we therefore urge the community to do further research in this field.

Final Conclusion

So, have we been successful in *'accelerating the search for the Pareto optimal set found by multiobjective genetic algorithms for multiobjective network design problems, in which externalities are the objectives and DTM measures the decision variables, using function approximations'*?

The answer to this question is not unambiguous. With Jerry R. Ehman[1] we *'choose not to "draw vast conclusions from 'half-vast' data"'.* We can however clearly state that a well selected approximation method is very well capable of accelerating the NSGA-II algorithm. Rejecting all solutions however, can also be considered an acceleration, which is why we needed to focus strongly on the quality of the Pareto front that was found by the AMANs.

Summarising all that has been said above, we conclude the following:

> *'based on our results it seems highly likely that the proposed AMANs can achieve a Pareto front that is comparable to the one found by the original NSGA-II algorithm, whilst reducing the computational effort with 30%'.*

8.2 Further Research

In retrospect we can easily conclude that not everything in this thesis has received the attention that it should actually have had. In this section we therefore identify subjects that require further research, either to improve the quality of the results or to improve our understanding of the problem and/or approach.

As mentioned quite frequently throughout this thesis there still is a lot of uncertainty about the results. For many of our decisions and conclusions we have used only one run of the AMANs and GA, which is statistically unacceptable.

[1] Jerry R. Ehman is an American astronomer who, in 1977, first detected the strong narrowband radio signal known as the 'Wow! signal'. At that time this 'Wow! signal' was considered by many to be proof for extraterrestrial life. Jerry R. Ehman however, came to the conclusion that he could neither accept nor reject the possibility that the signal was of extraterrestrial origin.

In order to provide more credible results at least a couple of dozen, if not a couple of hundred, runs are necessary. It is therefore that we strongly suggest that the different AMANs and the original NSGA-II algorithm are applied dozens of times to a set of test networks, after which statistical techniques can be applied to the performance measures that are defined in this thesis. It is based on those statistical results that we can conclude whether an approach is 'better' than another, but even then the famous saying of Evan Esar (1899 – 1995) *'Statistics: the only science that enables different experts using the same figures to draw different conclusions'* still holds.

Another issue that we would like to bring to notice is that we have examined only a limited number of variants for each of our research questions. We started of by only considering three approximation techniques (chapter 4) and we selected our acceleration possibilities from a limited literature study (chapter 5). As such, it is highly likely that we did not provide a full overview of the possibilities, in fact, providing a full overview is impossible. One could, however, perform an additional literature study in order to expand our overview to create a more comprehensive, and thus better, overview. This would also reduce the risk of not having examined an approach that later appears to be one of the best alternatives available. Furthermore, and perhaps even more importantly, we have examined only a limited number of variants of approximation methods. Although we examined 148 different variants in total, it still is possible that we did not find the best performing alternative. It therefore might be an interesting exercise to see whether one of the variants we consider to be amongst the best, can be fine tuned in order to increase the performance.

In the AMAN we selected we applied the Probability of Improvement (PoI) approach. However we showed that the convergence of this approach, when using the $x\%$ selection criterion, was not in accordance with what we expected. It therefore seems appropriate to perform an in depth analysis of the convergence of the PoI approach. Interesting subjects for this research would be for instance how the approximate objective values and the corresponding errors change over time (i.e. from generation to generation). This information can then be used to make a more scientific decision on which values of x (in $x\%$) should be used, if $x\%$ should be used at all.

The last subject that, in our opinion, has not yet received the attention it deserves, is the analysis of the resulting Pareto fronts. In chapter 7 we started by grouping the data and representing the information in boxplots, which could be used to draw some conclusions about the general effects of the different DTM measures. Furthermore we applied pairwise comparisons to find correlation effects between DTM measures. Unfortunately this approach failed, most likely due to a lack of data (our Pareto front consisted of only 385 solutions). It seems, however, that more could be done to increase our under-

standing of the problem. This research might perhaps be the most important recommendation, since it would greatly expand the possibilities of applying the proposed approach in realistic cases.

Bibliography

Alabas, C., Altiparmak, F. & Dengiz, B. (2002). A comparison of the performance of artificial inteliggence techniques for optimizing the number of kanbans. *Journal of the Operational Research Society*, *53*, 907–914.

Allen, T., Yu, L. & Bernshteyn, M. (2000). Low-cost response surface methods applied to the design of plastic fasteners. *Quality Engineering*, *12*(4), 583–591.

Arostegui, M. A., Jr., Kadipasaoglu, S. N. & Khumawala, B. M. (2006). An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics*, *103*, 742–754.

Ben-Israel, A. & Greville, T. N. E. (2003). *Generalized inverses: theory and applications* (2nd ed.). New York, NY, USA: Springer-Verlag.

Bhattacharya, M. (2008). Meta model based EA for complex optimization. *International Journal of Computational Intelligence*, *4*(1), 36–45.

Billings, S. A. & Zheng, G. L. (1995). Radial basis function network configuration using genetic algorithms. *Neural Networks*, *8*(6), 877–890.

Box, G. E. P., Hunter, W. G. & Hunter, J. S. (2005). *Statistics for experimenters: Design, innovation, and discovery* (2nd ed.). Chichester, UK: John Wiley & Sons.

Box, G. E. P. & Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *13*(1), 1–45.

Boyce, D. E. & Janson, B. N. (1980). A discrete transportation network design problem with combined trip distribution and assignment. *Transportation Research Part B: Methodological*, *14*(1–2), 147–154.

Braun, T. D., Siegel, H. J., Beck, N., Bölöni, L. L., Maheswaran, M., Reuther, A. I. et al. (2001). A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*, *61*, 810–837.

Büche, D., Schraudolph, N. N. & Koumoutsakos, P. (2005). Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, *35*(2), 183–194.

Bucher, C. G. & Bourgund, U. (1990). A fast and efficient response surface appraoch for structural reliability problems. *Structural Safety*, *7*, 57–66.

Cantarella, G. E., Pavone, G. & Vitetta, A. (2006). Heuristics for urban road network design: Lane layout and signal settings. *European Journal of Operational Research*, *175*, 1682–1695.

Cantarella, G. E. & Vitetta, A. (2006). The multi-criteria road network design problem in an urban area. *Transportation*, *33*, 567–588.

Chakroborty, P. (2003). Genetic algorithms for optimal urban transit network design. *Computer-Aided Civil and Infrastructure Engineering*, *18*, 184–200.

Chen, A., Kim, J., Lee, S. & Kim, Y. (2010). Stochastic multi-objective models for network design problem. *Expert Systems with Applications*, *37*, 1608-1619.

Chiou, S.-W. (2005). Bilevel programming for the continuous transport network design problem. *Transportation Research Part B: Methodological*, *39*, 361–383.

Conn, A. R., Gould, N. I. M. & Tointe, P. L. (2000). *Trust-region methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Corne, D. W., Jerram, N., Knowles, J. D. & Oates, M. J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO* (pp. 283–290). Morgan Kaufmann.

Corne, D. W., Knowles, J. D. & Oates, M. J. (2000). The Pareto Envelope-based Selection Algorithm for multiobjective optimisation. In M. Schoenauer et al. (Eds.), *Parallel Problem Solving from Nature – PPSN VI* (pp. 839–848). Berlin, Germany: Springer Berlin / Heidelberg.

Coulomb, J.-L., Kobetski, A., Costa, M. C., Maréchal, Y. & Jönsson, U. (2003). Comparison of radial basis function approximation techniques. *International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, *22*(3), 616–629.

Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, *22*(3), 239–252.

Deb, K. (2001). *Multiobjective optimization using evolutionary algorithms*. Chichester, UK: John Wiley & Sons.

Deb, K., Agarwal, S., Pratap, A. & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al. (Eds.), *Parallel Problem Solving from Nature – PPSN VI* (pp. 849–858). Berlin, Germany: Springer Berlin / Heidelberg.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Dellino, G., Lino, P., Meloni, C. & Rizzo, A. (2009). Kriging metamodel management in the design optimization of a CNG injection system. *Mathematics and Computers in Simulation*, *79*, 2345–2360.

Den Hartog, D., Kleijnen, J. P. C. & Siem, A. Y. D. (2006). The correct kriging variance estimated by bootstrapping. *The Journal of the Operational Research Society*, *57*(4), 400–409.

Dietrich, C. R. & Osborne, M. R. (1991). Estimation of covariance parameters in kriging via restricted maximum likelihood. *Mathematical Geology*, *23*(1), 119–135.

Drezner, Z. & Wesolowsky, G. O. (2003). Network design: selection and design of links and facility location. *Transportation Research Part A: Policy and Practice*, *37*, 241–256.

El-Beltagy, M. A., Nair, P. B. & Keane, A. J. (1999). Metamodeling techniques for

evolutionary optimization of computationally expensive problems: promises and limitations. In W. Banshaf et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO* (pp. 196–203). San Francisco, CA, USA: Morgan Kaufmann.

Emmerich, M. T. M., Giannakoglou, K. C. & Naujoks, B. (2006). Single- and multiobjective evolutionary optimization assisted by Gaussian random fields metamodels. *IEEE Transactions on Evolutionary Computation*, *10*(4), 421–439.

Friesz, T. L., Anandalingam, G., Mehta, N. J., Nam, K., Shah, S. J. & Tobin, R. L. (1993). The multiobjective equilibrium network design problem revisited: A simulated annealing approach. *European Journal of Operational Research*, *65*, 44–57.

Gano, S. E., Renaud, J. E., Martin, J. D. & Simpson, T. W. (2006). Update strategies for kriging models used in variable fidelity optimization. *Structural and Multidisciplinary Optimization*, *32*, 287–298.

Gao, Z., Wu, J. & Sun, H. (2005). Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, *39*, 479–495.

Gen, M., Cheng, R. & Oren, S. S. (2001). Network design techniques using adapted genetic algorithms. *Advances in Engineering Software*, *32*, 731–744.

Gen, M., Kumar, A. & Kim, J. R. (2005). Recent network design techniques using evolutionary algorithms. *International Journal of Production Economics*, *98*, 251–261.

Georgopoulou, C. A. & Giannakoglou, K. C. (2009). Multiobjective metamodel-assisted memetic algorithms. In C.-K. Goh, Y.-S. Ong & K. C. Tan (Eds.), *Multi-objective memetic algorithms* (pp. 153–181). Berlin, Germany: Springer Berlin / Heidelberg.

Grosan, C., Oltean, M. & Dumitrescu, D. (2003). Performance metrics for multi-objective optimization evolutionary algorithms. In *Proceedings of the Conference on Applied and Industrial Mathematics.* Oradea, Romania.

Gutmann, H.-M. (2001). A radial basis function method for global optimization. *Journal of Global Optimization*, *19*, 201–227.

Holland, J. H. (1975). *Adaption in natural and artficial systems.* Ann Arbor, MI, USA: University of Michigan Press.

Holmström, K., Quttineh, N.-H. & Edvall, M. M. (2008). An adaptive radial basis algorithm (ARBF) for expensive black-box mixed-integer constrained global optimization. *Journal of Optimization and Engineering*, *9*, 311–339.

Hussain, M. F., Barton, R. R. & Joshi, S. B. (2002). Metamodeling: Radial basis functions, versus polynomials. *European Journal of Operational Research*, *138*, 142–154.

Jeong, S., Obayashi, S. & Yamamoto, K. (2006). A kriging-based probabilistic optimization method with an adaptive search region. *Engineering Optimization*, *38*(5), 541–555.

Jin, R., Chen, W. & Simpson, T. W. (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, *23*, 1–13.

Jin, Y. (2005). A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, *9*(1), 3–12.

Jin, Y., Hüsken, M. & Sendhoff, B. (2003). Quality measures for approximate models in evolutionary computation. In A. M. Barry (Ed.), *Proceedings of the Bird*

of a Feather Workshop, Genetic and Evolutionary Computation Conference, GECCO (pp. 170–173).

Johnson, D. S., Lenstra, J. K. & Rinnooy Kan, A. H. G. (1978). The complexity of the network design problem. *Networks*, *8*, 279–285.

Jones, D. F., Mirrazavi, S. K. & Tamiz, M. (2002). Multi-objective meta-heursitics: An overview of the current state-of-the-art. *European Journal of Operational Research*, *137*, 1–9.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, *21*, 345–383.

Jones, D. R., Schonlau, M. & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, *13*, 455–492.

Journel, A. G. & E., R. M. (1989). When do we need a trend model in kriging? *Mathematical Geology*, *21*(7), 715–739.

Kannan, S., Slochanal, S. M. R. & Padhy, N. P. (2005). Application and comparison of metaheuristic techniques to generation expansion planning problem. *IEEE Transactions on Power Systems*, *20*(1), 466–475.

Karakasis, M. K. & Giannakoglou, K. C. (2006). On the use of metamodel-assisted, multi-objective evolutionary algorithms. *Engineering Optimization*, *38*(8), 941–957.

Karakasis, M. K., Koubogiannis, D. G. & Giannakoglou, K. C. (2007). Hierarchical distributed metamodel-assisted evolutionary algorithms in shape optimization. *International Journal for Numerical Methods in Fluids*, *53*, 455-469.

Keane, A. J. (2006). Statistical improvement criteria for use in multiobjective design optimization. *AIAA Journal*, *44*(4), 879–891.

Khoo, L. P. & Chen, C. H. (2001). Integration of response surface methodology with genetic algorithms. *International Journal of Advanced Manufacturing Technology*, *18*, 483–489.

Kim, C., Wang, S. & Choi, K. K. (2005). Efficient response surface modeling by using moving least-squares method and sensitivity. *AIAA Journal*, *43*(11), 2404–2411.

Kim, M., Hiroyasu, T., Miki, M. & Watanabe, S. (2004). SPEA2+: Improving the performance of the Strength Pareto Evolutionary Algorithm 2. In X. Yao et al. (Eds.), *Parallel Problem Solving from Nature – PPSN VIII* (pp. 742–751). Berlin, Germany: Springer Berlin / Heidelberg.

Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, *192*, 707–716.

Kleijnen, J. P. C., Van Beers, W. & Van Nieuwenhuyse, I. (2010). Constrained optimization in expensive simulation: Novel approach. *European Journal of Operational Research*, *202*, 164–174.

Knowles, J. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, *10*(1), 50–66.

Knowles, J. & Nakayama, H. (2008). Meta-modeling in multiobjective optimization. In J. Branke, K. Deb, K. Miettinen & R. Sł owiński (Eds.), *Multiobjective optimization* (pp. 245–284). Berlin, Germany: Springer Berlin / Heidelberg.

Knowles, J. D. & Corne, D. W. (2000a). Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, *8*(2), 149–172.

Knowles, J. D. & Corne, D. W. (2000b). M-PAES: A memetic algorithm for mul-

tiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation* (pp. 325–332).

Lau, H. C. W., Ho, G. T. S., Cheng, E. N. M., Ning, A. & Lee, C. K. M. (2007). Benchmarking of optimisation techniques based on genetic algorithms, tabu search and simulated annealing. *International Journal of Computer Applications in Technology*, *28*(2–3), 209–219.

Law, A. M. & Kelton, W. D. (2000). *Simulation modeling and analysis* (3rd ed.). Singapore: McGraw-Hill.

LeBlanc, L. J. & Abdulaal, M. (1978). An efficient dual approach to the urban road network design problem. *Computers & Mathematics with Applications*, *5*, 11–19.

Li, G., Li, M., Azarm, S., Al Hashimi, S., Al Ameri, T. & Al Qasas, N. (2009). Improving multi-objective genetic algorithms with adaptive design of experiments and online metamodeling. *Structural and Multidisciplinary Optimization*, *37*(5), 447–461.

Lim, D., Ong, Y.-S., Jin, Y. & Sendhoff, B. (2007). A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO* (pp. 1288–1295). London, UK.

Lophaven, S. N., Nielsen, H. B. & Søndergaard, J. (2002). *DACE - A Matlab Kriging Toolbox* (IMM-REP-2002-12). Technical University of Denmark.

Mardia, K. V. & Marshall, R. J. (1984). Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, *71*(1), 135–146.

Martin, J. D. & Simpson, T. W. (2005). Use of kriging models to approximate deterministic computer models. *AIAA Journal*, *43*(4), 853–863.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, *58*, 1246–1266.

Mathew, T. V. & Sharma, S. (2009). Capacity expansion problem for large urban transportation networks. *Journal of Transportation Engineering*, *135*(7), 406–415.

McKay, M. D., Beckman, R. J. & Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*(2), 239–245.

Mehnen, J., Michelitsch, T., Lasarczyk, C. & Bartz-Beielstein, T. (2007). Multiobjective evolutionary design of mold temperature control using DACE for parameter optimization. *International Journal of Applied Electromagnetics and Mechanics*, *25*, 661–667.

Meng, Q., Yang, H. & Bell, M. G. H. (2001). An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transportation Research Part B: Methodological*, *35*, 83–105.

Messac, A. & Mullur, A. A. (2008). A computationally efficient metamodeling approach for expensive multiobjective optimization. *Optimization and Engineering*, *9*(1), 37–67.

Messmer, A. & Papageorgiou, M. (1990). METANET: a macroscopic simulation program for motorway networks. *Traffic Engineering & Control*, *31*(8–9), 466–470.

Ministry of Housing, Spatial Planning and the Environment. (2002). *Reken- en meetvoorschift wegverkeerslawaai.*

Mitchell, M. (1996). *An introduction to genetic algorithms.* Cambridge, MA, USA: The MIT Press.

Moody, J. E. & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, *1*, 281–294.

Mullur, A. A. & Messac, A. (2005). Extended radial basis functions: More flexible and effective metamodeling. *AIAA Journal*, *43*(6).

Mullur, A. A. & Messac, A. (2006). Metamodeling using extended radial basis functions: a comparative approach. *Engineering with Computers*, *21*, 203–217.

Noman, N. & Iba, H. (2008). Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation*, *12*(1), 107–125.

Okabe, T., Jin, Y. & Sendhoff, B. (2003). A critical survey of performance indices for multi-objective optimisation. In *Proceedings of the congress on evolutionary computation* (Vol. 2, pp. 878–885).

Ölvander, J. (2005). Robustness considerations in multi-objective optimal design. *Journal of Engineering Design*, *16*(5), 511-523.

Ortúzar, J. d. D. & Willumsen, L. G. (2001). *Modelling transport*. Chichester, UK: John Wiley & Sons.

Payne, H. J. (1971). Models of freeway traffic and control. In G. A. Bekey (Ed.), *Mathematical models of public systems* (Vol. 1, pp. 51–61). La Jolia, CA, USA.

Pham, D. T. & Karaboga, D. (2000). *Intelligent optimisation techniques: Genetic algorithms, tabu search, simulated annealing and neural networks*. London, UK: Springer-Verlag.

Poorzahedy, H. & Rouhani, O. M. (2007). Hybrid meta-heuristic algorithms for solving network design problem. *European Journal of Operational Research*, *182*, 578–296.

Poorzahedy, H. & Turnquist, M. A. (1982). Approximate algorithms for the discrete network design problem. *Transportation Research Part B: Methodological*, *16*(1), 45–55.

Possel, B. (2009). *A genetic algorithm framework and simulated annealing framework for the multi-objective bi-level discrete network design problem*. Msc thesis.

Praveen, C. & Duvigneau, R. (2009). Low cost PSO using metamodels and inexact pre-evaluation: Application to aerodynamic shape design. *Computer Methods in Applied Mechanics and Engineering*, *198*, 1087–1096.

Ratle, A. (2001). Kriging as a surrogate fitness landscape in evolutionary optimization. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *15*, 37–49.

Sacks, J., Welch, W. J., Mitchell, T. J. & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, *4*(4), 409–435.

Sasena, M. J., Papalambros, P. & Goovaerts, P. (2002). Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, *34*(3), 263–278.

Schmöcker, J.-D., Ahuja, S. & Bell, M. G. H. (2008). Multi-objective signal control of urban junctions — framework and a London case study. *Transportation Research Part C: Emerging Technologies*, *16*, 454–470.

Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization*. Msc thesis.

Sharma, S., Ukkusuri, S. V. & Mathew, T. V. (2009). A pareto optimal multi-objective optimization for the robust transportation network design problem. In *Proceedings of the* 87^{th} *Annual Meeting of the Transportation Research Board*

(TRB). Washington, DC, USA.

Simpson, T. W., Peplinski, J. D., Koch, P. N. & Allen, J. K. (2001). Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, *17*, 129–150.

Srinivas, N. & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Journal of Evolutionary Computation*, *2*(3), 221–248.

Stevens, S. S. (1946). On the theory of scales of measurement. *Science*, *103*, 677–680.

Sumalee, A., Shepherd, S. & May, A. (2009). Road user charging design: dealing with multi-objectives and constraints. *Transportation*, *36*, 167–186.

Tan, K. C., Khor, E. F. & Lee, T. H. (2005). *Multiobjective evolutionary algorithms and applications.* London, UK: Springer-Verlag.

Tan, K. C., Lee, T. H. & Khor, E. F. (2001). Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. In *Proceedings of the IEEE Congress on Evolutionary Computation.* Seoul, Korea.

Tanaka, M., Mizoguchi, M. & Takami, I. (2007). Improvement of search performance in genetic algorithms with fitness prediction. *Electrical Engineering in Japan*, *158*(1), 60–68.

The Mathworks Inc. (2010). *Optimization toolbox^{TM}5: User's guide.* The Mathworks Inc.

Van Aerde, M. (1995). A single regime speed-flow-density relationship for freeways and arterials. In *Proceedings of the* 74^{th} *annual meeting of the transportation research board (TRB).* Washington, DC, USA.

Verschuren, P. & Doorewaard, H. (2005). *Het ontwerpen van een onderzoek* [Designing research] (3rd ed.). The Hague, The Netherlands: LEMMA.

Wardrop, J. G. (1952). Some theretical aspects of road traffic research. *Institute of Civil Engineers Proceedings: Engineering Divisions*, *1*(3), 325–362.

While, L., Hingston, P., Barone, L. & Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, *10*(1), 29 – 38.

Wismans, L. J. J., Van Berkum, E. C. & Bliemer, M. C. J. (2009). Multi objective optimization of traffic systems using dynamic traffic management measures. In *Extended abstracts of the International Conference on Models and Technologies for Intelligent Transportation Systems.* Rome, Italy.

Wismans, L. J. J., Van Berkum, E. C. & Bliemer, M. C. J. (2010). Multi-objective network design probem: minimizing externalities using dynamic traffic management measures. In *Extended abstracts of TRISTAN V* (pp. 808–811). Tromsø, Norway.

Wismans, L. J. J., Van Berkum, E. C. & Bliemer, M. C. J. (2011). Comparison of multiobjective evolutionary algorithms for optimization of externalities using dynamic traffic management measures. In 90^{th} *Annual Meeting of the Transportation Research Board (TRB) (submitted).* Washington, D.C., USA.

Xu, T., Wei, H. & Wang, Z.-D. (2009). Study on continuous network design problem using simulated annealing and genetic algorithm. *Expert Systems with Applications*, *36*, 2735–2741.

Yang, B. S., Yeun, Y.-S. & Ruy, W.-S. (2002). Managing approximation models in multiobjective optimization. *Structural and Multidisciplinary Optimization*, *24*, 141–156.

Youssef, H., Sait, S. M. & Adiche, H. (2001). Evolutionary algorithms, simulated

annealing and tabu search: a comparative study. *Engineering Applications of Artificial Intelligence*, *14*, 167-181.

Yunker, J. M. & Tew, J. D. (1994). Simulation optimization by genetic search. *Mathematics and Computers in Simulation*, *37*, 17–28.

Zhang, G. & Lu, J. (2007). Genetic algorithm for continuous network design problem. *Journal of Transportation Systems, Engineering and Information Technology*, *7*(1), 101–105.

Zhou, Z., Ong, Y.-S., Lim, M. H. & Lee, B. S. (2007). Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Computing*, *11*(10), 957–971.

Zhou, Z., Ong, Y.-S., Nair, P. B., Keane, A. J. & Lum, K. Y. (2007). Combining global and local surrogate models to accelerate evolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, *37*(1), 2007.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Unpublished doctoral dissertation.

Zitzler, E. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization. In *Proceedings of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, EUROGEN 2001* (pp. 95–100).

Zitzler, E. & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms: a comparative case study. In A. E. Eiben, T. Back, M. Schoenauer & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V.* Berlin, Germany: Springer-Verlag.

Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271.

Appendices

Appendix A

DTM Control Settings Test Network I

This appendix contains a complete overview of Dynamic Traffic Management control settings for Test Network I.

outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
500	500	80	60	1
600	600	80	60	1
700	700	80	60	1
800	800	80	60	1
900	900	80	60	1
1000	1000	80	60	1
1100	1100	80	60	1
1200	1200	80	60	1
1300	1300	80	60	1
1400	1400	80	60	1
1500	1500	80	60	1

(a) ATS - Regional Road to Regional Road

outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
500	500	80	60	1
600	600	80	60	1
700	700	80	60	1
800	800	80	60	1
900	900	80	60	1
1000	1000	80	60	1
1100	1100	80	60	1
1200	1200	80	60	1
1300	1300	80	60	1
1400	1400	80	60	1
1500	1500	80	60	1

(b) ATS - Regional Road to Local Road

Table A.1: DTM control settings for Test Network I

Car				
outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
2200	2200	120	80	2
2250	2250	100	80	2
2300	2300	80	75	2
Lorry				
outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
2200	2200	80	75	2
2250	2250	80	75	2
2300	2300	80	75	2

(a) VSS - Two Lane Motorway

Car				
outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
2100	2100	120	80	3
2150	2150	100	80	3
2200	2200	80	75	3
Lorry				
outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
2100	2100	80	75	3
2150	2150	80	75	3
2200	2200	80	75	3

(b) VSS - Three Lane Motorway

outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
500	500	50	40	1
600	600	50	40	1
700	700	50	40	1
800	800	50	40	1
900	900	50	40	1
1000	1000	50	40	1
1100	1100	50	40	1
1200	1200	50	40	1
1300	1300	50	40	1
1400	1400	50	40	1
1500	1500	50	40	1

(c) ATS - City Centre

Table A.1: DTM control settings for Test Network I (cont.)

Appendix **B**

DTM Control Settings Test Network II

This appendix contains a complete overview of Dynamic Traffic Management control settings for Test Network II.

outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
200	200	50	40	1
300	300	50	40	1
400	400	50	40	1
500	500	50	40	1
600	600	50	40	1
700	700	50	40	1
800	800	50	40	1
900	900	50	40	1
1000	1000	50	40	1
1100	1100	50	40	1
1200	1200	50	40	1

(a) ATS 1 & 2 - Traffic Flow South - North

outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
1200	1200	50	40	1
1100	1100	50	40	1
1000	1000	50	40	1
900	900	50	40	1
800	800	50	40	1
700	700	50	40	1
600	600	50	40	1
500	500	50	40	1
400	400	50	40	1
300	300	50	40	1
200	200	50	40	1

(b) ATS 1 & 2 - Traffic Flow North - South

Table B.1: DTM control settings for Test Network II

Car				
outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
2200	2200	120	80	2
2250	2250	100	80	2
2300	2300	80	75	2
Lorry				
outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
2200	2200	80	75	2
2250	2250	80	75	2
2300	2300	80	75	2

(a) VSS - Motorway

outflow capacity	saturation flow	free-flow speed	speed at capacity	number of lanes
2200	2200	120	80	2
2200	2200	120	80	3

(b) VLS - Motorway

Table B.1: DTM control settings for Test Network II (cont.)

Appendix C

Probability of Improvement

In this appendix we explain how the Probability of Improvement can be calculated using the HyperVolume approach. In chapter 4 and 5 we referred multiple times to the work by Emmerich et al. (2006) as one of the first to use PoI. They define the PoI as follows:

$$PoI = \int_{\mathbf{y} \in \mathcal{H}_f} PDF_{\mathbf{x}}(\mathbf{y})\, \mathrm{d}y \tag{C.1}$$

Here $PDF_{\mathbf{x}}$ is the probability density function for input $\mathbf{x}$ and $\mathcal{H}_f$ denotes the non-dominated hypervolume for $\mathbf{x_{min}}$. Which according to them is:

$$\mathcal{H}_f = [-\infty, f_{min}] \tag{C.2}$$

This approach however does assume a single objective problem, and Emmerich et al. (2006) focus in the remainder of their paper mainly on constrained problems.

We therefore use the work of Keane (2006) who gives a clear explanation on how the PoI can be calculated in a biobjective problem. Now let f_o^i denote the objective value of point i for objective function o, where point $i \in \mathcal{P}$ and $\mathcal{P}$ contains all solutions of the Pareto front. Note that the value of f_o^i is exactly known, since it has been evaluated by the expensive model. Furthermore we have a single point x of which we want to determine the probability of improvement. For this point x we have determined the estimated objective value $\widehat{y_o}$ for objective function o and the accompanying estimated error s_o.

Figure C.1 gives an overview of the situation if only a single solution is known, i.e. $\mathcal{P}$ contains only a single solution. In this case we have three possible ways of improving the solution (we assume a minimisation problem): a) we only improve in objective 1; b) we only improve in objective 2; or c) we improve in both objectives simultaneously. If we would now assume a Normal distribution, we can calculate the PoI by adding the probability that we improve the situation in objective one and the probability that we improve

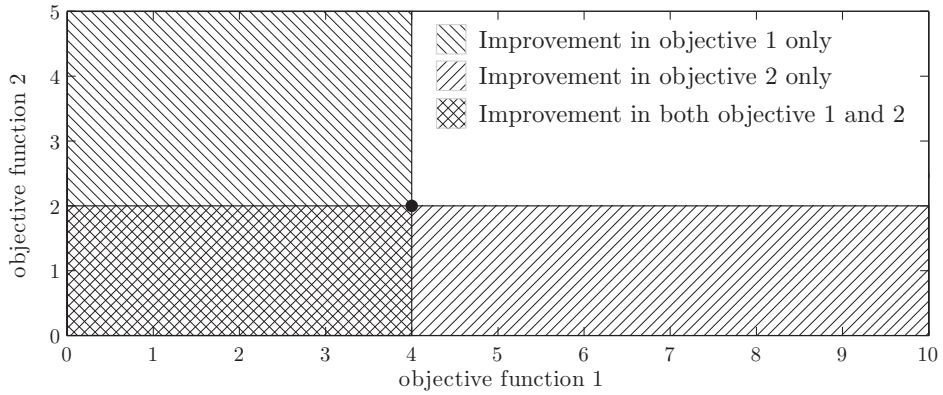


Figure C.1: Pareto front for a biobjective problem with one known solution

the situation in objective 2 and subtracting the probability that we improve both (because we have now calculated this probability twice). Thus in this case we can easily determine the probability of improvement by:

$$PoI = \Phi\left(\frac{f_1^1 - \hat{y}_1}{s_1}\right) + \Phi\left(\frac{f_2^1 - \hat{y}_2}{s_2}\right) - \Phi\left(\frac{f_1^1 - \hat{y}_1}{s_1}\right)\Phi\left(\frac{f_2^1 - \hat{y}_2}{s_2}\right) \tag{C.3}$$

Keane (2006) now expands this approach to a more generic approach for problems where more points are known. Furthermore he recognises that there are two ways to define an improvement in Pareto front. Figure C.2 shows that there are solutions that strictly dominate the existing set of solutions (grey), but also shows areas in which the so-called augmented solutions are located (hatched). Note that these areas are (for quite a large part) overlapping.

Now let M denote the total number of solutions in $\mathcal{P}$, i.e. in the Pareto front, and for notation convenience we use:

$$\Phi_o^i = \Phi\left(\frac{f_o^i - \hat{y}_o}{s_o}\right) \tag{C.4}$$

Then Keane (2006) provides two functions to determine the PoI, equation C.5 gives the PoI if augmented solutions are accepted, equation C.6 gives the PoI if only strictly dominating solutions are accepted. In order for this approach to work, it is necesary that the points in the Pareto front are sorted based on their objective value. Note that there is only a minor difference between the two equations.

$$PoI_{aug} = \Phi_1^1 + \sum_{i=1}^{M-1} \left\{\left(\Phi_1^{i+1} - \Phi_1^i\right) \cdot \Phi_2^i\right\} + \left(1 - \Phi_1^M\right) \cdot \Phi_2^M \tag{C.5}$$

$$PoI_{dom} = \Phi_1^1 + \sum_{i=1}^{M-1} \left\{\left(\Phi_1^{i+1} - \Phi_1^i\right) \cdot \Phi_2^{i+1}\right\} + \left(1 - \Phi_1^M\right) \cdot \Phi_2^M \tag{C.6}$$

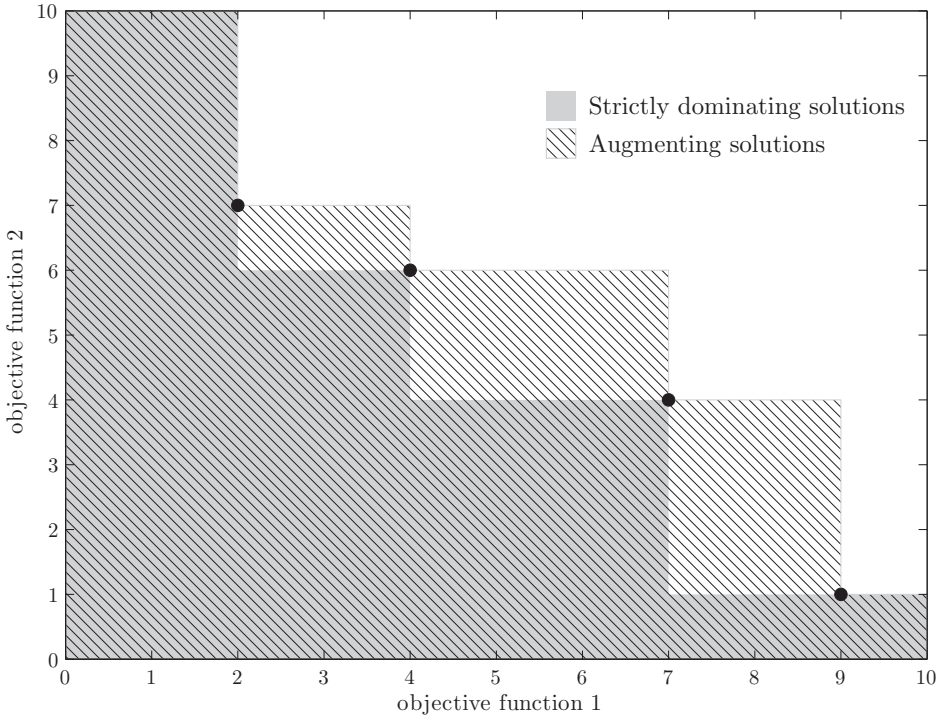


Figure C.2: Strictly dominating and Augmenting solutions

We will explain the approach by Keane (2006) using an example in which we want to determine the augmented PoI. For a new point (which position is irrelevant) we determined for four known solutions (thus $M = 4$) the probability that the objective value of this new point is lower. Figure C.3 shows the resulting values of ϕ_o^i for both objective functions ($o \in (1, 2)$) and all four known points ($i \in (1, \ldots, M)$). Note that Figure C.3 is not entirely similar to Figure C.1 and C.2. Instead of plotting the objective values on the axis, in this case for each solution in $\mathcal{P}$ the probability that the selected solution x is better is plotted, thereby creating a kind of unity square (all values are in the range [0,1]).

Clearly the gray area in Figure C.3 indicates the area where solutions are present that improve (augmenting) the current Pareto front. We can now use equation C.5 to determine the size of this area. For the first term of the equation ϕ_1^1 we find that this is equal to area I, since the range for both objective functions is $[0, 1]$. Now we let i run over $1, \ldots, M-1$ and determine the size of the second term in equation C.5 and we find for $i = 1, 2, 3$ the following solutions: $\left(\phi_1^2 - \phi_1^1\right) \cdot \phi_2^1$ which is equal to area II; $\left(\phi_1^3 - \phi_1^2\right) \cdot \phi_2^2$ which is equal to area III; and $\left(\phi_1^4 - \phi_1^3\right) \cdot \phi_2^3$ which is equal to area IV. Now we can determine the last term, which is $\left(1 - \phi_1^4\right) \cdot \phi_2^4$ which is equal to area V. This shows that equation C.5 does indeed provide us with the size of the

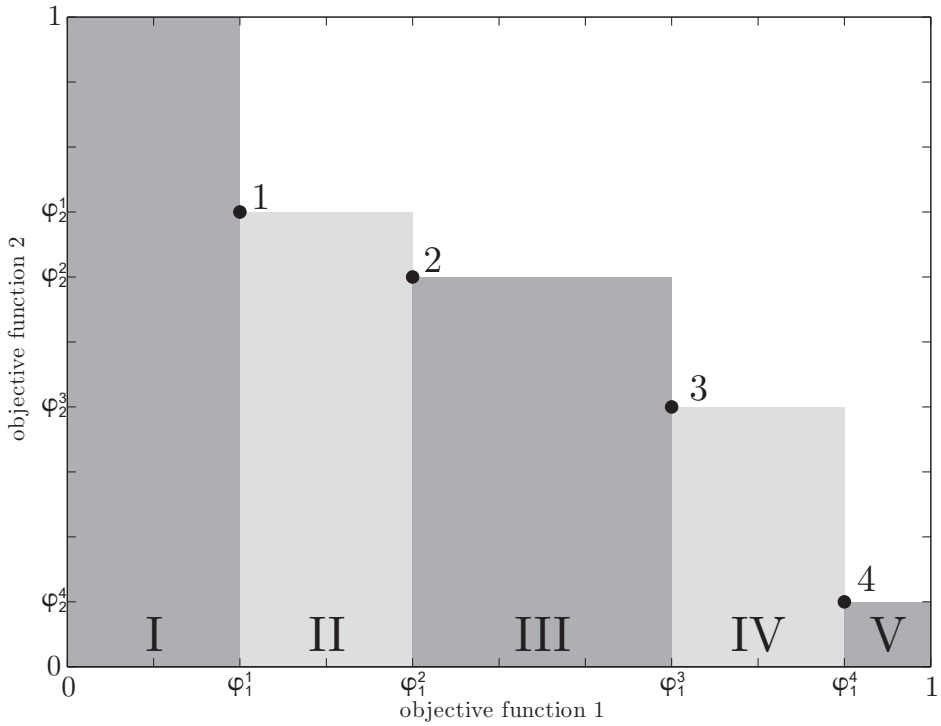


Figure C.3: Example for determining PoI_{aug} using the approach by Keane (2006)

area where improvements may be found. Because for each objective o it holds that $f_o \in [0, 1]$ we find that the size of the area is equal to the probability of improvement.

The question is now how we can change this approach to the case of multiobjective problems (i.e. more than two objective functions). First of all we select the augmented PoI, because this allows us to 'refine' the Pareto front instead of just finding solutions that are strictly dominating. Another benefit of selecting the augmented PoI is that if we look at a problem with three objective functions in a slightly different way (see Figure C.4) we notice that determining the PoI has a strong resemblance with determining the hypervolume of a 3D model. Area I consists of the PoI that we need to determine, but area II (displayed in grey) is a volume which can easily be determined using for instance the algorithm by While et al. (2006).

By plotting the probabilities instead of the objective values (as done in the example of Figure C.3) we create a unity cube (all values are in the range [0,1]). If we would now inverse this cube, i.e. instead of plotting Φ_o^i we plot $1 - \Phi_o^i$, we can easily use the Hypervolume by Slicing Objectives (HSO) algorithm of While et al. (2006) for determining the probability that the solution x will *not*

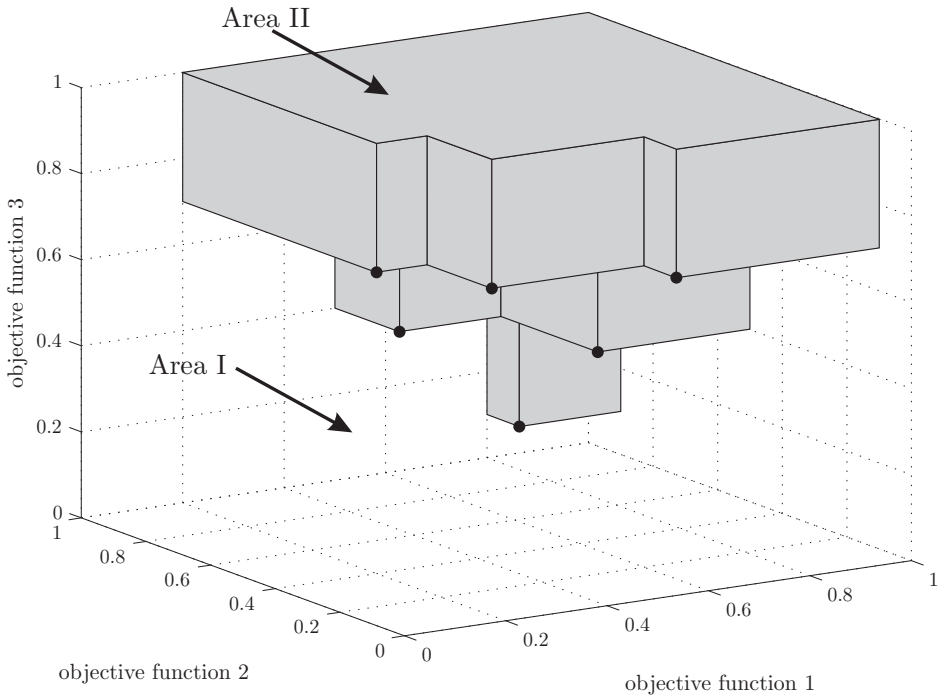


Figure C.4: Probability cube for a problem with three objectives

improve the current Pareto front. Of course now it is easy to determine the Probability of Improvement, which is 1 minus the volume of area II.

Appendix D

DTM Control Settings Testcase Almelo

This appendix contains a complete overview of Dynamic Traffic Management control settings for the testcase Almelo, as shown in Figure 7.3 of which a copy is shown below in Figure D.1.

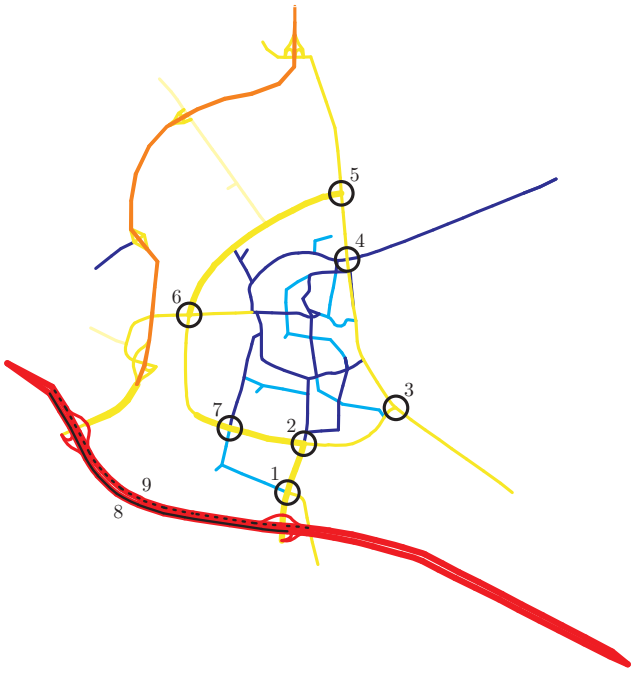


Figure D.1: DTM measures on the Almelo network

For each of the ATS systems we have indicated the directions of flow using North (N), East (E), South (S) and West (W), where North is assumed to be the top of the figure.

setting	N	E	S	W
a	2508	112	2508	201
b	2368	168	2368	302
c	2229	224	2229	402
d	2090	280	2090	503
e	1950	335	1950	604
f	1811	391	1811	704
g	1672	447	1672	805
h	1533	503	1533	906
i	1393	559	1393	1006

(a) Outflow capacities for ATS 1

setting	N	E	S	W
a	3622	279	2368	263
b	3421	418	2237	395
c	3220	557	2105	526
d	3019	697	1974	658
e	2817	836	1842	789
f	2616	975	1711	921
g	2415	1115	1579	1053
h	2214	1254	1447	1184
i	2012	1393	1316	1316

(b) Outflow capacities for ATS 2

setting	N	E	S
a	201	1811	1811
b	302	1711	1711
c	402	1610	1610
d	503	1509	1509
e	604	1409	1409
f	704	1309	1309
g	805	1207	1207
h	906	1107	1107
i	1006	1006	1006

(c) Outflow capacities for ATS 3

setting	N	E	S	W
a	1316	1006	1316	1006
b	1184	1107	1184	1107
c	1053	1207	1053	1207
d	921	1308	921	1308
e	789	1409	789	1409
f	658	1509	658	1509
g	526	1610	526	1610
h	395	1711	395	1711
i	263	1811	263	1811

(d) Outflow capacities for ATS 4

setting	N	S	W
a	604	1409	604
b	704	1308	704
c	805	1207	805
d	906	1107	906
e	1006	1006	1006
f	1107	906	1107
g	1207	805	1207
h	1308	704	1308
i	1409	604	1409

(e) Outflow capacities for ATS 5

setting	N	E	S	W
a	1316	1006	1393	1006
b	1184	1107	1254	1107
c	1053	1207	1115	1207
d	921	1308	975	1308
e	789	1409	836	1409
f	658	1509	697	1509
g	526	1610	557	1610
h	395	1711	418	1711
i	263	1811	279	1811

(f) Outflow capacities for ATS 6

Table D.1: DTM control settings for Almelo

setting	N	E	S	W
a	1409	836	1174	836
b	1308	975	1090	975
c	1207	1115	1006	1115
d	1107	1254	922	1254
e	1006	1393	839	1393
f	906	1533	755	1533
g	805	1672	671	1672
h	704	1811	581	1811
i	604	1950	503	1950

(a) Outflow capacities for ATS 7

setting	$v^{max;\text{car}}$
1	120
2	100
3	80

(b) Car speed limits for VSS 8

setting	$v^{max;\text{car}}$
1	120
2	100
3	80

(c) Car speed limits for VSS 8

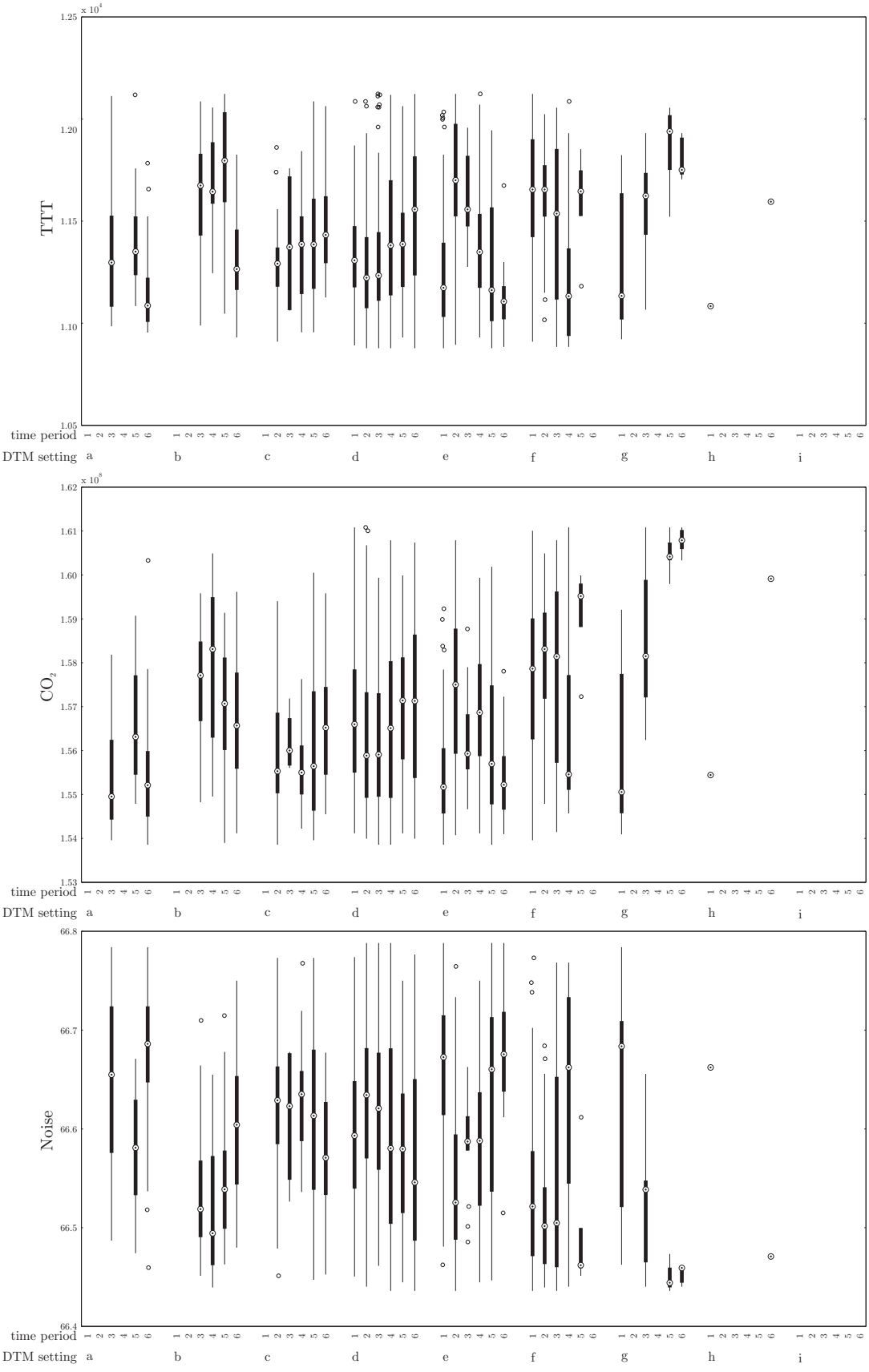
Table D.1: DTM control settings for Almelo (cont.)

Appendix E

Extensive Results for Testcase Almelo

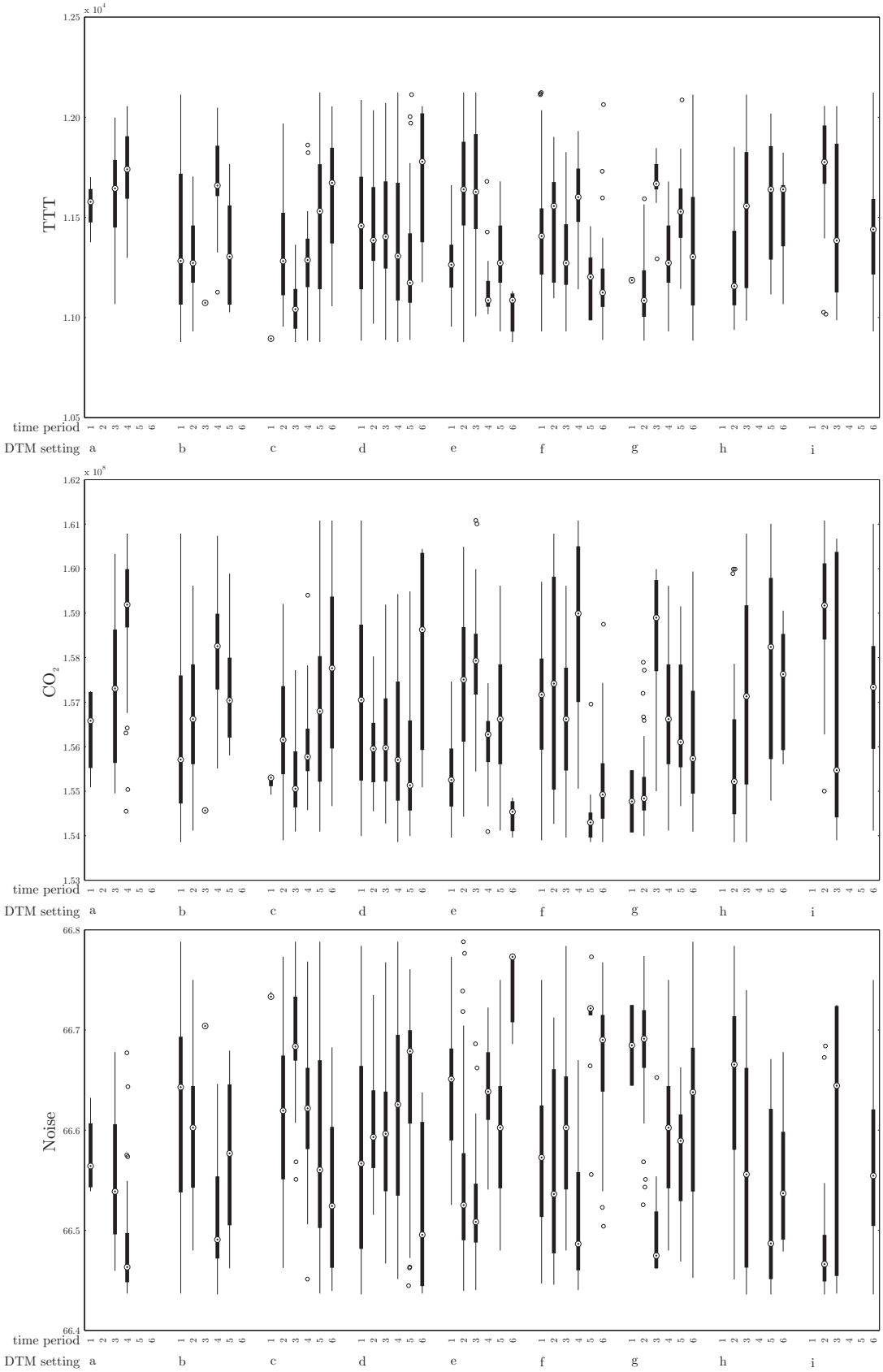
This appendix contains the extensive results for the traffic & transport analysis done on the Almelo network.

Figure E.1 on pages 166–174 contains the boxplots for the different DTM measures for all the objectives over all the settings and time periods. We grouped the results per setting, which makes it easier to analyse the effects of a specific setting.



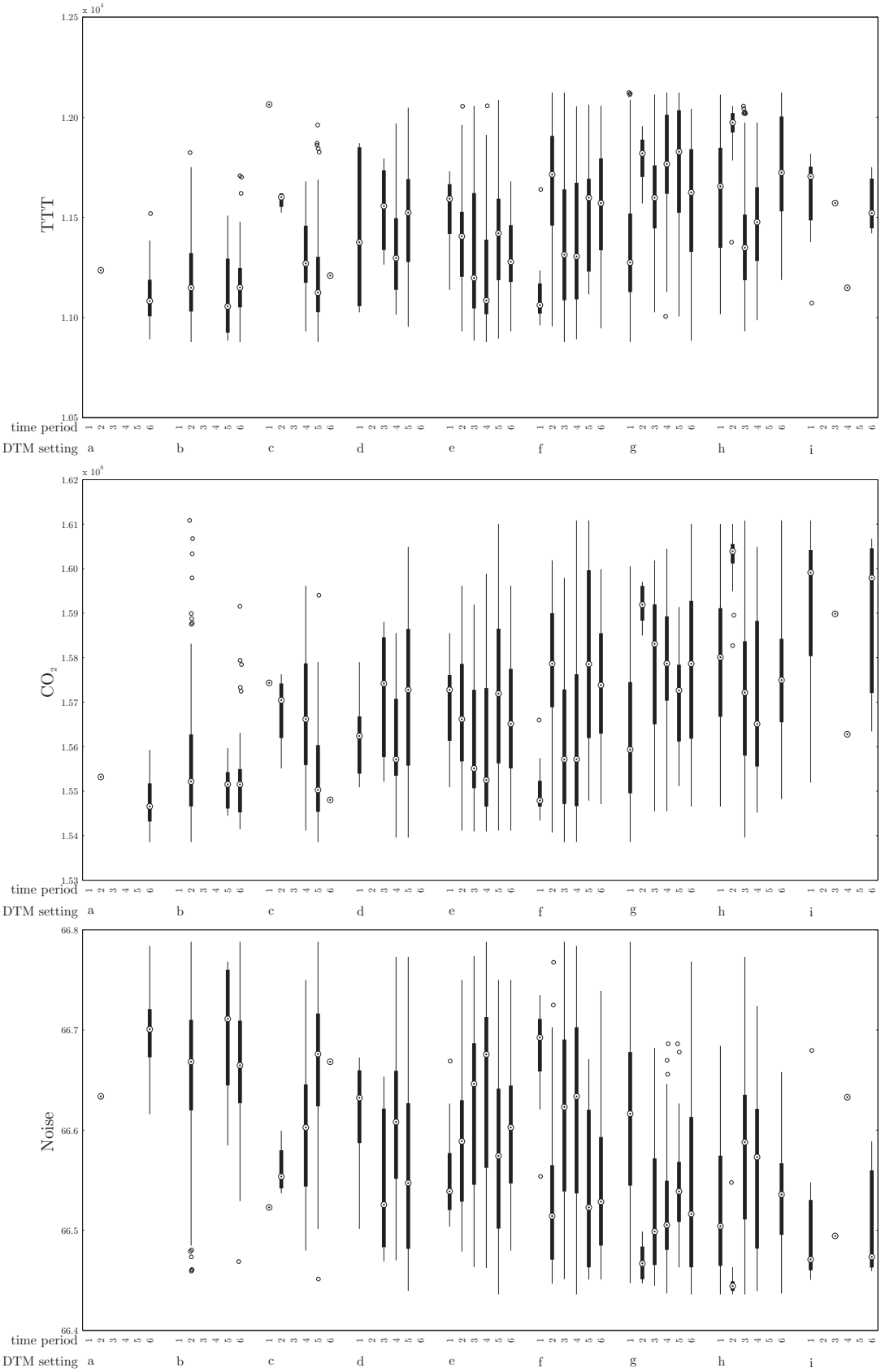
(a) Effects of ATS 1

Figure E.1: Boxplots of the effects for all DTM measures



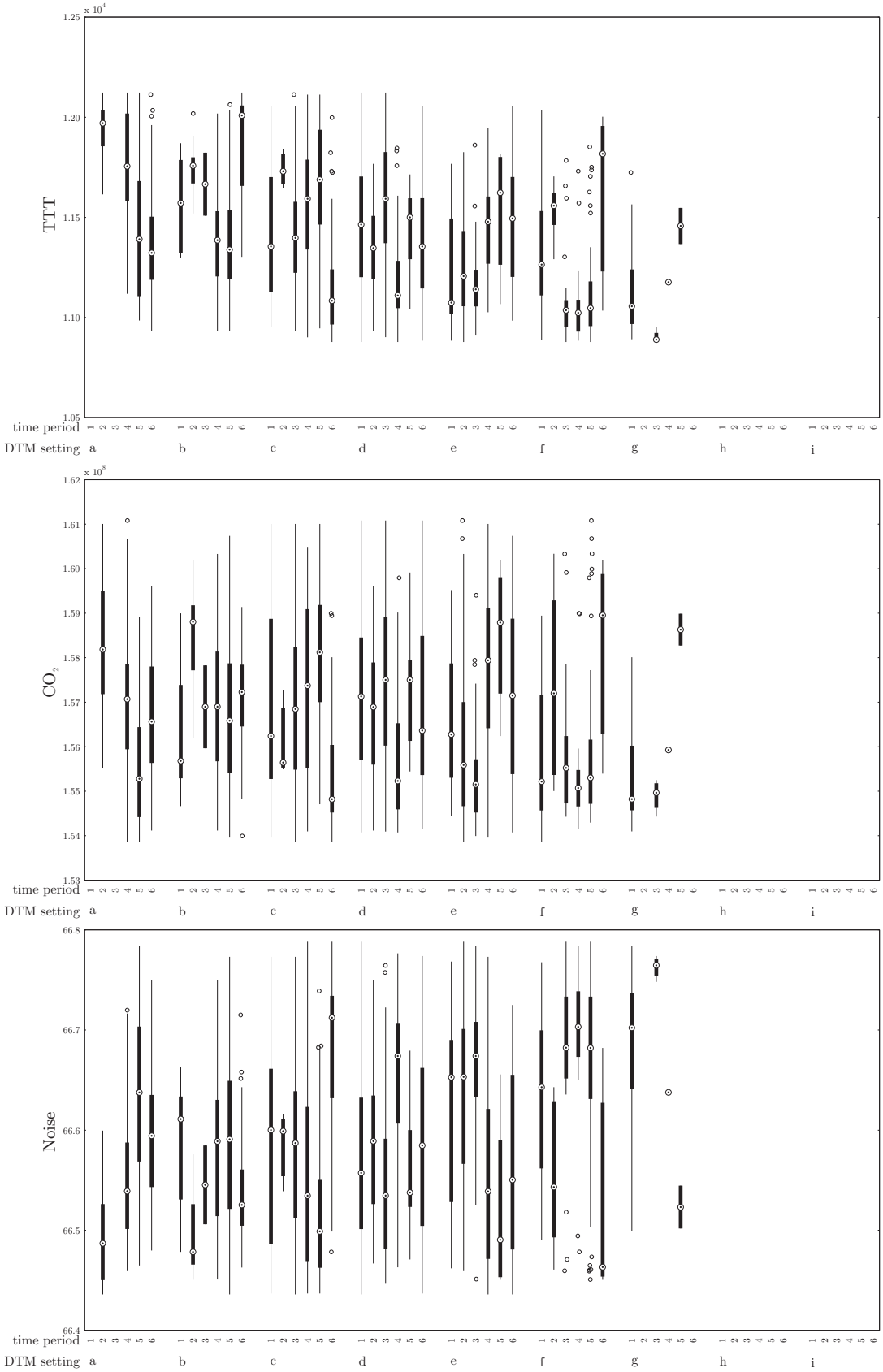
(a) Effects of ATS 2

Figure E.1: Boxplots of the effects for all DTM measures (cont.)



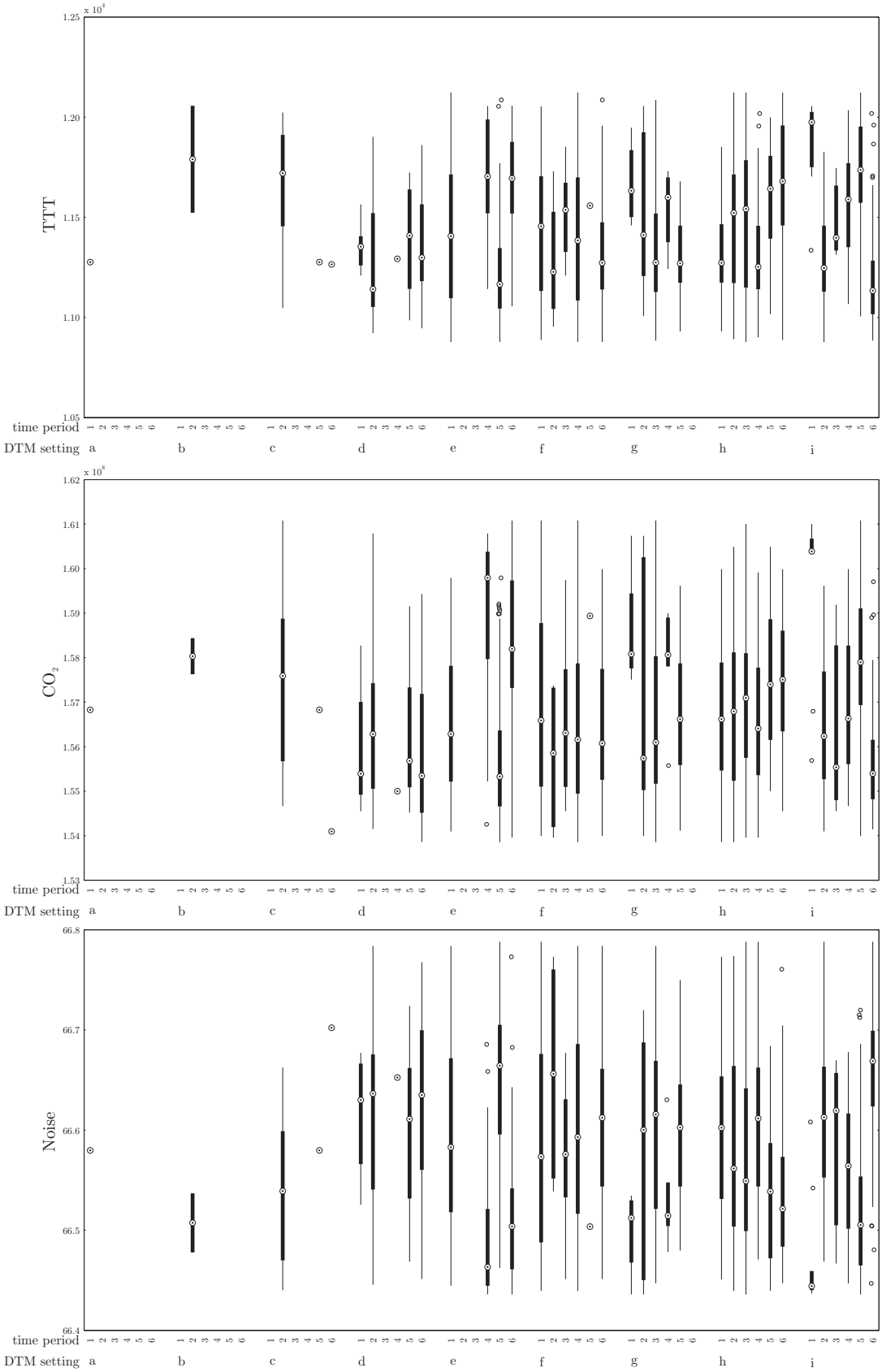
(a) Effects of ATS 3

Figure E.1: Boxplots of the effects for all DTM measures (cont.)



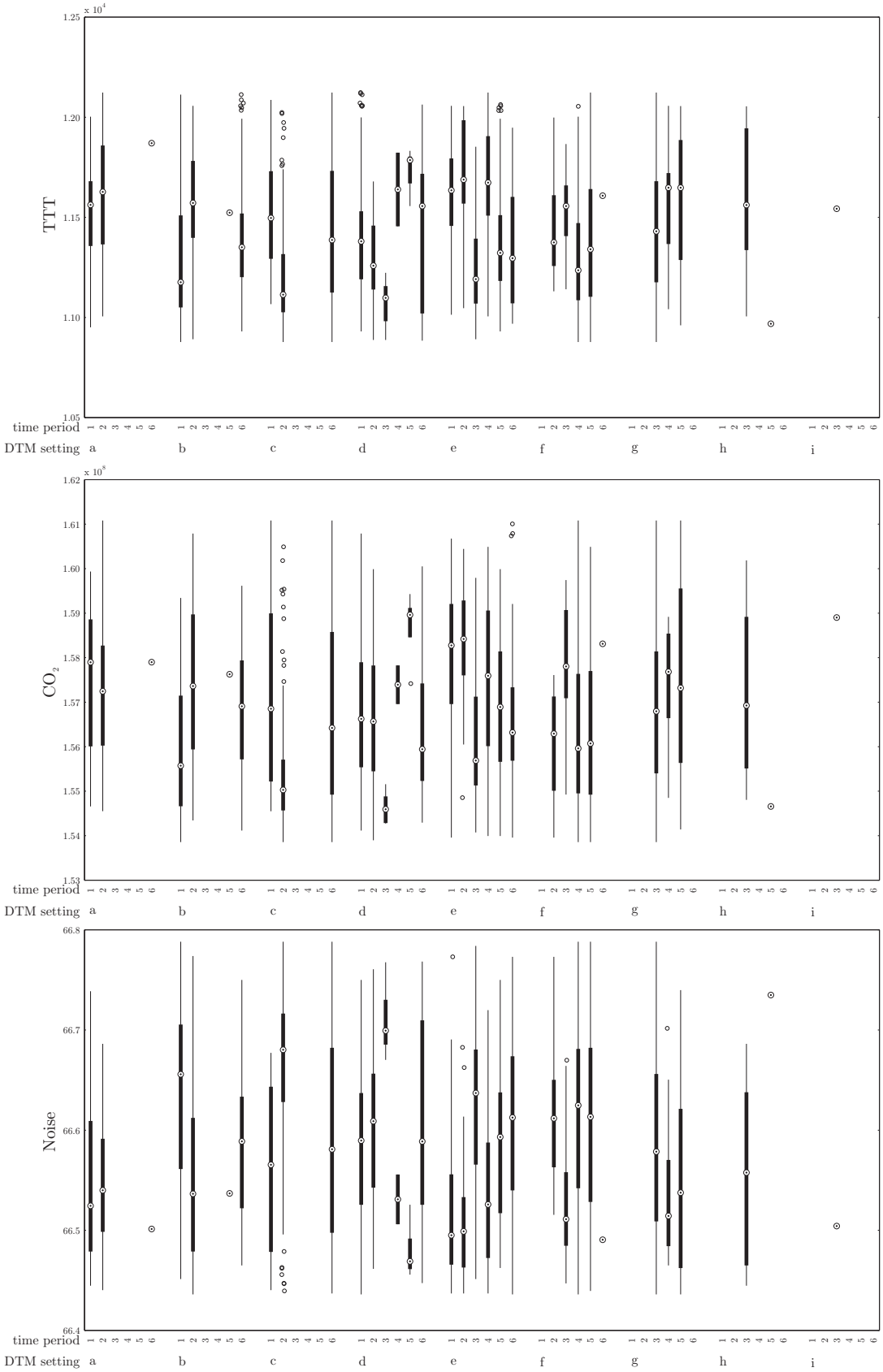
(a) Effects of ATS 4

Figure E.1: Boxplots of the effects for all DTM measures (cont.)



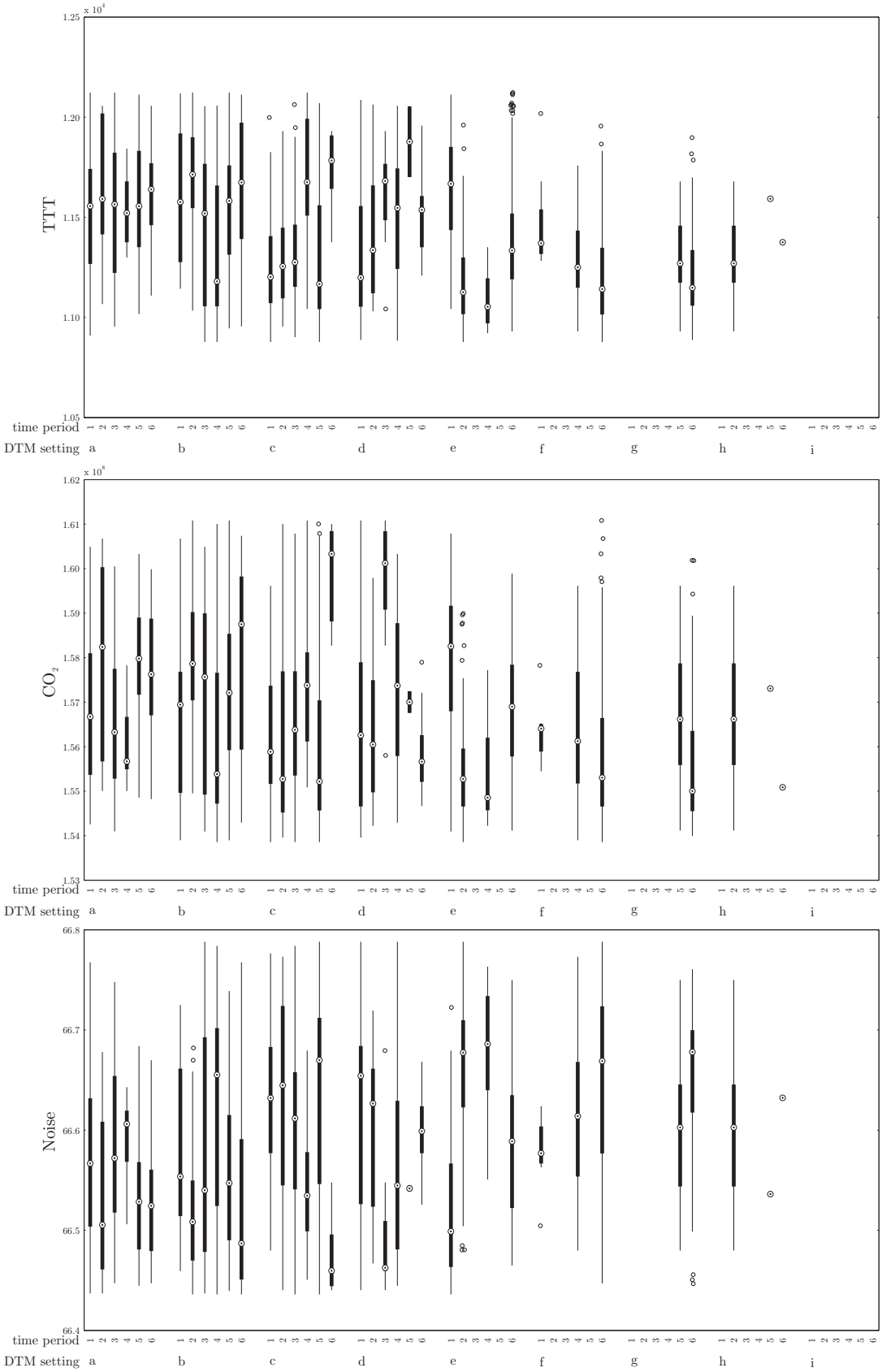
(a) Effects of ATS 5

Figure E.1: Boxplots of the effects for all DTM measures (cont.)



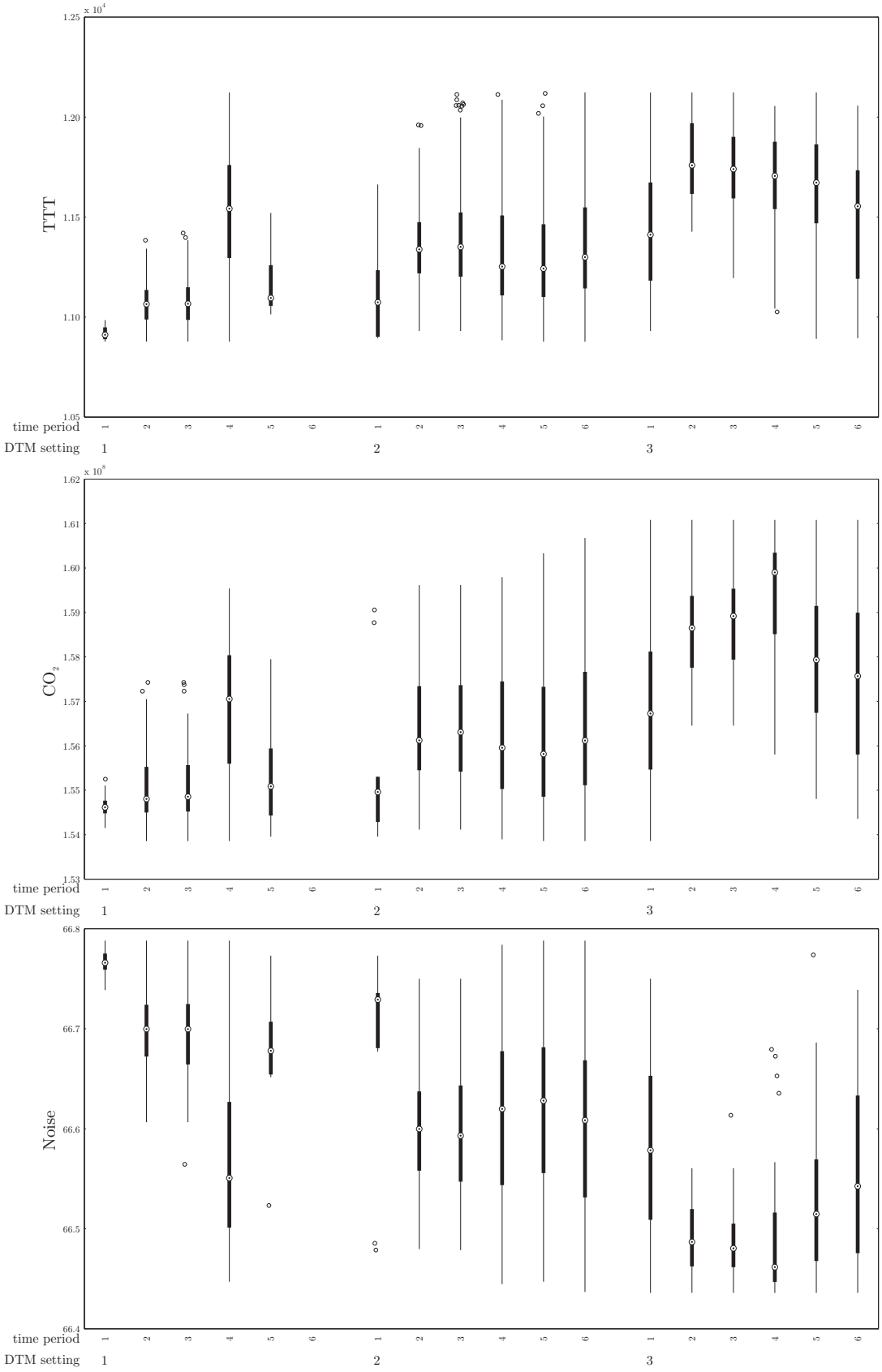
(a) Effects of ATS 6

Figure E.1: Boxplots of the effects for all DTM measures (cont.)



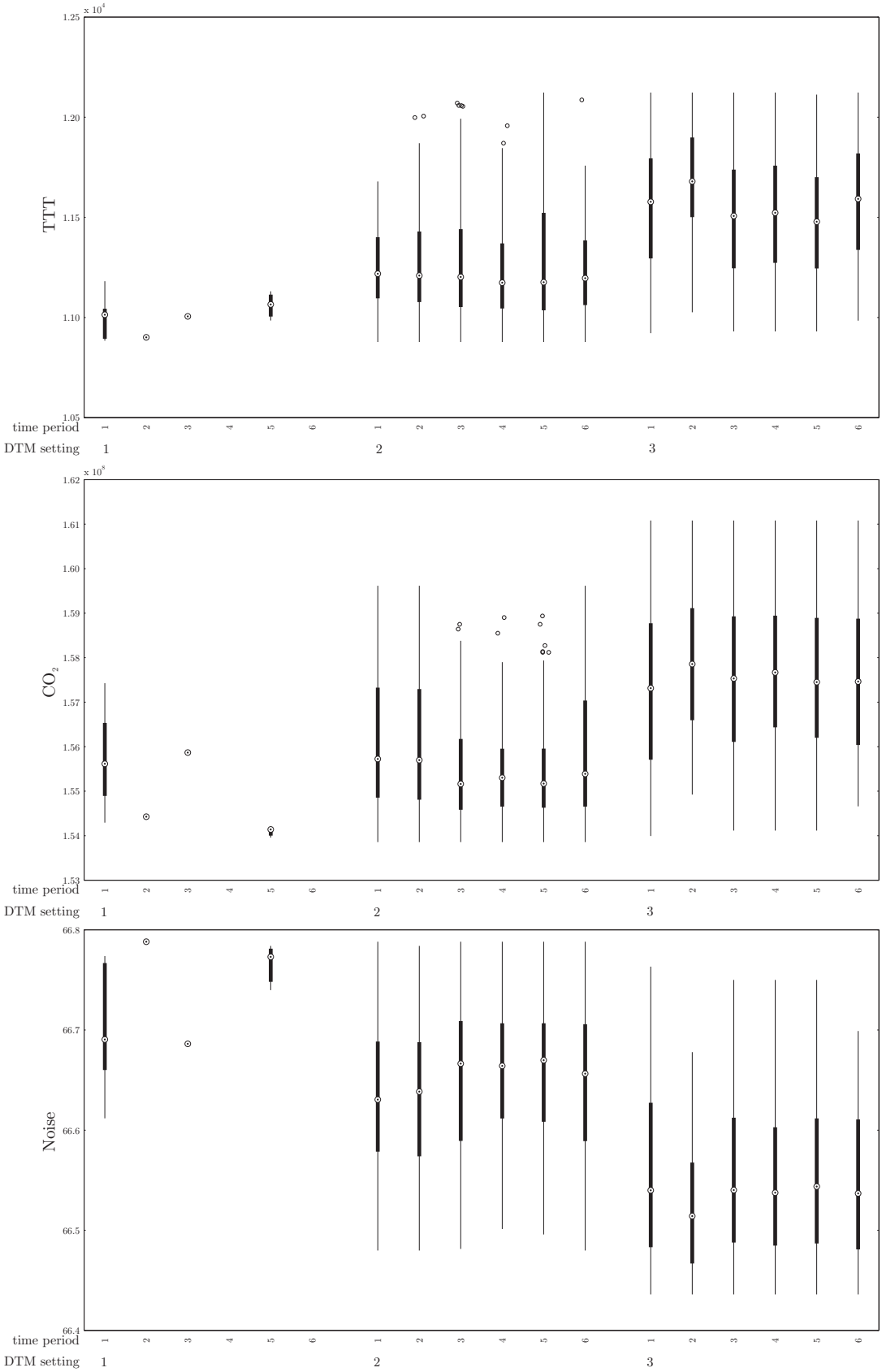
(a) Effects of ATS 7

Figure E.1: Boxplots of the effects for all DTM measures (cont.)



(a) Effects of VSS 8

Figure E.1: Boxplots of the effects for all DTM measures (cont.)



(a) Effects of VSS 9

Figure E.1: Boxplots of the effects for all DTM measures (cont.)