

Proceedings of the 106th European Study Group
Mathematics with Industry

SWI 2015

Utrecht, January 26–30, 2015

Editors:

Rob Bisseling

Martin Bootsma

Jason Frank

Ross Kang

Tobias Müller

Alessandro Sbrizzi

Cristian Spitoni

Paul Zegeling

Cover design by Faidra Barba
ISBN: 978-94-028-0161-3

Contents

Contents	3
Introduction	5
Always nice weather in Europe	7
Xiulei Cao, Alje van Dam, Bart de Leeuw, Carina Geldhauser, Johan Grasman, Ivan Kryven, Domenico Lahaye, Leonardo Morelli, Vivi Rottschäfer, Han Zhou	
Mobidot: inferring transportation modes from smartphone sensors	21
Rob Bisseling, Jason Frank, Haris Gavranovic, Jasper van Heugten, Anna Kruseman, Daphne van Leeuwen, Christian Reinhardt	
Adapting radiotherapy treatment to deformations in the patient	35
Claudiu Antonovici, Tugce Akkaya, Arjen Baarsma, KaYin Leung, Corine Meerman, Lisanne Rens	
Power line route optimisation in a finite spatial grid	55
Nieke Aerts, Emile Broeders, Erik Bruin, Ross J. Kang, Pedro Munari	
Patient-adaptive compressed sensing for MRI	85
Lee Gonzales, Koos Huijssen, Rakesh Jha, Tristan van Leeuwen, Alessandro Sbrizzi, Willem van Valenberg, Ian Zwaan, Johan van den Brink, Elwin de Weerd	
Statistical modeling of mechanical bearing life testing	103
Sébastien Blachère, Martin Bootsma, Alessandro Di Bucchianico, Mike Keane, Xinru Li, Andrea Roccaverde, Cristian Spitoni, Dong Yan	

Introduction

The *Study Group Mathematics with Industry* is an occasion for companies, non-profit organisations and government institutes to pose problems to University Mathematicians, who then try their best to solve these questions in teams during an intensive study week. These are the scientific proceedings of the 106th European Study Group Mathematics with Industry that took place Monday 26 January–Friday 30 January 2015 in Utrecht, The Netherlands. A total of 61 Mathematicians made a big effort to crack six diverse problems from industry during this week. On Monday, the companies presented their questions and the participants divided themselves into six groups, each devoted to one of the questions. On Friday each group presented the solution they had come up with during the week. Most participants were Mathematicians working for one of the Dutch Universities, but some participants also came from (much) further. Mathematicians from Belgium, Bosnia-Herzegovina, Brazil, Bulgaria, Georgia, Germany, Macedonia and the United Kingdom travelled to the Netherlands specifically for the purpose of taking part in the study group.

In the present volume, you will find the scientific reports of each of the six teams that give the details of the solutions that the teams came up with during the week. These reports were written by the participants themselves after the week had finished. There is also an accompanying volume with *popular proceedings*. These were written by science journalist Anouck Vrouwe, based on the presentations, scientific reports and interviews with the participants and company representatives. The popular proceedings were written in Dutch and are aimed at a general audience, while the current volume is aimed at experts and expose the full details of the approaches chosen by the study group participants.

The six questions came from a diverse range of companies and needed very different mathematical approaches. The Royal Netherlands Meteorological Institute (KNMI) asked what is the best way to combine a number of weather predictions into one “super model” (SUMO). The company Mobidot asked for the help of the Mathematicians in better analyzing the travel behaviour of people using an app that runs on their mobile phones. Philips Healthcare wanted to lower the costs of MRI scans by constructing a good image using fewer measurements than are presently used. The Netherlands Cancer Institute (NKI) asked the participants for radiation treatment programs that

run over multiple weeks and that adapt to changes in the patient, such as the changing position of the tumor due to weight loss or gain of the patient. The British NM group wanted methods to place power poles and power lines in such a way that all demands (such as minimal cost and minimal inconvenience for the people that live in the area) are met as much as possible. The company SKF asked for the most efficient way to test the reliability of the ball bearings they produce. In each of the six problems from industry, the participants managed to arrive at least at a partial solution, and in some cases to a complete solution that can immediately be used in practice.

This year, the study group could count on considerable attention from the media. Amongst others, the national radio show *EenVandaag* devoted items to the study group on three different days with short interviews with participants and company representatives. The local TV station *RTV Utrecht* also covered the study group, and the magazine *Technisch Weekblad* devoted an article to the study group.

To conclude, we would like to thank everyone that helped make the study group into the big success that it was. In the first place of course we thank our main sponsors the Netherlands Organisation for Scientific Research (NWO) and the Technology Foundation (STW) for their generous contribution. We thank the Royal Dutch Mathematical Society (KWG) for their financial contribution. We also thank the participating companies for taking part in the week and for offering challenging Mathematical problems from the cutting-edge of their operations. We thank Faidra Barba for graphic design. We thank Jean Arthur, Ria Bekkering and Cécile Lemette for secretarial support and Monica van de Garde for bringing our event to the attention of the press. Finally, we of course also wish to thank the participants for their effort and enthusiasm in finding solutions to the problems proposed by the companies.

Utrecht, November 2015

Rob Bisseling
Martin Bootsma
Jason Frank
Ross Kang
Tobias Müller
Alessandro Sbrizzi
Cristian Spitoni
Paul Zegeling (the organizers)

Always Nice Weather in Europe

Xiulei Cao (TU Eindhoven), Alje van Dam (Utrecht University), Bart de Leeuw (Utrecht University), Carina Geldhauser (University of Bonn), Johan Grasman (Universiteit Wageningen), Ivan Kryven (University of Amsterdam), Domenico Lahaye (Delft University of Technology), Leonardo Morelli (Leiden University), Vivi Rottschäfer (Leiden University), Han Zhou (Utrecht University)

November 25, 2015

Abstract

Weather forecasting relies on mathematical models that exhibit chaotic behavior. This renders the solution of these models very sensitive to errors in the model, to choices of the initial conditions and to truncation errors in the numerical solution procedure. Over the course of the past decade, various meteorological institutes in Europe have developed different atmosphere models. Each of these models has its strengths and weaknesses. The principle behind the so-called Super Modeling approach is to merge these existing models into a single larger model to combine common strengths while overcoming individual weaknesses. This approach was initially proposed and developed by the KNMI in the Netherlands to improve the reliability of its weather forecasts. The task formulated for this Study Group problem was to reevaluate the Super Modeling approach and to formulate recommendations for its future development.

1 Introduction

Meteorological institutes are continually seeking to improving their weather forecasts. Research is directed towards minimizing the discrepancy between mathematical models predicting atmospherical conditions and actual measurements. These mathematical models typically require the solution of systems of non-linear ordinary differential equations that allow for solutions with a chaotic behavior. Thus any solution procedure is prone to errors in the modeling, in the initial conditions and in the numerical time integration procedure. This

has lead to the development of different models by various European institutes with their own strengths and weaknesses. None of these models is currently accredited of giving the best overall simulation results.

To overcome the absence of a universally best weather simulation environment, the Dutch meteorological institute KNMI in collaboration with partner institutes pioneered an approach in which different models are combined into a larger model. This combination aims at exploiting the strength of different approaches while overcoming individual weaknesses. Weather models are combined by synchronizing their outputs, i.e., by penalizing the deviation of a single model from a common prediction of a time evolution. The guiding principle is that the common realization agrees better with reality than each of the individual ones. The underlying idea of *synchronization* [1] is known to play an important role in e.g. social sciences (e.g. in the common start and ending of applause for a performance) and in biology (e.g. in the migration of flocks). The KNMI and partners coined their approach the *Super Modeling* (SUMO) approach [2].

Given its recent development, the SUMO approach generates a large number of interesting research questions. Examples include:

- weather forecasting models describe the time evolution of an n -dimensional state vector. It is not a-priori clear how many and which components of this state vector should be coupled in order to obtain synchronization. Neither is clear whether this set should change in time. A mechanism to enforce synchronization using a small (and possibly time-dependent) set of components would be beneficial to have;
- in their studies, the KNMI and partners enforced synchronization by constraining the norm of the difference between state vectors corresponding to the individual models. This mechanism is referred to as *linear nudging*. It is not a-priori clear whether this mechanism is optimal and what alternative synchronization mechanism should be considered;
- it not immediately clear what value the coupling coefficient in a linear nudging technique should have and what strategies could be developed to obtain these values by matching with previously recorded data as in see [3];
- it is intuitively clear that a very weak coupling of M individual models results in a super model exhibiting the M individual dynamic behaviors. A very strong coupling on the other hand results in a super model with

a single dynamic that is somehow the average of the individual dynamics leading to incorrect results. The idea is that intermediate coupling strengths are most appropriate. However, KNMI and partners have observed that for certain choices of intermediate coupling strength a system with new dynamics arise. So-called *ghost-attractors* arise with predicted outcome: *always nice weather in Europe*. The question was to find an explanation for this.

The Study Group was asked to consider the above issues. The Lorenz-63 model will be used as an illustrative example and guide in the numerical studies.

This report is structured as follows. This Introduction will be followed by three main sections. In Section 2 we consider a technique based on coupling restricted to unstable directions in the tangent space. In Section 3 we consider the coupling of three copies of the Lorenz model. In Section 4 we study the appearance of ghost attractors. In Section 5 we study alternative coupling approaches. Conclusions finally are drawn in Section 6.

2 Dynamical Properties of Imperfect Models and the Supermodel

The true state of a physical system is assumed to be given by a set of observations $\{t_i, P(t_i)\}$, where $P(t)$ denotes the state vector of the truth system at time t . Available are a set of imperfect models for which the values of parameters can be obtained by fitting the model to the observations. The study focuses on systems with chaotic dynamics, which are described as follows by the system of differential equations

$$\frac{dx}{dt} = f(x) \quad (1)$$

in a n -dimensional state space. Let $x(t) = p(t)$ be a chaotic solution, then other trajectories nearby this chaotic orbit are analyzed by making use of the tangent linear system. By substituting

$$x(t) = p(t) + v(t)$$

into (1) and preserving only the linear terms one obtains the tangent linear system

$$\frac{dv}{dt} = F[p(t)]v(t) \text{ with } F[p(t)] = \{\partial f_i(p(t))/\partial x_j\}_{n \times n} \quad (2)$$

in which $v(t)$ denotes the perturbation from $p(t)$. The matrix F has time-dependent eigenvalues. Their averages over a full orbit are called the Lyapunov exponents. For dissipative systems the sum of these exponents is negative. The orbit $p(t)$ is chaotic if at least one of the exponents is positive. It means that in certain parts of the state space at least one eigenvalue must be positive and that an orbit must pass such a region from time to time in order to have a chaotic orbit.

The Lorenz-63 model is given by

$$\frac{dx_1}{dt} = \sigma(x_2 - x_1), \quad (3)$$

$$\frac{dx_2}{dt} = x_1(\rho - x_3) - x_2, \quad (4)$$

$$\frac{dx_3}{dt} = x_1x_2 - \beta x_3. \quad (5)$$

From the Lorenz-63 system it is known that in the 3D state space near the origin such a region exists. There the velocity of the trajectories has a large component in the direction of the x_3 -axis, so a bundle of trajectories moves in a direction of x_3 and exhibits after passage of the origin a strong tendency to diverge. This means that, if one wants to perturb a chaotic orbit $p(t)$, one has to do that in a direction perpendicular to the x_3 -axis. In order to synchronize two imperfect (Lorenz-63) models, which both pass in a similar manner the region near the origin, one has to apply a coupling of the form

$$\frac{dx_1}{dt} = f_1(x) + c_1(y_1 - x_1) \quad \frac{dy_1}{dt} = g_1(y) + c_1(x_1 - y_1) \quad (6)$$

$$\frac{dx_2}{dt} = f_2(x) + c_2(y_2 - x_2) \quad \frac{dy_2}{dt} = g_2(y) + c_2(x_2 - y_2) \quad (7)$$

$$\frac{dx_3}{dt} = f_3(x) + c_3(y_3 - x_3) \quad \frac{dy_3}{dt} = g_3(y) + c_3(x_3 - y_3) \quad (8)$$

where $f(x)$ and $g(y)$ may differ in just the values of the model parameters. Thus, for the Lorenz-63 system any vector $c = (c_1, c_2, 0)$ may give rise to synchronisation for the two coupled imperfect models. Also the size of the coupling vector c may play a role. The best coupling vector is found by composing the *super model*

$$S(t; c) = [x(t; c) + y(t; c)]/2 \quad (9)$$

and find the best fit of this model to the available observations $\{t_i, P(t_i)\}$. For higher dimensional systems such computations may need too much time.

Then one may concentrate on the eigenvector that corresponds with the largest (positive) eigenvalue. Together with the eigenvector, that corresponds with the eigenvalue equal to zero, it spans the *most unstable* manifold of the chaotic orbit, see Figure 1. In the direction of the flow given by the vector dp/dt a perturbation $v(t)$ is neither damped nor does it tend to explode, so in that direction the system is neutrally stable and yields therefore an eigenvalue equal to zero. Thus, one may take the eigenvector that corresponds with the largest eigenvalue as the coupling vector c with the appropriate length or choose its projection in the space perpendicular to dp/dt .

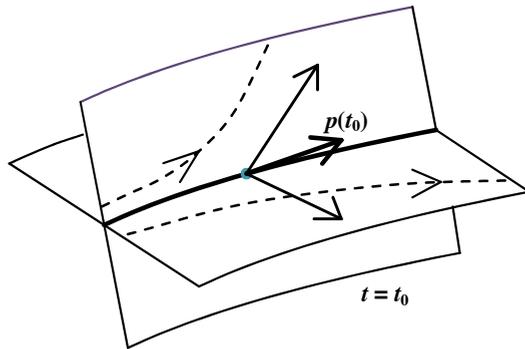


Figure 1: Dynamics of the trajectories near the chaotic orbit, being a stable strange attractor. There the 3D state space can be decomposed in two manifolds together with the vector $p(t)$. Near a point $p(t)$ on the attractor these manifolds are planes which are spanned by the eigenvector $p(t)$ and each of the two other eigenvectors. Depicted is the case that the two manifolds are unstable. For chaos it suffices that only one of the manifolds is unstable.

3 First numerical exploration of a coupled Lorenz-63 system

In this section, we study the following system of two identical Lorenz models with different initial data and linear coupling:

$$\begin{aligned}
\dot{x}_1 &= \sigma(y_1 - x_1) + C_1^x(x_2 - x_1) \\
\dot{y}_1 &= x_1(\rho - z_1) - y_1 \\
\dot{z}_1 &= x_1y_1 - \beta z_1 \\
\dot{x}_2 &= \sigma(y_2 - x_2) + C_2^x(x_1 - x_2) \\
\dot{y}_2 &= x_2(\rho - z_2) - y_2 \\
\dot{z}_2 &= x_2y_2 - \beta z_2
\end{aligned} \tag{10}$$

where we choose $\sigma = 10, \rho = 28$ and $\beta = \frac{8}{3}$. This is a typical choice of parameters that is used often for the Lorenz system; chaos is found and the famous butterfly attractor arises.

It has been shown recently, see [2], that limited information exchange between two identical Lorenz systems can lead to synchronization of the model states even when the systems are initialized from very different initial conditions and differ slightly in parameter values. The ability to synchronize with the truth measures the quality of the model.

Through trial and error, it has been found that often just a part of the state space vectors need to be exchanged between the models in order for the models to synchronize on a common solution.

A first question is whether the coupling should be symmetric in the phase space variables. For us *symmetric coupling* for a system means that the coupling constant for one variable in the first model must be the same as the coupling constant for the same variable in the second model. In terms of the system (10) it means that $C_1^x = C_2^x$. For three systems with different parameters it was demonstrated in [2] that couplings need not be symmetric.

Indeed, we could experimentally show that the solutions of two identical Lorenz models with different initial condition need to be coupled only in the x -variables, and the coupling does not have to be of the same strength in both equations. For the above coupled model (10), a total coupling strength of at least 9 was necessary, and synchronisation happened both when this coupling strength was imposed in only in one of the models, so either $C_1^x > 9$ and $C_2^x = 0$, or $C_1^x = 0$ and $C_2^x > 9$, or when the coupling strength was achieved cumulatively, i.e. with $C_1^x + C_2^x > 9$

An important question is: given two identical Lorenz Models with different initial conditions, which are the coupling strengths that lead to synchronization without essentially changing the dynamics?

Using a straight-forward implementation of two identical Lorenz models in Mathematica, we verified the following values: Synchronization happens for a cumulative coupling strength of above 9, with the critical values being $C_1^x = 8$ and $C_2^x = 1$, where a long time has to pass until the solutions synchronize: Only after $t = 60$ do we observe a sufficiently small error in the x -coordinates, which is then still several orders of magnitude bigger than in the symmetric case. For example, for $C_1^x = 10 = C_2^x$, the error falls below 10^{-6} for $t > 20$.

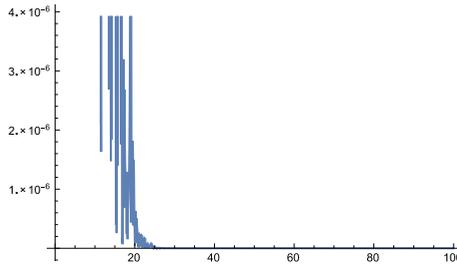


Figure 2: Error for $C_1^x = 10 = C_2^x$

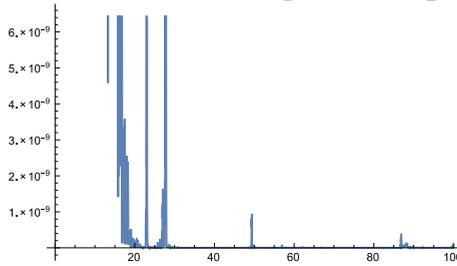


Figure 3: Error for $C_1^x = 20$ and $C_2^x = 10$

In terms of an upper bound on the coupling constants, we observed an unexpected robustness of the dynamics: Even for quite high values like $C_1^x = 20$ and $C_2^x = 10$, synchronization seems to happen for times $t > 30$, but sudden spikes in the error $|x_1(t) - x_2(t)|$ of order 10^{-6} keep appearing in irregular intervals.

These results are stable and not sensitive even to significant changes of the initial conditions (i.e. of values of ± 20 per phase space coordinate).

We observed experimentally that the achieved synchronization is lost even when the parameters of the Lorenz models are changed slightly. For example, choosing $\beta = 3$ instead of $\beta = \frac{8}{3}$ in one model destroys the effect completely.

4 Stability analysis of ghost attractors

4.1 Introduction

It has been observed [4] that coupling multiple models to form a super model may lead to the super model getting “stuck” in a certain part of phase space (such that it is always nice weather in Europe). In the context of a number of coupled Lorenz models, this can lead to all models being fixed at two points close to the unstable fixed points in the Lorenz butterfly [4]. In this case some models will be in a point on one wing, some others will lie on the other wing. The precise position of the points where they are fixed depends on the exact formulation of the super model.

We analyse this behaviour by considering a simpler model in which we couple identical models. We then analyse the fixed points and their stability. In case a fixed point is stable we can conclude that there is a set of initial conditions for which the supermodel converges to this fixed point solution.

4.2 The model and our simplifying ansatz

We consider $N + M$ coupled Lorenz63 models all with the same choice of the coefficients, with linear and uniform coupling and standard parameters, leading to the following equations (with $i = 1, 2, \dots, N + M$)

$$\begin{aligned}
 \dot{x}_i &= 10(y_i - x_i) + C \sum_{j=1}^{N+M} (x_j - x_i) \\
 \dot{y}_i &= x_i(28 - z_i) - y_i + C \sum_{j=1}^{N+M} (y_j - y_i) \\
 \dot{z}_i &= x_i y_i - \frac{8}{3} z_i + C \sum_{j=1}^{N+M} (z_j - z_i).
 \end{aligned} \tag{11}$$

Since we know from [4] that these oscillators may get stuck into a situation in which there are N oscillators in some fixed point and M other oscillators also together in one (possibly different) point, our approach now consists of finding all possible fixed points under the assumption that the oscillators are stuck in at most two points, for given C , N and M . If we label one fixed point

by $(x_*^{(1)}, y_*^{(1)}, z_*^{(1)})$ and assume that M oscillators are stuck in this point, and the other by $(x_*^{(2)}, y_*^{(2)}, z_*^{(2)})$ where N oscillators are stuck. Then the equations simplify to

$$\begin{aligned}\dot{x}_i &= 10(y_i - x_i) + CM(x_*^{(1)} - x_i) + CN(x_*^{(2)} - x_i) \\ \dot{y}_i &= x_i(28 - z_i) - y_i + CM(y_*^{(1)} - y_i) + CN(y_*^{(2)} - y_i) \\ \dot{z}_i &= x_i y_i - \frac{8}{3}z_i + CM(z_*^{(1)} - z_i) + CN(z_*^{(2)} - z_i).\end{aligned}\quad (12)$$

An important remark is that these equations are only consistent in case the oscillators lie actually exactly in these points. Hence, at these fixed points our system of equations reduces to M copies of

$$\begin{aligned}0 &= \dot{x}_*^{(1)} = 10(y_*^{(1)} - x_*^{(1)}) + CN(x_*^{(2)} - x_*^{(1)}) \\ 0 &= \dot{y}_*^{(1)} = x_*^{(1)}(28 - z_*^{(1)}) - y_*^{(1)} + CN(y_*^{(2)} - y_*^{(1)}) \\ 0 &= \dot{z}_*^{(1)} = x_*^{(1)} y_*^{(1)} - \frac{8}{3}z_*^{(1)} + CN(z_*^{(2)} - z_*^{(1)})\end{aligned}\quad (13)$$

and N copies of

$$\begin{aligned}0 &= \dot{x}_*^{(2)} = 10(y_*^{(2)} - x_*^{(2)}) + CM(x_*^{(1)} - x_*^{(2)}) \\ 0 &= \dot{y}_*^{(2)} = x_*^{(2)}(28 - z_*^{(2)}) - y_*^{(2)} + CM(y_*^{(1)} - y_*^{(2)}) \\ 0 &= \dot{z}_*^{(2)} = x_*^{(2)} y_*^{(2)} - \frac{8}{3}z_*^{(2)} + CM(z_*^{(1)} - z_*^{(2)}).\end{aligned}\quad (14)$$

We can now consider the system consisting of (one copy of each of) (13) and (14). The solutions of this combined system and the Jacobian at these fixed points can be found using Mathematica. From the eigenvalues of this Jacobian conclusions about the stability of the fixed points can be determined.

4.3 Stability analysis

We did the stability analysis for (13) and (14) for the parameter values $C = 0.6$, $N = 8$ and $M = 12$ and also for $C = 0.2$, $N = 1$ and $M = 19$.

For the first parameter set, $C = 0.6$, $N = 8$ and $M = 12$, we find 9 fixed points, of which 2 are stable. The fixed points can be categorized as follows: 3 unstable fixed points at the unstable fixed points of the original single system

(i.e. at the origin and wings)¹. There are also emergent fixed points that arise through the interactions between the oscillators. These can be separated into a pair of stable fixed points on the wings and two pairs of unstable fixed points with one group of oscillators on a wing and another group close to the origin. The stable fixed points on the wings also emerged in the simulations done in [4]. In figure 4 we depict all fixed points, together with a sample trajectory of one Lorenz63 model for reference.

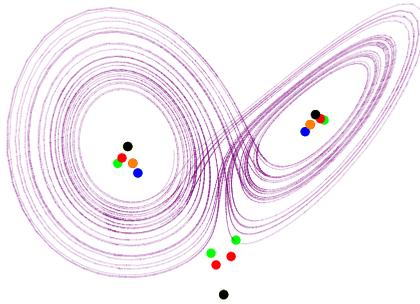


Figure 4: The fixed points for the parameters $C = 0.6$, $N = 8$ and $M = 12$. The single system fixed points are in black, the unstable emergent fixed points where N oscillators can be are green, the unstable emergent fixed points where M oscillators can be are red, the stable emergent fixed points where N oscillators can be are blue, the unstable emergent fixed points where M oscillators can be are orange and a sample trajectory of one Lorenz63 model is depicted in purple.

For the second parameter set, $C = 0.2$, $N = 1$ and $M = 19$, we find only 5 fixed points, none of which are stable. We retain the fixed points of the single system (which should not be too surprising) and we also retain two emergent unstable fixed points, where there are 19 oscillators close to the origin and

¹This corresponds to all systems being synchronized at an identical fixed point, such that the couplings terms are all zero.

one oscillator in a wing. Since there are no stable fixed points here, it is not surprising that configurations in which 19 oscillators balance out one have not been observed in the simulations [4]. In figure 5 we again depict all fixed points and a sample trajectory of one Lorenz63 model for reference.

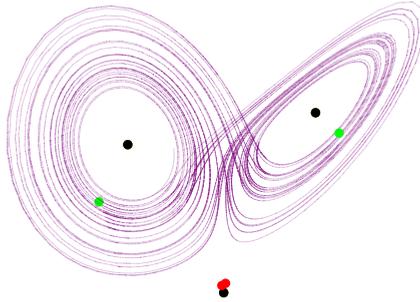


Figure 5: The fixed points for the parameters $C = 0.2$, $N = 1$ and $M = 19$. The single system fixed points are in black, the unstable emergent fixed points where N oscillators can be are green, the unstable emergent fixed points where M oscillators can be are red and a sample trajectory of one Lorenz63 model is depicted in purple.

Our analysis can in principle be applied to any values of C , N and M in order to find any possible ghost attractors. We have found that for more balanced values of N and M there are more fixed points and they may be stable. We observed that for very skewed distributions (all stable and some unstable) fixed points disappear. This analysis could possibly be extended to more complicated configurations of oscillators than a division into two groups. An extension to more complicated models could also be of practical use. An interesting open question that remains is how the unstable emergent fixed points influence the dynamics of the coupled system.

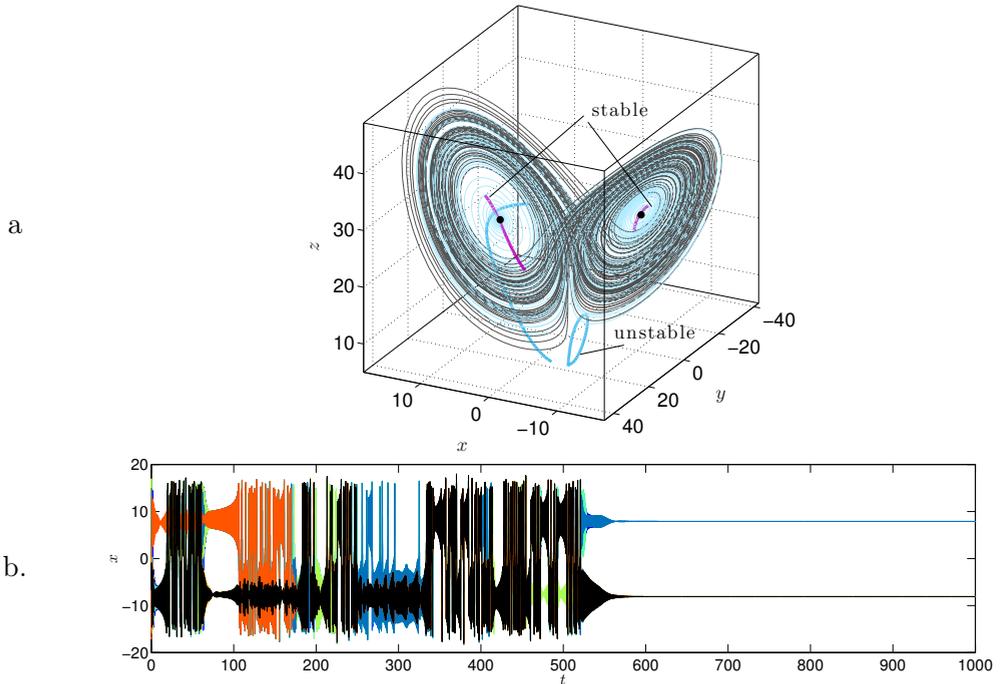


Figure 6: Simulation results for a Lorentz '63 based supermodel. *a.* Integration lines of the synchronised supermodel and two families of critical points that were obtained for various values of coupling strength C : stable (magenta) and unstable (blue). *b.* Time dynamics of x -variables depicts the synchronisation process leading to a steady state solution that is not featured by the original Lorentz '63 model.

5 A search for improved coupling mechanisms

Let a supermodel

$$\dot{x}_{i,j}(t) = L_i(x_{i,j}(t)) - \sum_{k \neq j} C_{j,k}(x_{i,j}(t) - x_{i,k}(t)), \quad i = 1, \dots, m, \quad j = 1, \dots, n; \quad (15)$$

be composed of n instances of autonomous basic models,

$$\dot{x}_i(t) = L_i(x_i(t)), \quad i = 1, \dots, m. \quad (16)$$

Here $C_{j,k} > 0$, are constants, that define the strength of connection for each couple of models. It is known that in some cases the dynamics of the synchro-

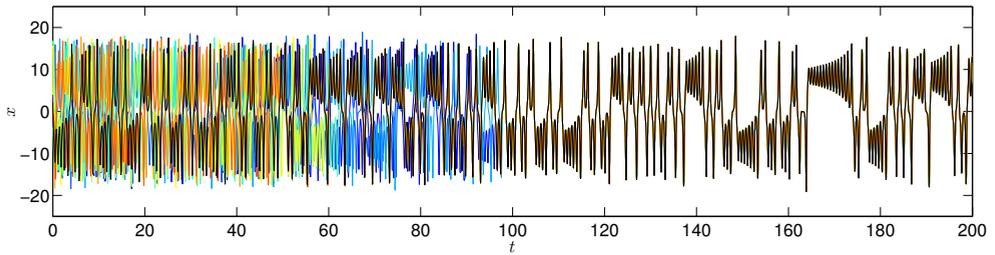


Figure 7: Time dynamics of x -variables of Lorenz '63 based supermodel with Gaussian kernel coupling, The synchronisation process leads to a chaotic but synchronous solution

nised supermodel (15) may be different from the dynamics of the basic model (16). For instance, numerical analysis show extra stable critical points appear in Lorenz'63 based supermodel as in Figure 6. One possible way to overcome the issue is to define the coupling mechanism to be active only locally, when a pair of submodels are close enough in the state space. This can be done by redefining (15) as

$$\dot{x}_{i,j}(t) = L_i(x_{i,j}(t)) - \sum_{k \neq j} C_{j,k} \phi_{j,k}(|x_{i,j}(t) - x_{i,k}(t)|)(x_{i,j}(t) - x_{i,k}(t)), \quad (17)$$

$$i = 1, \dots, m, j = 1, \dots, n;$$

where $\phi(x)$, $x \geq 0$ is a smooth function with a maximum at $x = 0$ that is finite supported or decays on infinity as $o(x^{-1})$. A good candidate for $\phi(x)$ is a Gaussian kernel, $\phi(x) = e^{-x^2/\sigma_{j,k}^2}$. Numerical simulations for a supermodel containing 10 identical Lorenz'63 models coupled with the Gaussian mechanism (17) reveal that although synchronisation process takes more time, the synchronised solution is not a steady state one, as can be seen in Figure 7. Note, the price to pay for using this approach is an extra parameter $\sigma_{j,k}$ that together with $C_{j,k}$ should be estimated by an optimisation/machine learning process.

6 Conclusions

Meteorological institutes are keen to increase the reliability of their weather forecasts. They are faced with the challenge that the underlying mathematical models exhibit chaotic behaviour and are therefore hard to analyze and

solve. To overcome this challenge, the KNMI and partners have developed the super-modeling approach in which a set of models with different strengths and weaknesses are coupled. We were asked to look into this approach and formulate recommendations for its future development. We found that coupling the state space variables in the hyper plane perpendicular to the orbit can be sufficient to obtain synchronization of the different models. The new dynamics of the super model was briefly looked into. Coupling mechanisms that differ from linear nudging were studied and coupling by Gaussians was found to be effective in particular circumstances. Overall, more research is required to obtain a better understanding of the super modeling approach to obtain more reliable weather forecasts.

References

- [1] Ljupco Kocarev. *Consensus and Synchronization in Complex Networks*. Springer-Verlag Berlin Heidelberg, 2013.
- [2] L. A. van den Berge, F. M. Selten, W. Wiegnerinck, and G. S. Duane. A multi-model ensemble method that combines imperfect models through learning. *Earth System Dynamics*, 2(1):161–177, 2011.
- [3] Miroslav Mirchev, Gregory S Duane, Wallace KS Tang, and Ljupco Kocarev. Improved modeling by coupling imperfect models. *Communications in Nonlinear Science and Numerical Simulation*, 17(7):2741–2751, 2012.
- [4] Wiegnerinck et al. Stabilization of ghost attractors in weakly connected networks of chaotic systems. May 2013.

Inferring transportation modes from smartphone sensors

Rob Bisseling (Utrecht University), Jason Frank (Utrecht University), Haris Gavranovic (University of Sarajevo), Jasper van Heugten (Radboud University Nijmegen), Anna Kruseman (Utrecht University), Daphne van Leeuwen (CWI), Christian Reinhardt (University of Leiden)

1 Introduction

Transportation in urban areas poses big challenges related to sustainability, safety and health of residents. A key step to improving policymaking in these respects is to collect and analyse data on how current resources are used, so that inefficiencies may be identified and addressed. The abundance of mobile devices makes it very attractive to harness the advanced data collection abilities of smartphones to tackle this question.

1.1 Mobidot problem statement

Mobidot b.v. develops software that can be implemented in smartphone applications to provide automated capturing and analysis of mobility traces of individuals via smartphone sensors. Their customers are businesses and government organisations interested in quantifying and improving the travel patterns of their employees or constituents, especially by providing those individuals with knowledge that motivates them to make safe, sustainable and healthy mobility choices.

The platform developed by Mobidot utilizes efficient real-time data acquisition using smartphone sensors, coupled with a central analysis server that cleans the raw data and compares it with available databases to infer travel trajectories and modes.

The problem formulated for the Study Group focuses on advancing the quality of data derivation. Mobidot infers the route, role, objective and mode of transportation from smartphone sensor data. Smartphones possess a variety of sensors, including GPS, mobile telephone (4G) and wi-fi signals, accelerometer and gyroscope sensors, etc. that could be used to determine the motion and position of the user, when coupled with geographic databases and known public transport tables.

The volume of monitoring and data recording must be weighed against battery drain, and consequently a sensing strategy must be devised to optimise information gathering with minimal energy usage. The first objective of the problem posed to the Study Group was to optimise data measurement versus battery usage, by (1) devising an optimal scheduling plan for sensing, detecting mode changes, (2) developing a method to locally filter and compress relevant information at the mobile site, and (3) developing a method to infer motion from incomplete data. A second objective posed to the Study Group was to detect obvious errors in the trip analysis and minimise false inferences.

For the Study Group Mathematics with Industry, Mobidot provided relevant data to develop data deduction improvements and test approaches and methods. This included sample smartphone multi-sensor data, sensor energy usage stats and samples of resulting anonymised mobility profiles.

1.2 Work plan of the Study Group

The Study Group initially began work on several fronts and eventually converged on two most promising lines of investigation. The first of these was an improvement to transportation mode identification using high-resolution accelerometer measurements to try to identify transportation mode signatures from vibrational data (bicycling rhythms, motor frequencies, etc.). Related to this, the second investigation line was a sensing strategy that would accommodate taking such high-resolution accelerometer readings without excessively straining battery charge.

In this paper we propose approaches to dealing with the above challenges. In Section 2, we focus on devising a sensing strategy that is both energy efficient and provides enough data to sense multimodal traveling. This in particular necessitates the detection of changes in the transportation mode. To this end high resolution data is crucial to reliably deduce changes. We propose to

supplement the currently used GPS/localization sensing with high-resolution accelerometer data in a three stage sampling procedure. This keeps both energy consumption low and prediction power high.

Once data is acquired it is necessary to infer motion signatures from it. This means that we need to process the time series data obtained from the measurements in a way that enables us to reliably distinguish between different modes of transportation. Current approaches use for example frequency analysis via Fourier transform methods. Here we investigate the use of Wavelet transform methods that provide local frequency information on time series signals. Using wavelet analysis, each time signal is efficiently converted into a distinct two dimensional signature, which in turn could be used to train a learning algorithm to distinguish different transport modalities. In Section 3 we discuss the use of Haar wavelets in some detail and stress how they are useful to detect characteristic changes in time series data. In Section 4 we show some sample applications of this method to accelerometer data taken from different modes of transportation. A powerful feature of the Wavelet approach is the great variety of available basis wavelets that enable one to look for changes with specific structures. We use as a second example the Mexican hat wavelet to analyse the same accelerometer data and discover characteristic structures for the different modes of transportation. The results suggest that further research might lead to powerful prediction tools via the 'right' choice of wavelet. Finally, in Section 5, we mention a method to determine the specific moment of modality change from accelerometer data stored on the phone.

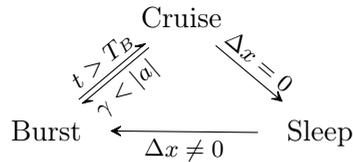
2 Sampling algorithm

In subsequent sections we discuss ideas about improving modality sensing by incorporating high-frequency accelerometer data. Modality changes are rare, however, and it is unnecessary to continually probe and analyse accelerometer data throughout a commuter trajectory. At the same time, sampling should be done as sparsely as possible to minimize battery drain. In this section we discuss a possible scheme for acquiring high resolution accelerometer readings through irregular short bursts.

We propose 3 modes of data sampling:

- burst: high frequency sampling for a fixed short duration
- cruise: variable frequency sampling
- sleep: low frequency sampling, only when stationary

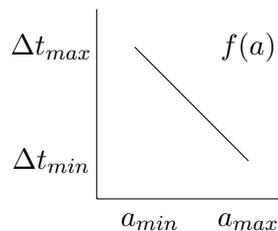
Switching between the modes is indicated in the scheme below:



Burst mode is a new feature for Mobidot and its main purpose is to enable high frequency sampling for modality detection from accelerometer data. In this mode, both accelerometer data and GPS data are collected. The Burst mode lasts for a fixed amount of time T_B seconds and the time step between two consecutive measurements is short Δt_B . After Burst mode, the system always goes to Cruise mode.

Cruise mode is a medium frequency sampling mode to track the trajectory and to detect sudden changes, which call for a change back to Burst mode. The sampling frequency of cruise mode is adaptive, depending on the change in acceleration as derived from the GPS data. When Cruise mode is evoked after a Burst, the time step starts at Δt_{min} .

Each time step later, the stepping time is adapted: $\Delta t = \max(\Delta t_{min}, f(a))$. The function f is linear with acceleration a in m/s^2 . So $f(a) := \Delta t_{max} - c * (a - a_{min})$, where $c = \frac{\Delta t_{max} - \Delta t_{min}}{a_{max} - a_{min}}$. This is shown schematically in the figure below.



In Cruise mode only GPS data is collected. The system can leave Cruise mode when:

- location is stationary for a time span longer than T_S : go to Sleep mode
- there is a sudden change in acceleration $|a| > \gamma$: go to Burst mode

Sleep mode is only invoked when the device is stationary. This mode is characterised by low frequency sampling Δt_S . When significant location change is detected, the system goes immediately into burst mode, to detect the modality of the new trip.

The mode switching is demonstrated in the following algorithm:

```

if mode==Burst
     $\Delta t = \Delta t_B$ 
    if  $T > T_B$ 
        mode =Cruise
    end
elseif mode ==Cruise
     $\Delta t = \max(\Delta t_{min}, f(a))$ 
    if  $\Delta x == 0$  and  $T_{stationary} > T_S$ 
        mode=Sleep
    elseif  $|a| > \gamma$ 
        mode =Burst
    end
elseif mode==Sleep
     $\Delta t = \Delta t_S$ 
    if  $\Delta x > 0$ 
        mode =Burst
    end
end

```

3 Discrete Wavelet Transform

In a practical application such as Mobidot’s smartphone app, a continuous function f is often sampled at a finite set of discrete time points. Let us define the function values as $x_i = f(t_i)$ where the $t_i = i\Delta t$, $i = 0, \dots, n - 1$ are n equidistant sample times.

Time series analysis is a large field. The Study Group has investigated the use of wavelet analysis as a potentially efficient means of transforming discrete accelerometer data into a form suitable for training a learning algorithm to distinguish transport modality.

Analysis of a discrete time series requires a discrete version of the wavelet transform, and we will explain this for the simplest possible wavelet, the Haar wavelet. This wavelet is based on computing sums and differences of neighbouring function values, thereby representing both averages and differences. The *averages* give a smoothed version of the time series, with hopefully more reliable statistics, whereas the *differences* reveal whether a significant change has occurred over time.

One level of the wavelet transform is defined by

$$y_{2i} = x_{2i} + x_{2i+1}, \quad y_{2i+1} = x_{2i} - x_{2i+1}, \quad \text{for } i = 0, \dots, n/2 - 1. \quad (1)$$

We denote this by $\text{dwt1}(\mathbf{x}, n)$, where the output \mathbf{y} overwrites the input \mathbf{x} . We can also carry this out on the first k components of the vector \mathbf{x} of length n , in which case we write $\text{dwt1}(\mathbf{x}, k)$.

In the complete Discrete Wavelet transform (DWT), all differences are recorded and then they remain unchanged afterwards. For the sums, however, the procedure is repeated, but now with half the previous length. This is facilitated by first permuting the vector \mathbf{y} by an even-odd sort, giving

$$z_i = y_{2i}, \quad z_{i+n/2} = y_{2i+1}, \quad \text{for } i = 0, \dots, n/2 - 1. \quad (2)$$

We denote this by $\text{sort}(\mathbf{y}, n)$, again assuming the output \mathbf{z} overwrites the input \mathbf{y} . The complete DWT is given as Algorithm 1.

Algorithm 1 Discrete Haar wavelet transform

Require: \mathbf{x} : vector of length n , with $n = 2^m$.

Ensure: \mathbf{y} : vector of length n , $\mathbf{y} = \text{DWT}(\mathbf{x}, n)$.

while $n > 1$ **do**

$\text{dwt1}(\mathbf{x}, n)$;

$\text{sort}(\mathbf{x}, n)$;

$n := n/2$;

return \mathbf{x} ;

As a result of executing the DWT, we obtain an output vector \mathbf{y} with

$$y_0 = x_0 + \dots + x_{n-1}, \quad (3)$$

so that y_0/n equals the average of all the input values. The next value

$$y_1 = (x_0 + \dots + x_{n/2-1}) - (x_{n/2} + \dots + x_{n-1}) \quad (4)$$

indicates whether a significant change in mean value can be detected between the first half of the time series and the second half. The other values y_i give such change information at more detailed levels of accuracy.

The total number of additions and subtractions in the DWT algorithm for the Haar wavelet equals $n + n/2 + \dots + 2 \approx 2n$, which is significantly smaller than the $5n \log_2 n$ floating-point operations (additions, subtractions, and multiplications) that would be needed for a standard radix-2 Fast Fourier Transform. For example, for $n = 1024$ the Haar wavelet transform is a factor 25 cheaper than the FFT. Note that the Haar wavelet does not need any multiplications (in contrast to the commonly used Daubechies wavelet), making it particularly cheap. Its use would lead to a much lower energy consumption in case the transform is computed on a smartphone.

4 Application of wavelet analysis

A Fourier transform of a signal in time only gives information on what frequencies are present in the total signal and thus the time-domain is lost. The wavelet transform provides a way to preserve the time-domain while also obtaining information about the frequency domain. The wavelet transform is most easily understood from the formula of the continuous wavelet transform

$$\Psi(\tau, s) = \frac{1}{\sqrt{s}} \int f(t) w\left(\frac{t - \tau}{s}\right) dt,$$

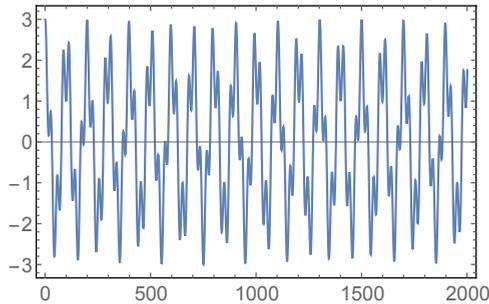
where τ is the location of the window over which we integrate, s is the scale, and $w(t)$ is the wavelet, for example the Haar wavelet as shown in Figure 1.

In this section we apply the theory of the wavelet to the accelerometer data collected by Study Group. However, first we apply wavelet analysis to an example.

Consider the signal composed of two cosine functions

$$f(t) = 2 \cos(20\pi t) + \cos\left(50\sqrt{2}\pi t\right). \tag{5}$$

This function is sampled on $t \in [0, 2]$ with steps of size 10^{-3} , leading to $n = 2000$ data points:



The wavelet transform is performed with the Haar wavelet using Mathematica and is shown in the lower left in Figure 1. Each box, numbered 1 to 7 in the plot, contains the same t -axis shown in the above image, namely $t \times 10^3 \in [0, 2000]$. As explained each level of refinement corresponds to convolution with the Haar wavelet at a different scale. The signal is thus compared to the Haar wavelet at each scale. In turn, each scale can be seen as a trade-off between frequency resolution and time resolution. In Figure 2 we compare the wavelet-transforms of each of the two cosines in the function $f(t)$ individually. Here we can see that at the sixth level of refinement there is a clear distinction between the two. The weight in the Haar transform of the complete function thus indicates that on this scale of the data the function contains a prominent slowly oscillating component. For each scale we can determine the energy fraction: $\{0.003, 0.012, 0.046, 0.138, 0.228, 0.385, 0.156, 0.0321\}$, where the energy of each scale is determined by the sum of squares of the values. The energy fraction can be used to rank the dominant contribution in the signal.

Although the Haar-wavelet transform is computationally very efficient, other wavelets can be tailored to find specific signals in data. For example, if we were to use the Meyer-wavelet the two oscillatory signals are clearly distinguishable as shown on the right-hand side of Figures 1 and 2. This suggests that we might use wavelets to pick out certain features in the Mobidot data specific to bicycles, trains or buses, by optimizing the wavelet basis for specific transport modes.

Next we apply the wavelet transform to the one-component of the 3-axis accelerometer data acquired by team member Jason Frank using a third party smartphone App. It is important to stress that a much more thorough data acquisition program is needed to characterise transport modes accurately. Here we provide only a random sample of time series data to indicate that differences can be discerned. Furthermore, our sampling rate was approximately 100–150 samples per second, which may be too low for detecting mechanical

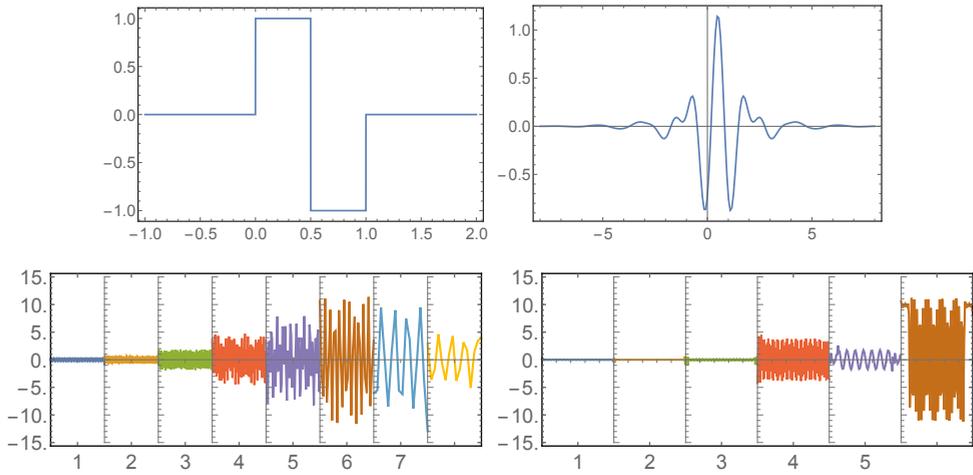


Figure 1: Left: The Haar wavelet (top) and the Haar transform of $f(t)$ (bottom). Right: The Meyer wavelet (top) and the Meyer transform of $f(t)$ (bottom).

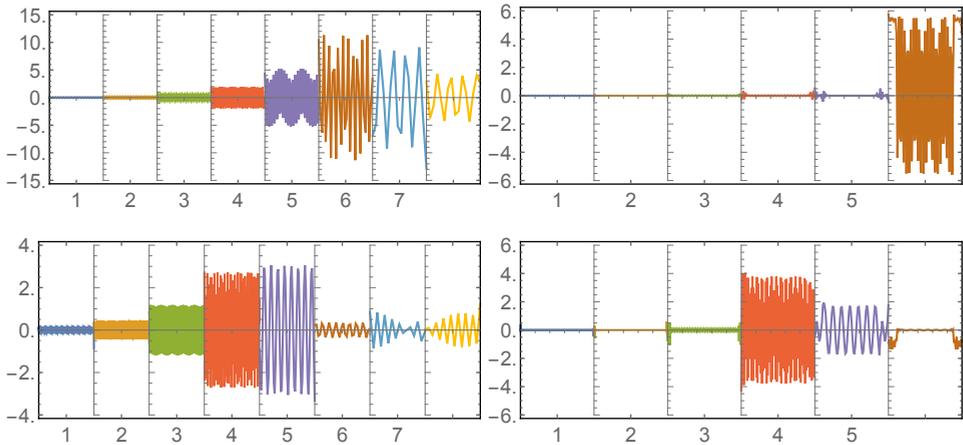
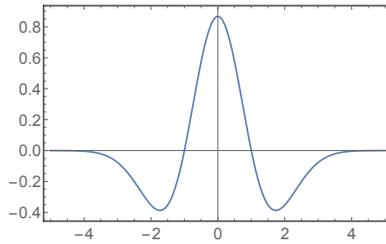


Figure 2: Left: The Haar-wavelet transform of the first (top) and second (bottom) term of $f(t)$. Right: Similarly, but for the Meyer wavelet.

vibrations in automobiles and buses. For visualisation purposes we will use the continuous wavelet transform with the Mexican-hat wavelet:



The time series data and its wavelet transform are shown in Figure 3. The results are shown with the scale (refinement) on the vertical axis and time on the horizontal axis. Each horizontal line in the image shows the absolute value of the wavelet transform with darker colours corresponding to larger magnitudes (cf. the amplitudes of the functions shown in Figure 2). Thus the image shows the dominant contribution to the wavelet transform at a particular scale.

It is clear that the bicycle shows distinctive features when analysed with this wavelet, which indicates that bicycles might be discernible. The car in general has high energies at low frequencies, while the train and bus are relatively quiet. A possible way to train mode-detection using wavelet analysis on the Mobidot data would be to record a sample of accelerometer data of fixed length of time, and subsequently apply the Haar transform to see if distinctions between different modes of transport can be identified based on which scales contain the largest energy fraction. If the Haar wavelet does not provide a clear distinction between modes of transport one could start to train on different wavelets. Using combinations of Haar wavelets one could potentially develop a training method which varies the wavelet form until an optimum wavelet is found. However, since Mobidot would like the wavelet transform with the optimum wavelet to be implemented on the mobile device, there will in general be a competition between the ease of detection and the computational cost.

5 Determine specific moment of transportation modality change

Continual recording of GPS location would prohibitively drain smartphone battery charge. Therefore Mobidot collects GPS data at regular intervals depending on detected movement. As a result it is hard to determine the exact starting time of a trip. If a change is detected one can only guess the starting

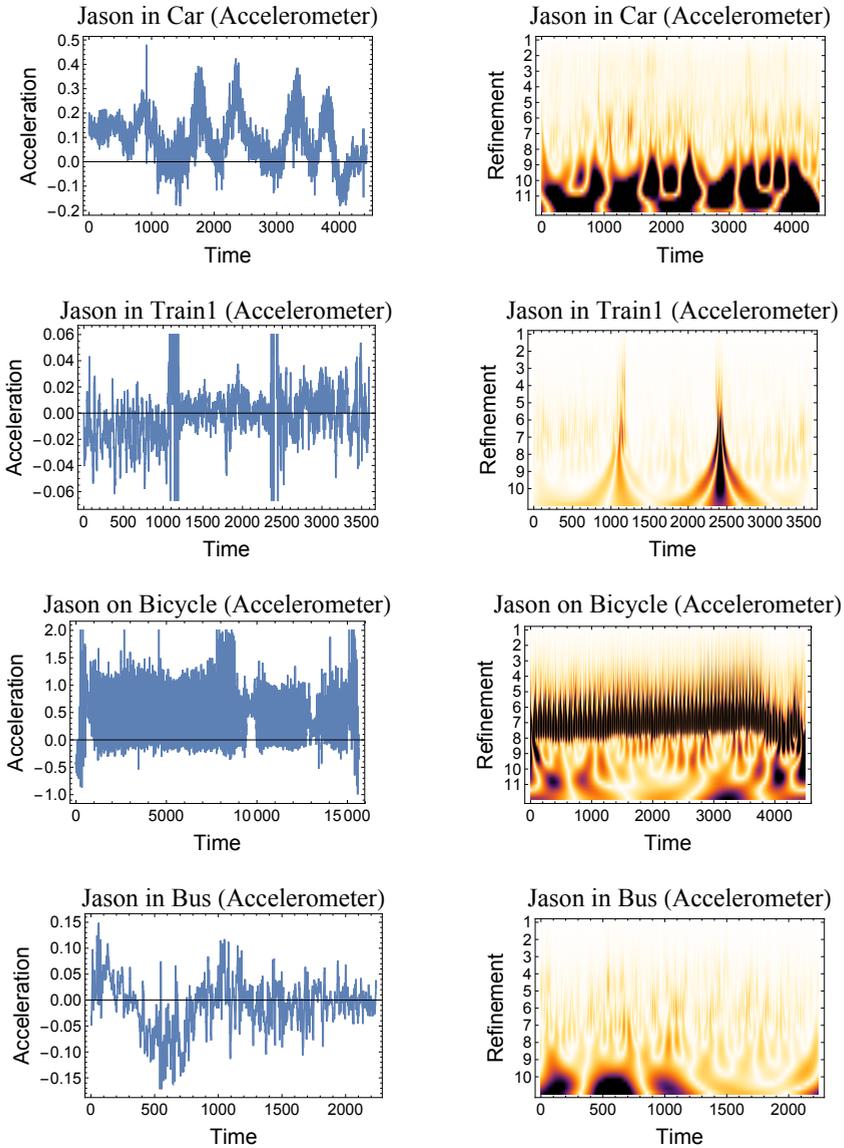


Figure 3: Accelerometer data and Mexican-hat wavelet transform for four different modes of transport.

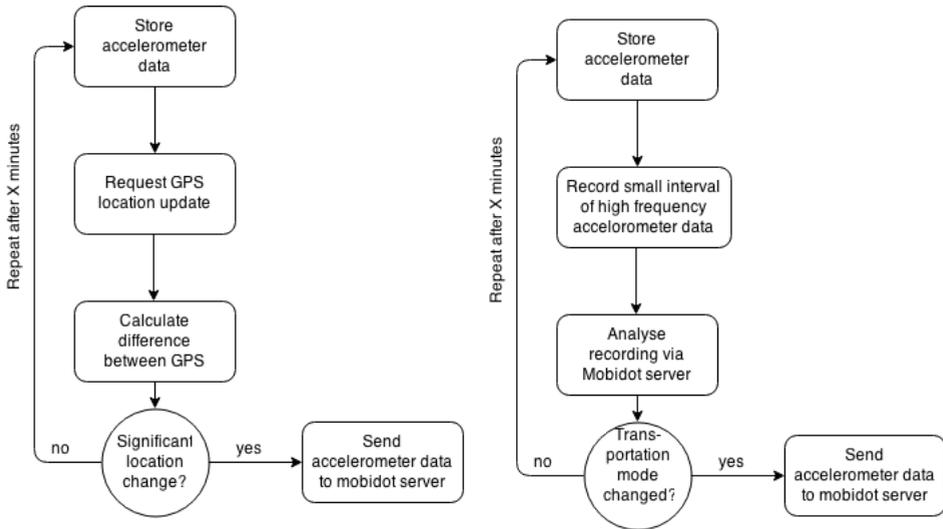


Figure 4: Visual representation of mode change detection via accelerometer sensor.

time via interpolation methods.

To overcome this problem the Mobidot application could augment GPS update requests with accelerometer data. The accelerometer in smartphones is a background application that is used for to detect gestures in the device. One can think of horizontal or vertical display switching or the natural 3D shadow effects for icons. Assuming the accelerometer is already running continuously, we can collect and store this data temporarily. After a certain time interval, the location detector can give an update about the current location. When no location change is detected, the accelerometer data can be deleted. However, if there is a significant change between the current and previous location we can analyse the accelerometer data of that time interval. This time interval can be analysed via the wavelet transform proposed in Section 4 to determine the moment of mode change.

This method can be used to detect the start of a trip, but it can also detect a change in transportation mode. To give a better overview of the method a graphical representation is shown in Figure 4. The left figure indicates the update method when the phone is in stationary mode. The right one represents a regular check to see changes in transportation type within a trip. Via small bursts of high frequency accelerometer data as described in Section 2 a mode

change can be detected. If a change is detected we can use the temporarily stored accelerometer data from the phone and track at which moment the change took place.

6 Conclusions

We have reported on preliminary research to improve transport modality sensing using smartphone data acquisition. Our primary conclusions are:

- High resolution accelerometer data exhibits noticeable differences among different modalities such as bicycle, automobile, bus and train. Possibly this approach could be combined with currently used location service data to improve modality inference.
- Wavelet transforms offer an inexpensive and potentially powerful approach to obtain local frequency information on accelerometer signals. A more thorough multi-scale application of wavelets yields distinctive pictures of transport signals, that could be used to train learning algorithms.
- Accelerometer data may be effectively sampled in short, high-resolution bursts. These acquisitions can be incorporated in a multi-phase sensing strategy to preserve battery charge.
- The wavelet approach and multi-phase sensing strategy can be combined to improve the detection of mode changes during transit.

Adapting radiotherapy treatment to deformations in the patient

Claudiu Antonovici (CWI Amsterdam, Leiden University), Tugce Akkaya (TU Delft), Arjen Baarsma (Utrecht University), KaYin Leung (Utrecht University, UMC Utrecht), Corine Meerman (Leiden University), Lisanne Rens (CWI Amsterdam, Leiden University)

Abstract

The physical anatomy of cancer patients who receive radiotherapy treatment may change over time due to e.g. weight loss and tumor deformation. The treatment plan for the patient is based on a CT scan that is made approximately one week prior to treatment. However, when the anatomy of the patient has changed, the treatment plan may no longer be optimal. How can adjustments to the treatment plan be made when changes in the anatomy of the patient are observed? In this paper we report on this ‘cancer treatment’ problem posed by the Netherlands Cancer Institute. We formulate the problem as an optimization problem. Different deformations are investigated for a toy model and optimization methods are tested on this model. We consider three different optimization algorithms, with a main focus on the Gauss-Newton method. This method turns out to work relatively well for some specific deformations in our simple model. Other methods are also considered and their advantages and disadvantages are described, keeping in mind more realistic situations than we consider here.

keywords: adaptive radiotherapy, dose planning, deformations, optimization, Gauss-Newton, Particle Swarm

1 Introduction

This paper reports on findings of the ‘cancer treatment’ problem, as posed by the Netherlands Cancer Institute (NKI) at the Study Group Mathematics with Industry 2015, held at Utrecht University. Every year, 469 per 100.000 inhabitants over 18 years in The Netherlands are diagnosed with cancer. Around

fifty percent of cancer patients are still alive five years after the diagnosis and the number of treatment with radiotherapy increases [5]. The Netherlands Cancer Institute (NKI) is a national center dedicated to cancer and plays an important role as a national and international center of scientific and clinical expertise, development and training. One of their ongoing research topics is on improving the efficacy of radiotherapy treatment plans.

If a person is scheduled to receive radiotherapy treatment, a CT scan is made prior to the treatment. The time between the CT scan and the actual treatment is usually between one and two weeks. The CT scan is made to obtain detailed information about the tumor and the surrounding area. Using this information, an accurate though time intensive optimization method is used to acquire the optimal settings of the machine and the corresponding dose distribution. To explain these terms, a closer look at the treatment settings is needed.

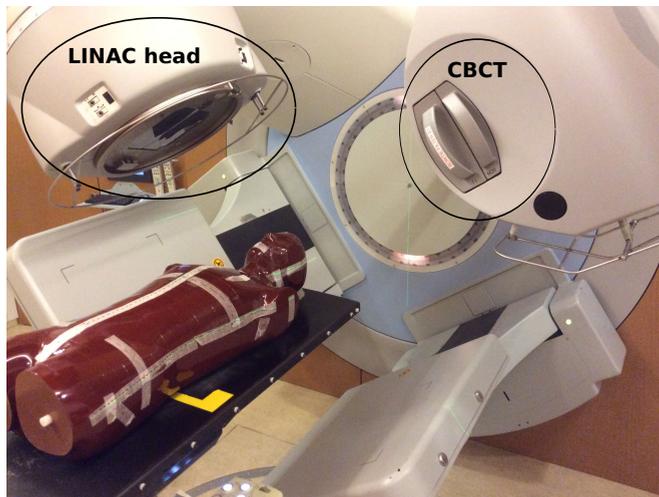


Figure 1: Treatment settings. Image courtesy of Rene Tielenburg.

Figure 1 shows the treatment set-up. In this set-up the patient lies on the treatment table. The radiation source is located in the LINAC (linear accelerators) head. Within this head there are leaves, the settings of which determine the shape of the beam that radiates the patient. This head rotates around the table. At every rotation step, the settings of the leaves within this device, are adjusted such that after a full rotation the optimal dose is delivered. An optimal treatment is defined to mostly affect the tumor, whereas the surrounding tissue, especially the vital organs, should be left alone as much as possible to

avoid damaging healthy tissue.

The actual treatment based on the treatment plan derived from the CT scan ideally starts one (but maximum two) week(s) after the CT scan was made. In this period, the anatomy of the patient may have changed, causing a lot of differences compared to when the CT scan was taken, such as changes in the size of the tumor, weight loss of the patient, posture changes and many other deformations and changes. Thus, applying the original treatment plan to the patient may lead to a sub-optimal dose distribution.

To avoid large amounts of healthy tissue being targeted by radiotherapy, the current procedure is to take another scan of lesser quality of the patient right before the treatment starts. This scan is taken with the imaging device CBCT (cone beam CT) which is part of the treatment device (Figure 1). If the outcome of this scan differs substantially from the data of the CT scan, the treatment is stopped and the entire procedure starts all over, i.e. another CT scan and subsequently a new treatment plan is made. If the deformations are small, the original treatment is applied to the patient.

NKI asked us to develop or propose a method that determines the optimal settings of the leaves using the initial correct dose, the initial data from the CT scan and the data of the CBCT. These new settings have to be obtained within a few minutes because the treatment has to be applied directly after the scan of the CBCT.

In Section 2, we formulate the problem as an optimization problem. In section 3 three different methods to solve this optimization problem are proposed to solve this type of problem: the Gauss-Newton method, evolutionary algorithms (with a focus on the Particle Swarm optimization algorithm), and a gradual deformation based approach. In Section 4, four different tumor-deformation scenarios are considered and the Gauss-Newton is used to solve the optimization problem for these scenarios.¹ The Particle Swarm optimization is applied to one of the four scenarios and compared to the result obtained with Gauss-Newton. Finally, in Section 5 conclusions and recommendations are presented.

¹After the study week it was discovered that there was a mistake in the code that we had worked with. We corrected this afterwards and the results presented in these proceedings are the results for the corrected code. As a bonus, finding this mistake also greatly improved our results.

2 Formulation of the problem

2.1 The optimization problem

The problem put forward by the NKI is essentially an optimization problem, so we formulated it as such. More specifically, we approached it as a non-linear least squares problem of the form

$$\arg \min_{\phi} \|D(\phi, P_1) - D_{\delta}\|^2. \quad (1)$$

Here ϕ denotes the configuration of the linear accelerator, P_1 is a set of parameters describing the patient during treatment and $D(\phi, P_1)$ is the dose delivered to this patient when the configuration ϕ is used. The dose D_{δ} , which we try to approximate, is similar to the dose absorbed by the undeformed patient under the original plan, but has been transformed using the deformation of the patient

We proceed by providing a more detailed description of the variables ϕ and P , as well as the dose function D and the target dose D_{δ} .

2.2 Variables and parameters

During a treatment session, the LINAC head moves around the patient, stopping at N_{stops} different positions along the way. At each position, the opening formed by the multileaf collimator can be adjusted by changing the positions of its $2N_{\text{leaves}}$ leaves. We denote the positions of the j -th pair of leaves while the head is at position i by α_{ij} and β_{ij} , and the time the head stays at position i by τ_i (Figure 2). These variables are subject to the linear constraints $\alpha_{\min} \leq \alpha_{ij} \leq \beta_{ij} \leq \beta_{\min}$ and $\tau_i \geq 0$. For the machine used by the NKI, $N_{\text{stops}} = 90$ and $N_{\text{leaves}} = 40$, which means that the total number of variables is $2N_{\text{stops}}N_{\text{leaves}} + N_{\text{stops}} = 7290$. This number can be doubled by having the LINAC head make two passes, so that it stops at every position twice.

For our purposes, the patient is effectively described by a density function $\rho(\mathbf{r})$ and an attenuation function $\mu(\mathbf{r})$. Together with the configuration of the device, these determine how much energy is absorbed at every point in the patient's body. For the sake of brevity, we write $\phi = (\alpha, \beta, \tau)$ for the full configuration of the linear accelerator and $P = (\mu, \rho)$ for the relevant data concerning the patient.

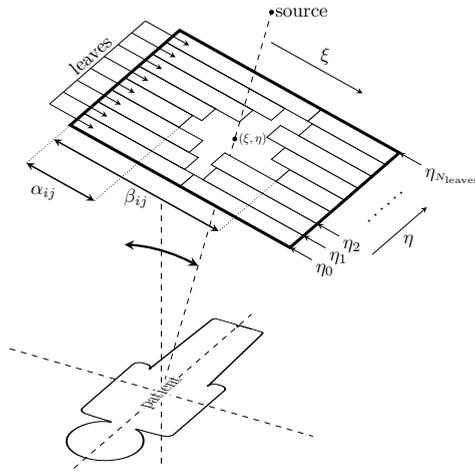


Figure 2: Overview of the leaf configuration.

2.3 The dose function

The total dose absorbed by a point \mathbf{r} inside the patient's body during a treatment session is the sum of the contributions of all rays passing through the multileaf collimator.

If we ignore partial transmission through the rounded edges of the leaves, the total dose absorbed at a point \mathbf{r} is given by

$$D(\phi, P)(\mathbf{r}) = \sum_{i=1}^{N_{\text{stops}}} \tau_i \sum_{j=1}^{N_{\text{leaves}}} \int_{\alpha_{ij}}^{\beta_{ij}} \int_{\eta_{i-1}}^{\eta_i} d_i(\xi, \eta, P)(\mathbf{r}) d\eta d\xi, \quad (2)$$

where $d_i(\xi, \eta, P)$ is the dose delivered per unit time by an infinitesimally small (conical) beam going through the point (ξ, η) in the collimator while the emitter is at position i . The dose function is integrated over the opening formed by each pair of leaves, subsequently a sum over all leaf pairs and all emitter positions is taken.

The full expression for the infinitesimal dose $d_i(\xi, \eta, P)(\mathbf{r})$ also involves multiple integrals, so finding the full dose profile $D(\phi, P)$ in this way can be quite time consuming. Since this will need to be done at least several times during the optimization process and time is a relevant constraint, the use of approximate methods is in order. Two popular methods are the collapsed cone algorithm,

which is described in detail in [1], and the analytic pencil beam kernel approach detailed in [6].

We have employed the analytic pencil beam kernel approach, which is the simpler of the two. This method approximates $d_i(\xi, \eta; P)$ by a simple analytic expression. The approximation relies on a large number of assumptions, both about the patient and the emitter: the patient is assumed to be homogeneous and flat, and the emitter is assumed to produce a beam which is parallel, uniform and incident normally to the patient. Nevertheless, this approach appears to be accurate to less than 2% when applied to actual treatment plans in regions where the dose gradient is small [6].

2.4 The deformation

The CT scan made prior to treatment is used to determine the parameters $P_0 = (\mu_0, \rho_0)$ that best describe the patient at this time. The attenuation μ_0 is measured directly, and ρ_0 can be inferred from knowledge about human anatomy. This data is used to determine a treatment plan, which in particular involves a configuration ϕ_0 for the linear accelerator such that the applied dose $D_0 = D(\phi_0, P_0)$ is suitable for treatment of the tumor.

The problem is that this preliminary scan may no longer be accurate during treatment because the patients will have naturally undergone some deformations of the types described in the introduction and is now described by a new set of parameters (P_1, μ_1) . A new scan can be made on the spot to determine the new attenuation profile μ_1 , which can in many situations be used to derive a displacement vector field $\boldsymbol{\delta}$ that describes the deformation that the patient has undergone to go from P_0 to the new configuration P_1 . This vector field assigns to every point \mathbf{r} in the deformed patient a displacement vector $\boldsymbol{\delta}(\mathbf{r})$ such that the corresponding point in the original patient is, at least approximately, $\mathbf{r} + \boldsymbol{\delta}(\mathbf{r})$. More specifically, it is such that the set of parameters $P_\delta = (\mu_\delta, \rho_\delta)$, with

$$\mu_\delta(\mathbf{r}) = \mu_0(\mathbf{r} + \boldsymbol{\delta}(\mathbf{r})), \quad \rho_\delta(\mathbf{r}) = \rho_0(\mathbf{r} + \boldsymbol{\delta}(\mathbf{r})),$$

closely approximates P_1 .

2.5 The objective function

Since solving the original optimization problem again for the deformed situation is unfeasible due to time constraints, another approach is called for. The approach we took is to use the deformation field δ to deform the original dose distribution $D_0 = D(\phi_0, P_0)$ along with the patient, and to attempt to find a new configuration ϕ such that the delivered dose $D(\phi, P_1)$ approximates the deformed distribution D_δ as closely as possible. This deformed dose distribution is defined in the same way as the deformed density and attenuation, and is thus given by

$$D_\delta(\mathbf{r}) = D_0(\mathbf{r} + \delta(\mathbf{r})).$$

While this distribution may not be optimal for the deformed patient P_1 , it should be as acceptable as the original dose distribution D_0 was for the undeformed patient P_0 .

Formulated as an optimization problem, we look for a configuration ϕ , close to ϕ_0 , that minimizes the distance between the corresponding dose $D(\phi, P_1)$ and the target dose D_δ . This is the solution to the non-linear least squares problem

$$\arg \min_{\phi} \|D(\phi, P_1) - D_\delta\|^2, \quad (3)$$

which uses the square integral norm

$$\|\Delta\|^2 = \iiint_V \Delta(\mathbf{r})^2 d^3\mathbf{r}.$$

The precise form of the objective function in equation (3) depends on how the dose distribution D is discretized. If a rectangular grid $\Lambda = \{\mathbf{r}_{ijk} \mid i, j, k\}$ is used, then

$$\|D(\phi, P_1) - D_\delta\|^2 = \sum_{i,j,k} (D(\phi, P_1)(\mathbf{r}_{ijk}) - D_\delta(\mathbf{r}_{ijk}))^2$$

up to some (immaterial) constant factor.

3 Methods

In this section we discuss three different methods to apply to the optimization problem described in Section 2. The three methods are compared to each

other and pro and cons are discussed. Finally, in Section 3.4, we describe the simplified model that we considered during the study week. This model was used in our investigations in Section 4.

3.1 Gauss-Newton type methods

Since the optimization problem (3) is a non-linear least squares problem, one can try to solve it using the Gauss-Newton algorithm, or similar algorithms such as Levenberg-Marquardt. A description of these algorithms can be found in many textbooks, such as [3] or [4]. These algorithms require the computation of the first order partial derivatives of the dose $D(\phi, P)$ with respect to the components of $\phi = (\alpha, \beta, \tau)$, but no second order derivatives. They are fast when applied to problems which are only mildly non-linear, which one might expect to be the case for very small deformations, but can be slow otherwise.

Without any modifications, the Gauss-Newton algorithm is not very reliable as it is not even guaranteed to converge to a local minimum. Damped version of Gauss-Newton, such as the Levenberg-Marquardt algorithm, avoid this problem by effectively taking smaller steps when necessary. Convergence to an insignificant local minimum is an inherent possibility for all algorithms of this type.

In the Gauss-Newton method the dose function $D(\phi, P_1)$ is approximated around the initial guess ϕ_0 by its first order approximation $D(\phi_0, P_1) + J_D(\phi_0, P_1)(\phi - \phi_0)$ and solving the linear least square problem

$$\arg \min_{\phi} \|D(\phi_0, P_1) + J_D(\phi_0, P_1)(\phi - \phi_0) - D_{\delta}\|^2. \quad (4)$$

This process is repeated, taking the solution from the previous iteration as the new initial guess.

Fortunately, the derivatives required for these algorithms are relatively easy to compute, as the parameters α_{ij} and β_{ij} only appear in the boundaries of the integrals in equation (2) and the dose is manifestly linear in τ_i . The partial derivatives with respect to α_{ij} and β_{ij} are given by

$$\frac{\partial D(\phi, P)}{\partial \alpha_{ij}} = -\tau_i \int_{\eta_{i-1}}^{\eta_i} d_i(\alpha_{ij}, \eta; P) d\eta$$

and

$$\frac{\partial D(\phi, P)}{\partial \beta_{ij}} = \tau_i \int_{\eta_{i-1}}^{\eta_i} d_i(\beta_{ij}, \eta, P) d\eta$$

respectively. Computing these should be significantly less costly than computing $D(\phi, P)$ itself as the domain of integration in these expressions is 1-dimensional, rather than 2-dimensional.

The derivatives with respect to the variables τ_i are given by

$$\frac{\partial D(\phi, P)}{\partial \tau_i} = \sum_{j=1}^{N_{\text{leaves}}} \int_{\alpha_{ij}}^{\beta_{ij}} \int_{\eta_{i-1}}^{\eta_i} d_i(\xi, \eta, P) d\eta d\xi.$$

Note that these do not need to be computed separately, as these integrals should already have been evaluated to determine the dose $D(\phi, P)$.

The Jacobian matrix $J_D(\phi, P)$ which contains all of these partial derivatives has $2N_{\text{stops}}N_{\text{leaves}} + N_{\text{stops}}$ columns (which is 7290 for the NKI set-up), while the number of rows depends on the discretization of D . It will have one row for every voxel if a grid is used, and most of its entries will be zero in this case since $\frac{\partial}{\partial \alpha_{ij}}D(\phi, P)(\mathbf{r})$ is only non-zero if the ray from the emitter to \mathbf{r} passes near to the edge of the leaf corresponding to the variable α_{ij} (and similarly for $\frac{\partial}{\partial \beta_{ij}}D(\phi, P)(\mathbf{r})$). This sparsity can be used to speed up the computation of $J_D(\phi, P)$, as well as any matrix operations applied to it.

Closed leaf pairs should be given extra attention, since the point at which they are closed does not affect the dose and is thus rather arbitrary. When it becomes necessary to open a pair of leaves that was previously closed, these algorithms will only discover this if the point at which an opening should be created is already close to the (arbitrary) leaf edge positions. Closed leaf pairs should therefore always be positioned such that their edges project onto a point in the patient where a high dose is required, or close to neighboring leaf pairs which are already open. It may be necessary to consider different positions for such leaf pairs while running the algorithm and opening them where doing so would produce the greatest improvement to the objective function.

Since the derivatives of D with respect to α_{ij} and β_{ij} add up to zero whenever $\alpha_{ij} = \beta_{ij}$, the Jacobian $J_D(\phi, P_1)$ will not have full column rank for configurations with closed pairs of leaves. While this is not hugely problematic, it can lead to the Gauss-Newton algorithm prescribing large (unnecessary) changes to the position of closed leaf pairs. One way to avoid this is by keeping the

center $\frac{1}{2}(\alpha_{ij} + \beta_{ij})$ fixed (and only varying the size of the opening) whenever $\alpha_{ij} = \beta_{ij}$ at any particular iteration step.

Newton's method, which works by repeatedly making a second order approximation of $\|D(\phi_0 + \Phi, P_1) - D_\delta\|^2$ in Φ , might also be worth considering. Unlike the Gauss-Newton algorithm, Newton's method is guaranteed to converge locally and this convergence is significantly faster (quadratic, rather than linear). The major disadvantage to this approach is that it requires taking derivatives of D up to second order. In our situation, at least when using the analytic pencil beam kernel approach, the second order derivative of D should still be manageable. While there are in principle $(2N_{\text{stops}}N_{\text{leaves}} + N_{\text{stops}})^2$ mixed second order partial derivatives, almost all of these are again zero. Note in particular that $\frac{\partial^2}{\partial\alpha_{ij}\partial\beta_{i'j'}}D(\phi, P_1) = 0$ for all pairs (i, j) and (i', j') and that $\frac{\partial^2}{\partial\alpha_{ij}\partial\alpha_{i'j'}}D(\phi, P_1)$ can be non-zero only if $(i, j) = (i', j')$.

3.2 Evolutionary algorithms

The wide variety of tissues in a patient causes inhomogeneities with respect to attenuation properties and the pencil beam function may be not accurate enough. As a result of these complexities, the surface of the objective function may be very irregular, which may cause methods based on Gauss Newton to end up in local minima. This may lead to inappropriate settings for the treatment. Whether this is the case needs to be investigated.

A possible way to deal with such complexities and avoid local minima, if they occur, is by applying other types of optimization algorithms. One could for instance think of evolutionary computation as a class of optimization methods that are metaheuristic with stochastic elements [2]. These methods can be applied to any type of objective function, since they are used as a black box: no derivatives are needed. Examples of these type of methods are evolutionary algorithms or swarm type algorithms. Evolutionary computation generally consists of a set of individuals, where for each individual the fitness (objective function value) is evaluated. The algorithm then selects/replaces/moves/breeds from (depending on the algorithm) these individuals iteratively, to eventually close in on the optimal solution. In the study week we considered one such evolutionary algorithm in more detail, the so-called Particle Swarm optimization (PSO).

The PSO considers a group of individuals, uniformly distributed around an

initial guess of parameter values. For each individual, the objective function is calculated. Then in each iterative step, the position (parameter values) of each individual is updated. An individual moves in the direction of its current velocity, the best position this individual found so far and the best position the complete group found so far, with some randomness included.

3.3 Gradual deformations

In this project, we also designed another method based on the Gauss-Newton method that might avoid drifts to inappropriate solutions and be nearly as fast as the Gauss-Newton method. The idea is based on the fact that there are two scans of the patient: the first CT scan used for the treatment plan and one obtained directly before treatment. So, between those two time points, the patient has deformed slowly. During the optimization procedure, we can artificially deform the original patient situation $P_0 = (\mu_0, \rho_0)$ to the deformed situation $P_\delta = (\mu_\delta, \rho_\delta)$, in N_s steps by using $P_i = (\mu_i, \rho_i)$, with

$$\begin{aligned}\mu_i(\mathbf{r}) &= \mu_0(\mathbf{r} + \frac{n}{N_s}\delta(\mathbf{r})) \\ \rho_i(\mathbf{r}) &= \rho_0(\mathbf{r} + \frac{n}{N_s}\delta(\mathbf{r}))\end{aligned}$$

for $n \in \{1, 2, \dots, N_s\}$. The desired dose distribution D_n at step n is defined as the goal dose distribution at deformation step n :

$$D_n = D_0(\mathbf{r} + \frac{n}{N_s}\delta(\mathbf{r})).$$

Now, every deformation step n , we may use Gauss-Newton (or any another optimization algorithm) iteratively to find the settings ϕ_n that minimize:

$$\|D(\phi_n, P_n) - D_n\|^2. \tag{5}$$

Subsequently, these settings are used as an initial guess for the next deformation step $n + 1$. The concept behind this algorithm is that finding optimal settings for small deformations might be much easier and faster, and we can better take the correct path from original settings to optimal settings, avoiding drifts to an inappropriate solution.

3.4 Toy model

In the study week we focused on a simple model of the optimization problem to test our different methods, in particular the Gauss-Newton algorithm described

in Section 3.1. In this model we assume the following. First of all, the patient is assumed to be a 2D square lattice with a convex-like tumor. The beam is fixed at one point and the beam direction is orthogonal to the patient plane, i.e. the beam is right above the patient. So, we ignore that the LINAC head can travel around the patient with different durations at each stop. The optimization problem then reduces to the optimization of the leaf configuration only. This specific setting that we chose for the problem is visualized in Figure 3 below.

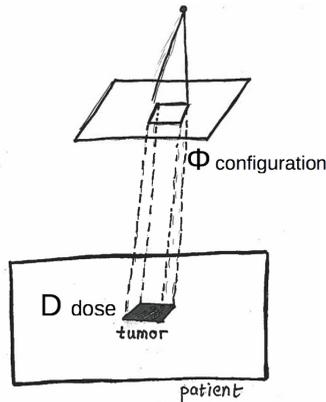


Figure 3: Model setting

Here, for a specific dose requirement D there is a leaf configuration ϕ which ensures that the patient gets the required dose. The goal is then to deform the tumor in a simple way and check whether the algorithm outputs a configuration which gives the same dose distribution, but modified according to the tumor deformation. For example, by simply shifting the tumor to the right as seen in Figure 4, we expect the optimization algorithm to output a configuration for the leaves such that the dose profile follows the same shift.

Aside of the qualitative behavior of the solution, we were also interested in computing the error with respect to some measure. For simplicity, we chose a relative error defined in the following way

$$\text{Err}(\tilde{D}, D) = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M (\tilde{D}(i, j) - D(i, j))^2}{\sum_{i=1}^N \sum_{j=1}^M \tilde{D}(i, j)^2}} \quad (6)$$

where \tilde{D} is the desired dose, D is the dose corresponding to a configuration

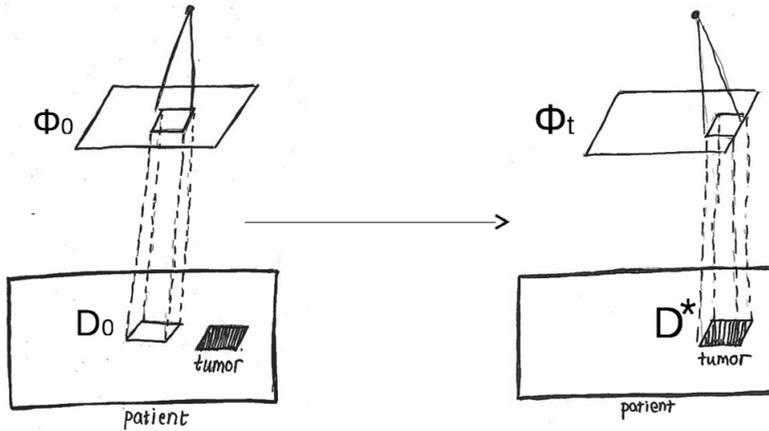


Figure 4: Simple deformation and desired dose profile

given by the optimization algorithm and both N and M correspond to the dimensions of the patient computational domain.

4 Results

In this section we mainly consider the Gauss-Newton algorithm for the optimization problem described in Section 3.4. For the purpose of this project, we chose a 40-by-40 pixels computational domain and a beam emitter with 10 pairs of leaves. Then we assumed for each scenario that at the beginning of the treatment, the patient presents a diamond-shaped tumor. Four qualitatively different deformations of the tumor were considered: vertical shift, lateral shift, diagonal shift, and a shrinkage of the tumor. For each setting, we considered the classical Gauss-Newton algorithms over 20 steps and at each step we recorded the error defined through Equation (6).

Finally, we applied the PSO to the vertical shift of the tumor. For this algorithm we used a matlab code² with 10 individuals and 50 iteration steps.

²*Particle_Swarm_Optimization.m*© Primit Biswas.
<http://www.mathworks.com/matlabcentral/fileexchange>

4.1 Vertical shift deformation

Figure 5 shows the results for a downward shift of the tumor by 4 pixels. A good solution was found after about seven iterations, which took about 46 seconds to run. After that no further improvements were observed, and the relative error between the desired dose and the dose that is actually realised stabilises at about 19.7%. It is uncertain whether the final solution is a global minimum for the objective function. The speed of convergence definitely seems promising for real-life applications.

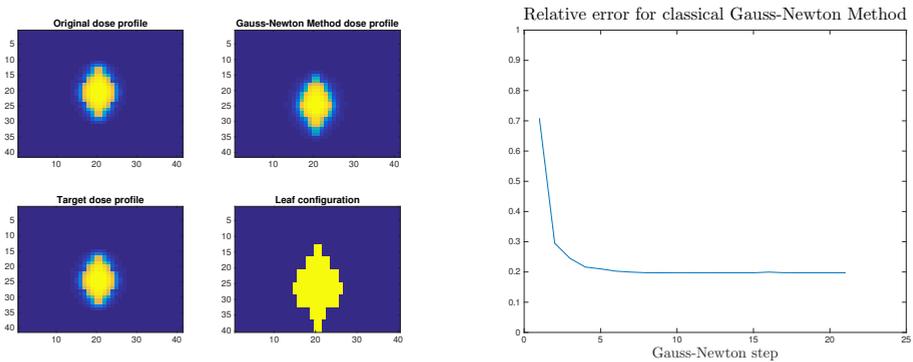


Figure 5: Gauss-Newton algorithm for the vertical shift case of the tumor. The first plot shows the original dose profile, the target profile after the shift, the profile after the last Gauss-Newton step and the configuration of the leaves. The second plot presents the error evolution throughout the iteration steps.

4.2 Lateral shift deformation

For a lateral deformation, we considered a 4 pixel movement to the right. A decent solution is found after about 10 iterations, after a runtime of about 80 seconds. The final error is about 21.9%, as shown in Figure 6. What is interesting to note is that the bottom part of the opening is missing in the final dose leaf configuration because this pair of leaves was closed in the first iteration step at a point which was too far away from the shifted tumor. By moving the closed leaf pair to a point inside the tumor and resuming the algorithm a relative error of about 6.9% is obtained. (This lower error is explained by the fact that a lateral shift of the tumor can be precisely compensated by a corresponding shift of all leaf positions.)

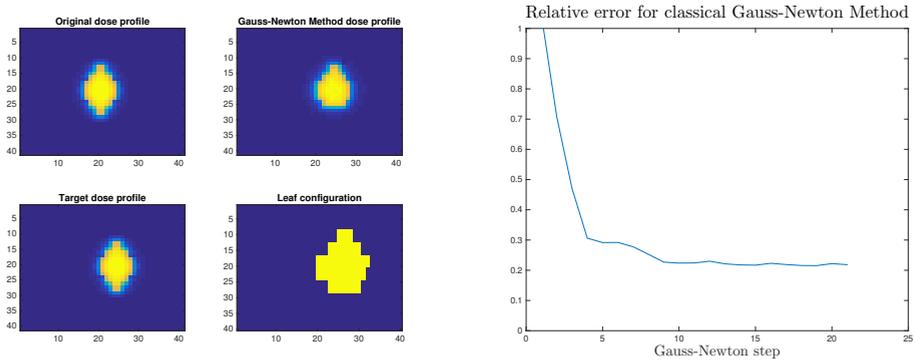


Figure 6: Gauss-Newton algorithm for the lateral shift case of the tumor. The first plot shows the original dose profile, the target profile after the shift, the profile after the last Gauss-Newton step and the configuration of the leaves. The second plot presents the error evolution throughout the iteration steps.

4.3 Diagonal shift deformation

The case of a diagonal shift correspond to a combination of both vertical and lateral changes of the tumor. In our case we chose a 4 pixel deformation in both rightward and downward directions. As expected from the lateral and the vertical case (Figure 7) we see a similar but slightly higher error of 23.8% at the last iteration step after a total runtime of 157 seconds and 20 iterations (the error does not decrease significantly after that). The sudden decrease at the end is due to closed leaf pair that is suddenly opened and could have been found at an earlier stage.

4.4 Size deformation

For the last deformation, we considered a different scenario. Before we only focused on directions in which the tumor can move during the treatment, but in this case we are looking at a size deformation. This particular type of deformation can be justified through the fact that one expects, or hopes, the tumor to shrink during the treatment and thus, the leaf configuration should change accordingly. More precisely, we considered a 50% decrease in size without any lateral or vertical movement involved. After about 7 iterations and 48 seconds of simulation we see the outcome presented in Figure 8. This time the error was about 27.1%.

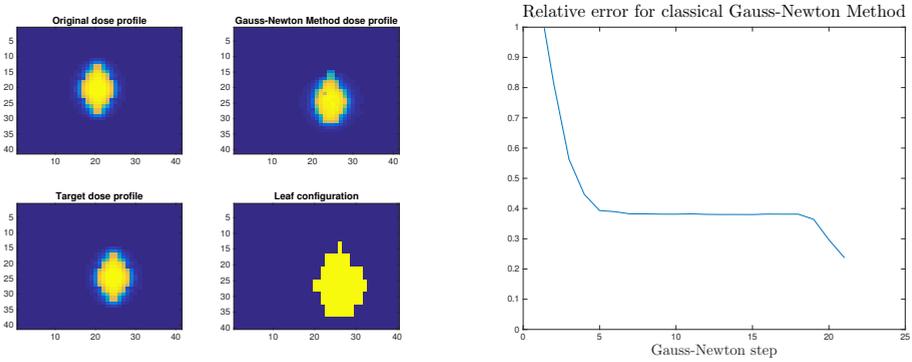


Figure 7: Gauss-Newton algorithm for the diagonal shift case of the tumor. The first plot shows the original dose profile, the target profile after the shift, the profile after the last Gauss-Newton step and the configuration of the leaves. The second plot presents the error evolution throughout the iteration steps.

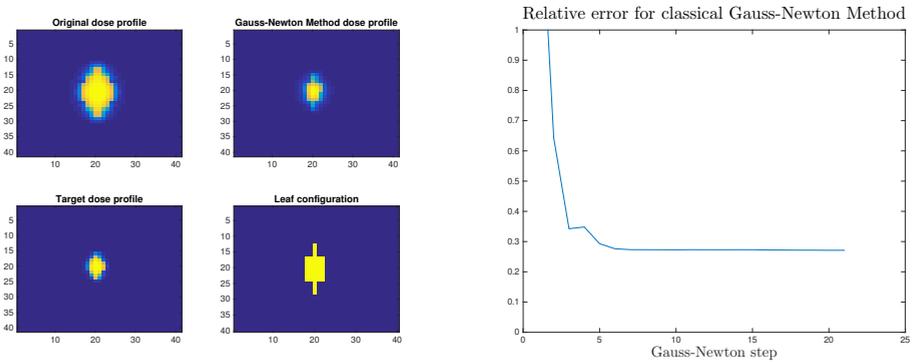


Figure 8: Gauss-Newton algorithm for the shrinking case of the tumor. The first plot shows the original dose profile, the target profile after the shift, the profile after the last Gauss-Newton step and the configuration of the leaves. The second plot presents the error evolution throughout the iteration steps.

4.5 Partical Swarm: vertical shift

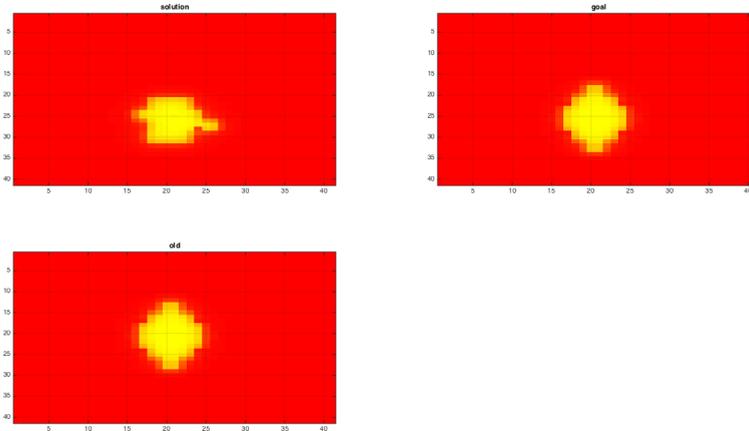


Figure 9: PSO applied to downward shift. Shown is dose profile before optimization (old), after optimization (solution) and shifted dose (goal).

Figure 9 shows that the PSO algorithm works quite well for this shift. However, the solution seems to be less accurate than the one obtained by the Gauss-Newton method (Figure 5). The PSO algorithm with these settings also took approximately half an hour to run, much longer than Gauss-Newton. This long duration is caused by the large number of expensive function evaluations, namely for each individual and each iteration. However, it might be profitable to use such black box algorithm, because one does not need to compute more derivatives when the number of parameters increases, but then again, one might need more individuals in PSO. Fortunately, PSO, and other evolutionary computations, can be sped up by employing parallel computing. However, this was outside the scope of the study week.

5 Discussion, conclusions and recommendations

We formulated the problem posed to us by the NKI as an optimization problem. We considered a simplified model: a two-dimensional rectangular patient and tumor and used the pencil beam approach to calculate the dose distribution. Obviously, this is unrealistic but it is a good starting point from a mathematical

point of view. Instead of considering all possible machine settings we only focused on the settings of the leaves. In the model, the deformations that were considered were simple so that the optimal settings were known. We tested several algorithms to see how good they were at finding these optimal settings.

We considered three numerical optimization methods. First we examined the Gauss-Newton method which we applied to four different scenarios: vertical, lateral, diagonal shifts, and tumor shrinkage. As discussed in Section 4, the Gauss-Newton method already appears to work rather well for all types of deformations that were considered. It may nevertheless be advisable to switch to a similar, but more robust, algorithm such as Levenberg-Marquardt. The position of closed leaf pairs is a potential problem for such algorithms.

Another algorithm that we discussed were evolutionary-type algorithms, in particular the PSO. We applied the PSO to the scenario of a vertical shift of the tumor. For our particular test case, the solution obtained through the PSO was less accurate than that of the Gauss-Newton method (compare Figure 5 to Figure 9). It also took a much longer running time, which is not feasible for real life situations. On the other hand, the PSO has the advantage that it can escape from local minima, unlike to the Gauss-Newton method. Faster and eventually more accurate results with the PSO algorithm might be obtained if parallel computing is used (this was outside the scope of the study week).

Finally, we also proposed a method, which we call a gradual deformation method. This method might be much easier and faster than the PSO. An additional advantage of this method is that it might avoid drifts to inappropriate solutions (local minima). Testing this approach was unfortunately outside the scope of the study week.

One of the ideas that came up during the study week was on adding a penalty term to the minimization problem. The rationale behind this is that deformations will typically not be very large and therefore the optimal settings should be somewhat close to original settings as well. More specifically, one could try to add a penalty term $\gamma \|\phi - \phi_0\|^2$ to the objective function, changing the optimization problem (1) to

$$\arg \min_{\phi} \|D(\phi, P_1) - D_{\delta}\|^2 + \gamma \|\phi - \phi_0\|^2.$$

for some number $\gamma > 0$ (where one should normalize both terms so that they become unitless). Doing this will ensure that ϕ does not stray too far from the

original configuration ϕ_0 during the optimization process. Unfortunately there was no time during the study week to work this out further. We considered the minimization problem with $\gamma = 0$, i.e. no penalty term was included in our results. This penalty term deserves further investigation.

Further investigations can be directed at the improvements of our approach by considering a more realistic model than the toy model discussed in Section 3.4. This model should take into account all settings of the machine (such as the larger number of leaves, the rotation steps and the time spent at each angle) and a description of the patient which is more faithful to reality.

Acknowledgements

We thank Simon van Kranen for helpful comments on the manuscript.

References

- [1] A. Ahnesjö. Collapsed cone convolution of radiant energy for photon dose calculation in heterogeneous media. *Med. Phys.*, 16(4):577–592, 1989.
- [2] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996.
- [3] Å. Björck. *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. ISBN 0-89871-360-9.
- [4] R. Fletcher. *Practical methods of optimization. Vol. 1*. John Wiley & Sons, Ltd., Chichester, 1980. ISBN 0-471-27711-8. Unconstrained optimization, A Wiley-Interscience Publication.
- [5] e. a. Siesling, S.S. *Kankerzorg in beeld*. Integraal kankercentrum Nederland, 2014. ISBN 978-90-72175-00-7.
- [6] Y. Watanabee. Point dose calculations using an analytical pencil beam kernel for IMRT plan checking. *Phys. Med. Biol.*, 46:1031–1038, 2001.

Power line route optimisation in a finite spatial grid

Nieke Aerts (TU Berlin), Emile Broeders (Utrecht University), Erik Bruin (Utrecht University), Ross J. Kang (Radboud University Nijmegen), Pedro Munari (Federal University of Sao Carlos)

1 Introduction

Power lines are integral to one of our most important infrastructure systems: the power supply network. They are run all over the world, across all kinds of landscape. While they are vital to our day-to-day life, they also have some negative influence on the environment, the view, and on other man-made structures. Building cost-effective power lines with limited environmental and social impact is a critical task. This paper considers a problem introduced by the British company NM Group, which specialises in surveying and planning for power line infrastructure. The main question which NM Group posed and which we address in our report is the following.

“How can we find a power line route of minimum cost?”

The cost of building a power line route is determined by many factors, not all of which are purely material. In order to specify these costs, NM Group has identified a few main factors, all of which are later incorporated into the mathematical model we study.

1. Occupation of an area by power line structures.

- (a) SVETIG.

We assume that for each specific area of land the negative influence (cost) of existing infrastructure and landscape can be categorised. The categories, the so-called SVETIG factors, are Socioeconomic/Political, Vegetation, protected Environment, Terrain, technical Infrastructure, and Geotechnical. The way that this is

quantified is out of scope for this report, and for modeling purposes we may assume these are arbitrary positive weights in the plane, over which we integrate.

(b) Structure type.

Aside from the attributes of the area itself, the type of structure also may be taken into account, depending on the granularity. In particular, the area can be passed over with heavy electric cables suspended in the air or it can be occupied by the base of a power line support tower. Clearly the presence of a support tower is more disruptive than that of a suspended cable.

2. Material construction costs.

The more lengthy the power line is, the more costly the materials used for that line, in linear proportion. For costs associated with the construction and placement of towers, two further (multiplicative) penalty factors come into play.

(a) The distance between two consecutive towers. (“Stretch penalty”)

When building a power line, the cables go straight from the top of one tower to the top of the next, i.e. along a straight line segment in the plane. In the third dimension, however, gravity takes its toll in terms of both the weight and sag of the suspended wire between those towers. Therefore, the further apart two consecutive towers are placed, the greater must be the load-bearing capacity and height of the two towers. Extremely long distance between consecutive towers is highly costly or impossible. Rarely is a very long span required, when there are no other options, for instance if the power line must cross a river or canyon.

(b) The angle between the incoming and outgoing lines of a tower. (“Angle penalty”)

When the power line segments supported by one tower form a sharper angle (with respect to the plane), the weight of the cables exerts a force on the tower roughly in the direction of the angle’s bisector. To counteract this force, some counterbalance or reinforcement of the tower is necessary. For instance when the angle is roughly 180° , only a simple tower is needed, whereas if the angle is off by more than 10° , then a stronger type of tower must be used.

In fact, the penalty factor is more akin to a step function, due to the need for increasingly robust types of tower.

1.1 Discretisation and macro/micro separation

For practical reasons, the problem as stated above must be discretised — that is, the spatial region under consideration is dissected using a regular grid, each cell of which is assigned a weight according to an amortised SVETIG factor. Moreover, NM Group deemed it necessary to make a separation into two levels of discretisation, the second of which may be interpreted to be a refinement of the first. Naturally, we refer to the first as the macro scale problem and the second as the micro scale problem. Later, the reader doubtless will notice that the macro and micro level problems are qualitatively distinct, and as such there could be justifiable objection about whether the combination of solutions to these two sub-problems constitutes an overall solution to the original problem as stated above. However, in our work we have taken this separation heuristic as given, partly because it is justified by the limitations of data-acquisition resources, and partly because we also propose heuristic and/or approximative approaches due to computational difficulties inherent to the global problem. It is worth noting that in our report we have independently chosen the type of tessellation — be it hexagonal, triangular, square — out of convenience, though with suitable routine modifications our methods apply to any regular grid pattern, at either scale. In summary, we split the problem as follows:

1. Macro scale.

At this level, NM Group has indicated that the overall area of consideration is typically a region of about 100 by 100 kilometres. This area is divided into cells of approximately 2 kilometres in diameter. The desired output of this sub-problem is a “rough” routing consisting of a connected sequence of macro cells, which we refer to as a corridor.

2. Micro scale.

The small subset of the cells identified at the macro scale are examined more closely for the micro scale problem. The selected macro cells are divided according to a finer grid consisting of micro cells at most 50 metres in diameter. The desired output of this sub-problem is the overall desired minimum cost power line route.

A major reason for the two-layer separation of the problem is based on costs incurred by NM Group for determining the SVETIG weighting factors. The SVETIG factors applied to the macro cells are inferred from cheap or free satellite-based imagery and mapping data. By contrast, it is assumed that, after the specific desired macro cells are identified, special surveillance missions are carried out over those regions, with the aid of lasers mounted on helicopters and similar (in particular, rather costly) data-acquisition methods, in order to determine the more detailed SVETIG factors used at micro scale.

Since the micro level data is gathered and formulated by NM Group itself, there is some flexibility in the defined size of micro scale cells. Naturally, the quality of the solution may improve by using a finer tessellation on input; however, much more computation time might be required to obtain the output.

1.2 Routing costs at different scales

As alluded to above, we permit significant differences in treatment of the problem when considered at different scales.

From the macro viewpoint, one may interpret it that we are unable to “see” precise details of the routing: we need not account for the most accurate length of the route, the specific number of towers placed, nor the corresponding angles. Instead, we only roughly account for the presence and SVETIG impact of power line routing through a given macro cell. As we will see later, this macro problem reduces to a relatively simple minimum cost path problem, solved efficiently using Dijkstra’s algorithm.

For the micro viewpoint, we narrow in on the output of the macro scale solution, i.e. we restrict attention to the subset of macro cells identified in the first step and divide them into smaller cells. Here, more intricate aspects of the routing problem come into effect. Specifically, for a placement of the towers at the centres of some micro cells, we calculate the cost of the corresponding power line routing by incorporating all the detailed cost contributions. That means we consider not only the (detailed) SVETIG factors, but also structure type, inter-tower distances, and angles at support towers.

Although essentially any reasonable mathematical combination of the cost factors described above can be handled by our method, we assume for concrete-

ness and simplicity that the following cost functions apply for routing through a given micro cell:

- The cost of placing a tower is
 - the material unit cost of a support tower
 - multiplied by the SVETIG factor for placing a tower in that cell
 - multiplied by the stretch penalty
 - multiplied by the angle penalty.
- The cost of suspending power line is
 - the material cost per unit length of power line
 - multiplied by the SVETIG factor for suspending power line in that cell
 - multiplied by the length of power line intersecting that cell.

We require no special assumptions about the unit costs, SVETIG factors, or penalty functions, except merely that they are fixed or monotone (decreasing with respect to angle or increasing with respect to stretch length) and that they are provided to us beforehand by NM Group.

1.3 Avoidance of “rubberbanding”

As a side remark, it is worth noting that NM Group originally gave a slightly different formulation of the power line routing problem. They had suggested to split the problem into three sub-problems, the macro and micro layers followed by an ad hoc “rubberbanding” protocol. Essentially, this delays all angle penalty considerations until the end. In other words, they suggested first to find a micro solution excluding angle considerations, and then afterwards to run a local perturbation procedure to fix and compensate for any undesirable angle penalties.

In our work, we have circumvented the need for (or at least limited the benefit of) this local adjustment procedure by directly incorporating angle penalties

into the mathematical model. Of course, because of discretisation, there is still the possibility of cost saving by “rubberbanding” upon the output of our suggested algorithms, but typically this saving will be of less interest as it will be much smaller than it was for their original approach. In order to eliminate as much as possible the potential “rubberbanding” savings, the discretisation could be made as fine as possible, but this would at the same time increase computational requirements. We in fact recommend NM Group survey according to micro cells that are the size of a tower’s base (rather than 50 metres in diameter).

1.4 Outline and overview

In this report, we are mainly concerned with algorithmic solution strategies for the above discretised problem. That is, subject to pre-determined unit costs, SVETIG factors, and penalty functions, we demonstrate how to effectively compute the desired optimal power line route, by solving first the macro and then the micro level sub-problem. We first show that the macro level problem can be solved both exactly and efficiently by a relatively straightforward application of Dijkstra’s algorithm; we describe this solution in Section 2. We then show that the micro level problem can be solved both exactly and efficiently using a more involved modification and application of Dijkstra’s algorithm; we describe this solution in Section 3. The efficiency for the exact micro level solution is of a theoretical nature (i.e. the computational problem is solvable in polynomial time), and unfortunately our algorithm in unvarnished form is unlikely to produce timely solutions for the problems typically encountered by NM Group in practice. We therefore found it appropriate to propose various potential practical approaches to solving the micro level problem which we describe in Section 4.

2 Macro step efficient exact solution

In the macro step a corridor is to be found. Within this corridor lies the minimum cost path that connects the start and end points of the power line route. Afterwards, more detailed information about the corridor is gathered. Our algorithm will find the minimum cost corridor.

First an auxiliary weighted graph is built, then a minimum cost path in this graph is found by applying for example Dijkstra's shortest path algorithm [2]. We describe how to construct the weighted graph such that the shortest path output is a minimum cost corridor. Moreover, the output corridor minimises the Euclidean length of a polygonal curve between its end cells, optimised over all such polygonal curves whose corners lie only at the centres of cells.

The previous algorithm used by NM Group was based on finding the minimum cost path in a graph with weight on each of the vertices. In the hexagonal grid this graph is obtained by letting every hexagon be a vertex which is connected to all its neighbouring hexagons (see Figure 1). A problem that arises is that the two paths connecting the top-left and bottom-right hexagons in Figure 2 both go through the same number of hexagons. If every hexagon has the same cost, the algorithm of Dijkstra might well select the top-right path. When the actual power line is built it will not necessarily pass through the centers of the hexagons. Therefore the other path turns out to be much cheaper, as the eventual total length is shorter. Hence, our first goal is to devise an algorithm that selects a path of minimum cost and takes into account some measure of Euclidean length of the path.

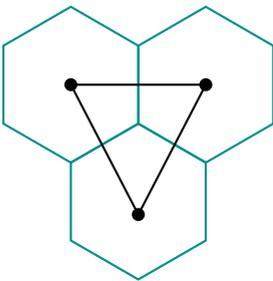


Figure 1: Each hexagon gets one vertex inside and every vertex is connected with the six vertices in the six neighbouring hexagons.

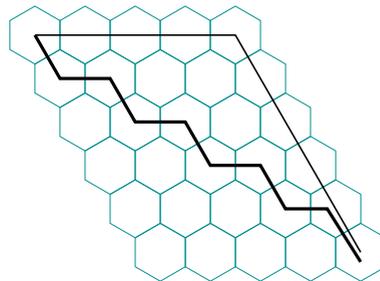


Figure 2: The two paths connecting the top-left and bottom-right hexagons go through the same number of hexagons.

2.1 Hexagonal grid

Instead of changing the algorithm, a different graph is constructed. The new graph also encodes the Euclidean distance between the two neighbours in a path as a cost. We will first describe how the graph is obtained and then show why a minimum cost path algorithm will output the desired path.

Each hexagon gets six vertices inside which are cyclically connected forming a new hexagon. Let $v(h, e)$ be the vertex in the hexagon h closest to border-edge e . Let h^* be the neighbouring hexagon on the edge e , see Figure ?? where e is the dashed edge between the hexagons h and h^* .

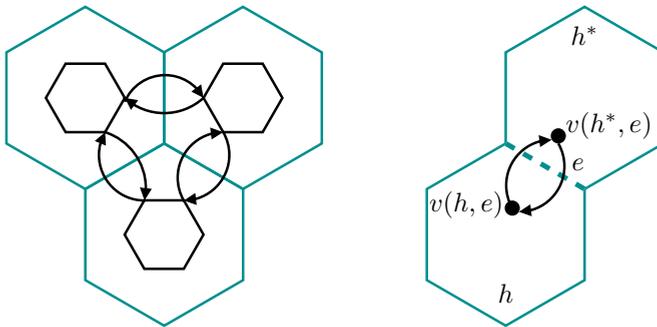


Figure 3: Left: Each hexagon (in cyan) gets a hexagon inside, where each corner is a vertex. Each vertex is connected with an undirected edge to its neighbours in the hexagon and with a directed edge to and from the closest vertex in the neighbouring hexagon. Right: the vertices inside a hexagon are labeled $v(h, e)$ for begin in hexagon h closest to border e of the hexagon.

There is a directed edge from $v(h, e) \rightarrow v(h^*, e)$ having weight equal to the cost of the hexagon h^* , i.e., the weight of the hexagon it enters. There is also an edge from $v(h^*, e) \rightarrow v(h, e)$ having weight equal to the cost of the hexagon h , again the weight of the hexagon it enters. Each interior edge in the hexagon has very small weight. Let c_{min} be the cost of the hexagon with minimal cost, and n the number of hexagons. Then every edge interior to a hexagon is assigned a weight $c_{min}/(3n)$.

A minimum cost path in this graph is a minimum cost path on the map. Moreover, for each inner polygon in this path, the distance between the centers of

its two neighbours is taken into account as a measure of the length of the path. In Figure 4 two examples are given that illustrate how the distance measure works. Let x_h be the distance between the center points of the neighbours of a hexagon h , that is interior to a path. The sum of x_h over all inner hexagons h of the path is less for the red paths than for the black paths. The red paths are also of lower cost than the black paths since there are less interior edges used.

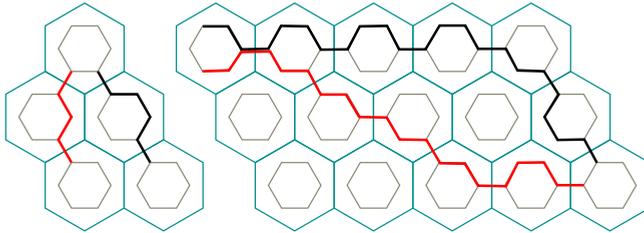


Figure 4: The path on the left (in red) is of lower cost, as it uses fewer interior edges.

2.2 Square grid

In a similar way as for the hexagonal grid, an auxiliary graph can be constructed for the square grid. A minimum cost path in this graph relates to a minimum cost routing in the square grid. Each square gets four vertices inside, cyclically connected. Each vertex has an outgoing edge to the vertex in the next square. This edge gets the weight of entering the neighbouring square. Let c_{min} be the cost of the square with minimal cost, and n the number of squares. Then every edge interior to a hexagon is assigned a weight $c_{min}/(2n)$.

Finding a minimum cost path in this graph gives a minimum cost path on the map such that expensive squares are avoided and the path takes into account some measure of Euclidean length of the path.

3 Micro step exact solution

The corridor that is obtained in the macro-step is now filled in with more detail. We assume that the grid on top of the map is fine, that is, the area covered by one polygon (triangle, square or hexagon) is close to the area needed to build a post. This ensures that we do not have to take into account where in a polygon the post is placed, each post will be placed in the centre of a polygon. For each polygon we now consider the more specific costs as defined in Section 1.2.

In Section 3.1 we will describe how to build an auxiliary graph, with costs on the edges, such that a minimum cost path in this graph gives a minimum cost solution to the micro problem. Dijkstra's algorithm can be used to find such a minimum cost path in the auxiliary graph. In Section 3.2 we will show that there is a bijection between the paths in the auxiliary graph and the routings in the micro corridor. Moreover we show that the minimum cost paths in the auxiliary graph are in bijection to the minimum cost routings in the micro corridor. In Section 3.3 we describe how to calculate the costs on the edges of the auxiliary graph in the case of a square grid. We conclude by discussing the complexity of Dijkstra's algorithm on the auxiliary graph in Section 3.4.

3.1 Preprocessing for Dijkstra's algorithm

We assume that the maximal distance between two towers is bounded by some value K . This is a valid assumption, as there is a value for which the cost that comes from the distance between two consecutive posts becomes too high to be feasible or reasonable. Recall that this cost is denoted by stretch penalty. We will construct a directed graph such that a minimum cost path in this graph describes a minimum cost route in the original grid.

Suppose that from a polygon we can reach k other polygons with one stretch, i.e., the two polygons are consecutive posts in a path. Note that this includes steps of all lengths up to the maximum length K (see Figure 5). Recall that the posts are placed in the centre of a polygon and the maximum distance between two posts is thus K . Therefore, the set of reachable polygons from a polygon p is the set of polygons with centre point at distance at most K from the centre point of p .

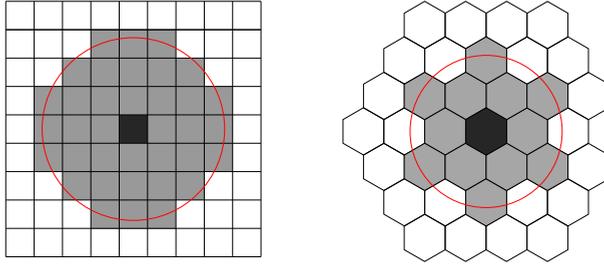


Figure 5: The dark square (hexagon) represents the current location, the lighter areas represent the possible landing spots coming from the dark area.

Let A be the starting point, B be the target point and let p be any polygon. For each polygon p let n_1, \dots, n_k be the reachable polygons. We introduce k dummy vertices for p . Each dummy vertex is labeled (p, n_l) for $l = 1, \dots, k$. A dummy vertex (p, n_l) represents that *the current location is p* and this location is reached by *coming from n_l* . For A there is only one vertex introduced, denoted by (a, \emptyset) , which is the starting point of the path. For B there is a vertex (b, j) for all polygons j from which B can be reached and a vertex (\emptyset, b) which is the end point of the path.

The vertex (a, \emptyset) has only outgoing edges: $(a, \emptyset) \rightarrow (m, a)$ for every reachable polygon $m = 1, \dots, k$ of A . The vertex (\emptyset, b) has only incoming edges: $(m, b) \rightarrow (\emptyset, b)$ for every reachable polygon $m = 1, \dots, k$ of B . For every other vertex (p, n_l) there is a directed edge to (n_m, p) for each reachable polygon $m = 1, \dots, k$. A small example is given in Figure 6. The weight on such an edge between (p, i) and (j, p) represents the costs made for placing a post in p .

The costs involved in placing a post in p between i and j are:

- the material unit cost of a support tower;
- multiplied by the cost of the placement at p (the SVETIG factor);
- multiplied by the cost of the angle structure needed at p ;
- multiplied by the stretch penalty, which depends on the maximum of the two distances between p and i and between p and j .

There are also costs of suspending the power line. On an edge $(p, i) \rightarrow (j, p)$

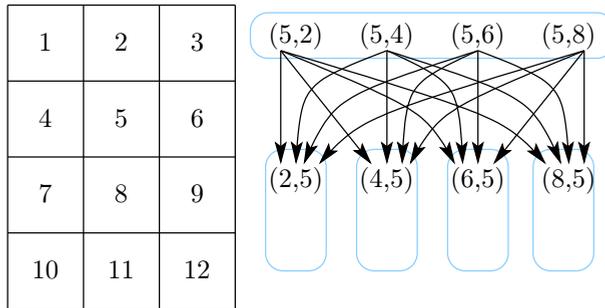


Figure 6: Part of the graph that is built: the square labeled 5 can stretch to any of its direct neighbours 2, 4, 6 and 8, therefore the vertices $(5, 2)$, $(5, 4)$, $(5, 6)$ and $(5, 8)$ are introduced. From each of these there is a directed edge to all the reachable neighbours: $(2, 5)$, $(4, 5)$, $(6, 5)$ and $(8, 5)$.

we incorporate the suspending costs between p and j . The suspending costs between i and p will be taken into account in the edge that ends in (p, i) .

Recall that the suspending costs per polygon are:

- the material cost per unit length of power line;
- multiplied by the SVETIG factor for suspending the power line in this polygon;
- multiplied by the length of the power line intersecting this polygon.

All the relevant information for the placing costs in p and the suspending costs between p and j is known at the time of selecting the edge $(p, i) \rightarrow (j, p)$. Therefore, all the costs can be incorporated on this edge.

The outgoing edges of (a, \emptyset) are charged in the same way, the placing of the post in A and the suspending costs between A and the target. The incoming edges of (\emptyset, b) are only charged with the placing cost of the tower in B . In Section 3.3 we will describe in detail how to calculate the costs on an edge in the case of a square grid.

The directed graph has at most k dummy vertices for each polygon in the grid. The polygons on the boundary will give rise to strictly less than k dummy vertices. Suppose the grid has m polygons. Then the graph will have approx-

imately $k \cdot m$ vertices. Each vertex has at most k outgoing edges. Therefore the resulting graph will have approximately $k^2 \cdot m$ edges.

On the resulting directed graph, Dijkstra's algorithm or any other shortest path solver can be applied to obtain the minimum cost route in the original grid. We will discuss the running time of this algorithm later in Section 3.4. First we will show that there is a bijection between a minimum cost path in the directed graph and a minimum cost routing in the grid.

3.2 Validation

Let A and B be the points that need to be connected by a power line. Let \mathcal{P} be the collection of paths that connect A and B , such that the distance between two neighbouring posts is at most K . Let k be the maximum number of polygons at distance at most K . Let D be the directed graph obtained as described in the previous section, where (a, \emptyset) and (\emptyset, b) represent A and B and D is such that at most k polygons are at distance at most K . Let \mathcal{Q} the set of directed paths from A to B in D .

Claim. There is a bijection ϕ between \mathcal{P} and \mathcal{Q} such that for every pair $p \in \mathcal{P}$ and $\phi(p) = q \in \mathcal{Q}$ the costs of p and q are equal.

Proof. Let $p \in \mathcal{P}$ be a path from A to B using the polygons $A = i_0, i_1, \dots, i_m, i_{m+1} = B$. We show how to obtain $q = \phi(p)$. Start with $q = ((a, \emptyset), (i_1, a))$. For every step between two polygons i_j and i_{j+1} in p , the distance is at most K . Therefore, the directed edge $(i_j, i_{j-1}) \rightarrow (i_{j+1}, i_j)$ must exist in D and we can add (i_{j+1}, i_j) to q . Last, we add (\emptyset, b) to q and we have $q \in \mathcal{Q}$.

On the other hand, every path $q \in \mathcal{Q}$ is mapped to a path $\phi^{-1}(q) = p \in \mathcal{P}$. Let q be the path:

$$(a, \emptyset) \rightarrow (i_1, a) \rightarrow \dots \rightarrow (i_j, i_{j-1}) \rightarrow (i_{j+1}, i_j) \rightarrow \dots \rightarrow (b, i_m) \rightarrow (\emptyset, b).$$

The path p can be obtained from q by selecting the path in \mathcal{P} that consecutively visits the polygons A, i_1, \dots, i_m, B . This path must exist in \mathcal{P} since all steps in q imply that the consecutive posts are at most at distance K from each other.

Let $p \in \mathcal{P}$ and $\phi(p) = q \in \mathcal{Q}$. The costs of the polygons used (with placing a post or by suspending over it) in p are the same as in q . The distances between

two subsequent posts and the angles at the posts are also the same in p as in q . Therefore, the cost of p must be equal to the cost of q . \square

It follows that if p is a minimum cost path in \mathcal{P} then $\phi(p) = q$ must be a minimum cost path in \mathcal{Q} . Therefore, the solution that a minimum cost path algorithm will give when applied to the directed graph, is a minimum cost path within the macro corridor for the original problem.

3.3 Calculation of costs

In this section we will describe how to calculate the cost of an edge in the directed graph. We will consider the case of a square grid, the calculations can be done for a hexagonal grid as well, if necessary. In the square grid we let the topleft corner be the origin. We start by introducing some notations.

- p, q, r, \dots generally denote points in the Euclidean plane, which are the center point of the squares s_p, s_q, s_r, \dots respectively.
- i, j generally denote the horizontal respectively vertical distance from the origin, and therefore also indicate a column (i) and a row (j).
- $c_S(p, q)$ is the cost function of the wires going over the area between p and q , say the suspension costs.
- $c_P(p, q, r)$ is the cost of placing a post in q coming from p and going to r , which is the landing cost of the tower for the given angle.

We consider the edge $(s_q, s_p) \rightarrow (s_r, s_q)$, this denotes that we have subsequent post in p then q and then r . The weight of this edge is:

$$c_S(q, r) + c_P(p, q, r) .$$

When $(s_q, s_p) \rightarrow (s_r, s_q)$ is chosen, the costs of everything that has to do with the post in s_q is charged as well as the cost of the wiring on top of the squares between q and r .

For the incoming edges to (\emptyset, s_B) , there is only the costs of placing a post in B :

$$(s_B, s_q) \rightarrow (\emptyset, s_B) \quad \text{has cost} \quad c_P(q, B, \emptyset) .$$

3.3.1 Calculation of suspension costs

The function c_S depends on the material cost, the SVETIG factors of the polygons over which the suspending takes place and the length of the power line over such a polygon. To capture this in the definition of c_S we introduce some notation.

- l_{pq} is the line on which p and q lie.
- $\text{cros}_V(pq, x)$ is the crossing of the line l_{pq} with the vertical at coordinate x , $\text{cros}_H(pq, y)$ is the crossing of the line l_{pq} with the horizontal at coordinate y .
- S_{ij}, S_p represent the SVETIG factor of the squares s_{ij} and s_p respectively.
- $\text{len}(pq)$ is the length of the line-segment between p and q and $\text{len}(s_{ij}, pq)$ is the length of the line l_{pq} in the square s_{ij} .
- W represents the cost of the wire per length factor.

The value $c_S(p, q)$ is calculated by summing over all columns in which the line segment appears (i), then over the rows in which the line segment appears in the i -th column (j). The sum then consists of the length of the line segment within square s_{ij} ($\text{len}(s_{ij}, qr)$) times the weight of the square s_{ij} . If necessary, the additional costs of the wires of length $\text{len}(p, q)$ can be added. Formally, we have the following expression:

$$c_S(p, q) = \text{len}(pq) \cdot W + \sum_{i=\lfloor p_x \rfloor}^{\lfloor q_x \rfloor} \sum_{j=\text{cros}_V(pq, i)}^{\text{cros}_V(pq, i+1)} S_{ij} \cdot \text{len}(s_{ij}, pq).$$

We will now explain how to calculate the crossing points and the length of the line-segment in a particular square. Any point on the line that goes through p and q is described with:

$$l_{pq}(t) = (q - p)t + p.$$

The line segment between p and q is given by varying t from 0 to 1. To obtain a value for $\text{cros}_V(pq, i)$, we use the time t^* such that the x -coordinate of $l_{pq}(t)$

is i . Here $\hat{\mathbf{e}}_y$ denotes the vertical unit vector.

$$\text{cross}_V(pq, i) = \begin{cases} p_y & \text{if } t^* \leq 0 \\ q_y & \text{if } t^* \geq 0 \\ l_{pq}(t^*) \cdot \hat{\mathbf{e}}_y & \text{otherwise} \end{cases}$$

For every square s_{ij} the point of entry (e_n) and the point of exit (e_x) of the line l_{pq} are computed. The length of the line-segment in the square is then equal to the Euclidean distance between e_n and e_x , that is,

$$\text{len}(s_{ij}, pq) = \|e_n - e_x\|_2.$$

3.3.2 Calculation of tower costs

The function c_P depends on the material cost of a support tower, the SVETIG factors of the polygon in which the tower is built, the angle structure that is needed for this tower and the stretch penalty. To capture this in the definition of c_P we introduce some notation.

- $P(q)$ represents the cost of placing a post in s_q .
- $L(p, q, r)$ represents the stretch penalty, that is, the extra costs at q induced by the maximum distance between q and its two neighbours.
- $A(p, q, r)$ represents the angle penalty that depends on the angle at q .

Using this notation, we calculate the cost of placing a post in q coming from p and going to r as:

$$c_P(p, q, r) = P(q) \cdot L(p, q, r) \cdot A(p, q, r).$$

Both L and A are step functions.

The function L depends on the maximum of $\text{len}(pq)$ and $\text{len}(qr)$. There is a table that contains the value of $L(p, q, r)$ given $\max(\text{len}(pq), \text{len}(qr)) < \ell$ for different values of ℓ .

The angle at q can be computed using the coordinates of p , q and r . Consider the two vectors $\vec{a} = (p_x - q_x, p_y - q_y)$ and $\vec{b} = (r_x - q_x, r_y - q_y)$. The angle

at q is the same as the angle between the two vectors. The angle between two vectors, denoted by θ , can be computed using the default dot product

$$\vec{a} \cdot \vec{b} = \|\vec{a}\|_2 \|\vec{b}\|_2 \cos(\theta)$$

where $\vec{a} \cdot \vec{b}$ represents the dot product between two vectors. Let p_x, p_y (respectively q_x, q_y and r_x, r_y) represent the coordinates of p (respectively q and r) then the angle at q is given by:

$$\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|_2 \|\vec{b}\|_2} = \frac{(p_x - q_x)(r_x - q_x) + (p_y - q_y)(r_y - q_y)}{\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \sqrt{(r_x - q_x)^2 + (r_y - q_y)^2}}.$$

The function A depends on the angle at q . There is a table that contains the value of $A(p, q, r)$ given that the angle at q is at most α for different values of α .

3.4 Complexity of solving shortest path problems

In the previous section we have described how the problem can be formulated as a minimum cost path problem in a directed graph. This is equivalent to a weighted shortest path problem in a directed graph. There are several algorithms known that solve this problem in polynomial time. An example is the famous algorithm of Dijkstra [2]. Dijkstra's original algorithm has a running time of $\mathcal{O}(|V|^2)$ where $|V|$ denotes the number of vertices in the graph.

The implementation due to Fredman and Tarjan [3] is asymptotically the fastest known shortest-path algorithm for directed graphs with non-negative weights. This algorithm has time complexity $\mathcal{O}(|E| + |V| \log(|V|))$ and space complexity $\mathcal{O}(|V|^2)$, where $|E|$ is the number of edges. Consider the graph we have built in the previous sections. The graph has $k^2 m$ edges and km vertices, where k is the number of reachable neighbours and m is the number of polygons in the micro corridor. This relates to a running time of $\mathcal{O}(k^2 m + km \log(km))$ and space requirement of $\mathcal{O}(k^2 m^2)$.

Let us briefly consider the rough boundaries of a problem that could arise in practice (as we were informed by NM Group). Suppose that the grid consists of squares and each square covers 20×20 square metres. The distance between A and B is approximately 50 kilometres and the micro corridor is approximately

50 kilometres long and 4 kilometres wide. This yields $m \approx 2500 \times 200 = 500,000$. Suppose the distance between two posts is at most 2 kilometres. Recall that k is the number of squares whose centre point lies within distance $K = 2$ kilometres, from the centre point of some chosen square. For a square grid k can be found using the solution to the so-called Gauss circle problem: “How many lattice points are there in a circle with radius r that is centred at the origin”. For bounded radius this number can be found using the following formula [7]:

$$N(r) = 1 + \sum_{i=0}^{\infty} \left(\left\lfloor \frac{r^2}{4i+1} \right\rfloor - \left\lfloor \frac{r^2}{4i+3} \right\rfloor \right).$$

In our case the lattice points are the centres of the squares, therefore one unit relates to 20 metres. The radius in units is $2000/20 = 100$ and $N(100) = 31,417$. In this example the directed graph will have approximately $500,000 \times 31,417 \approx 15.8$ billion vertices and even more edges.

Although we have shown that the micro level problem can be solved efficiently in theory, the above rough calculation suggests that in practice it may be prohibitively expensive, both in terms of computational time and memory storage requirements, to obtain an exact solution. In the next section, we will describe some heuristic algorithmic strategies which could be efficient in practice.

4 Heuristics and metaheuristics

In this section, we propose a set of simple heuristics that have the purpose of quickly providing solutions to the micro level problem. First, we present the main ideas and the algorithmic description of constructive heuristics that try to build initial paths. Then, we propose several improvement heuristics based on quick perturbations of paths. Finally, we describe ways of combining the proposed heuristics by using a metaheuristic, namely the Tabu Search strategy.

To describe the heuristics, we use the following nomenclature and notation. Let N be the number of polygons used in the micro step, so that $H = \{h_1, h_2, \dots, h_N\}$ is the set of polygons. We assign each polygon to a unique pair (i, j) that corresponds to its grid position. This way we may conveniently refer

to a polygon $h_k \in H$ as $h_{(i,j)}$ by using this map. We assume the grid is given by n_R rows and n_C columns, such that $n_R \times n_C = N$. We denote by h_S and h_E the starting and ending polygons of the problem, respectively. In addition, (i_S, j_S) and (i_E, j_E) are the grid positions of h_S and h_E , respectively.

4.1 Constructive heuristics

The first heuristic we propose is a greedy algorithm that starts from h_S and iteratively selects other polygons as landing points until h_E is reached. This constructive heuristic, called CH1, is detailed in Algorithm 1. Note that we use the function $movecost_{ij}(p, q)$ to compute the total cost of jumping from any polygon $h_{(i,j)}$ to another reachable polygon $h_{(p,q)}$. Such cost can be implemented as described in Section 3.3. This heuristic goes from one polygon to another only if there are no obstacles between them. There is no guarantee of finding a feasible path at the end of the heuristic (we provide some ways of getting rid of this disadvantage ahead in this section).

A tentative way of improving heuristic CH1 would be to allow infeasible jumps, *i.e.* going from one polygon to another even though there are obstacles between them. To be consistent with the optimisation objective, such infeasible jumps should incur penalisation costs. With this in mind, we propose a second constructive heuristic, called CH2, which is similar to CH1 except for allowing infeasible paths. CH2 is presented in Algorithm 2. Note that we use the function $movecost_{infeas}_{ij}(p, q)$ to compute the cost of jumping from a polygon $h_{(i,j)}$ to a reachable polygon $h_{(p,q)}$. This function is similar to that used in Algorithm 1, but has to incorporate additional costs to penalise infeasible jumps.

The third constructive heuristic, CH3, works in a different way in relation to the two previous ones. It starts with the path $(i_S, j_S) \rightarrow (i_E, j_E)$ even though this may be infeasible (due to obstacles or a large number of jumps). Then, new polygons are inserted in the path iteratively, in order to improve the total cost of the path. The heuristic is defined in Algorithm 3.

In all the algorithms presented above, we can introduce randomness with the aim of improving their performance. This can be done in many different ways and we mention a few in the following, without going into details. One first idea is to insert random perturbation costs in the computation of the moving

```

PATH =  $\{(i_S, j_S)\}$ ;
 $(i, j) = (i_S, j_S)$ ;
end_reached = 0;
while  $i \leq n_R$  and  $j \leq n_C$  and end_reached = 0 do
    set  $h_{(i,j)}$  as the current landing point;
     $w = 3$ ;
    candidates = 0;
    while candidates = 0 and  $w \leq n_R$  do
        for  $r = 1$  to  $w$  do
            for  $s = -w$  to  $w$  do
                if  $i + r \leq n_R$  and  $1 \leq j + s \leq n_C$  and move is feasible then
                    if  $i + r = i_E$  and  $j + s = j_E$  then
                         $a_{rs} = -1$ ;
                        end_reached = 1;
                    else
                         $a_{rs} = \text{movecost}_{ij}(i + r, j + s)$ ;
                    end
                    candidates = 1;
                end
            end
        end
         $w = w + 1$ ;
    end
    if (candidates = 0) then
        STOP: no solution found;
    end
     $(\hat{r}, \hat{s}) = \text{argmin}_{r,s} \{a_{rs}\}$ ;
    update total costs;
     $i = i + \hat{r}$ ;
     $j = j + \hat{s}$ ;
    add  $(i, j)$  to PATH;
end
return PATH as solution;

```

Algorithm 1: Constructive heuristic 1 (CH1).

```

PATH =  $\{(i_S, j_S)\}$ ;
 $(i, j) = (i_S, j_S)$ ;
end_reached = 0;
w = 3;
while  $i \leq n_R$  and  $j \leq n_C$  and end_reached = 0 do
    set  $h_{(i,j)}$  as the current landing point;
    for r = 1 to w do
        for s = -w to w do
            if  $i + r \leq n_R$  and  $1 \leq j + s \leq n_C$  then
                if  $i + r = i_E$  and  $j + s = j_E$  then
                    |  $a_{rs} = -1$ ;
                else
                    |  $a_{rs} = \text{movecostinfeas}_{ij}(i + r, j + s)$ ;
                end
                candidates = 1;
            end
        end
    end
    end
     $(\hat{r}, \hat{s}) = \text{argmin}_{r,s} \{a_{rs}\}$ ;
    update total costs;
     $i = i + \hat{r}$ ;
     $j = j + \hat{s}$ ;
    add  $(i, j)$  to PATH;
end
return PATH as solution;

```

Algorithm 2: Constructive heuristic 2 (CH2).

```

PATH =  $\{(i_S, j_S), (i_E, j_E)\}$ ;
set total cost as movecostinfeasiSjS(iE, jE);
candidates = 1;
w = 3;
while i < nR and j < nC and candidates = 1 do
  candidates = 0;
  best_cost = ∞;
  foreach two neighbour pairs (i1, j1) and (i2, j2) in PATH do
    i = (i1 + i2)/2;
    j = (j1 + j2)/2;
    for r = -w to w do
      for s = -w to w do
        if  $1 \leq i + r \leq n_R$  and  $1 \leq j + s \leq n_C$  then
          if  $i + r \neq i_1$  or  $i + r \neq i_2$  or  $j + s \neq j_1$  or  $j + s \neq j_2$  then
            ars = insertioncostinfeas(i + r, j + s);
            candidates = 1;
          end
        end
      end
    end
    (r̂, ŝ) = argminr,s{ars};
    if (best_cost > ar̂ŝ) then
      î = i + r̂;
      ĵ = j + ŝ;
    end
  end
  if (candidates = 0) then
    | STOP: no solution found;
  end
  update total costs;
  add (î, ĵ) to PATH;
end
return PATH as solution;

```

Algorithm 3: Constructive heuristic 3 (CH3).

and insertion costs. This would force the algorithm to choose jumps that are not the best at the moment, but which may contribute to a better global solution (as the algorithm becomes less greedy). Another idea would be to restrict the jumps only to randomly chosen polygons in a small neighbourhood of the current one, instead of considering all the reachable polygons. When included in the previous algorithms, these ideas (and many others) have the potential to construct better routes, especially when used in a metaheuristic context.

4.2 Path perturbation heuristics

The heuristics proposed so far have the purpose of quickly providing an initial solution of the problem. On the other hand, the quality of the solution may be poor and in the worst case no path is obtained. Therefore, we propose some perturbation heuristics, also known as improvement heuristics, with the purpose of improving the paths obtained by the constructive heuristics. These are also simple and quick heuristics and they work by changing an existing path by means of adding, removing or replacing one or more polygons. Hence, we assume that an initial path of K polygons is available, which we denote by the ordered set $PATH = \{(i_1, j_1), (i_2, j_2), \dots, (i_K, j_K)\}$. The heuristics are described as follows:

1. *Add polygons to path.* For each $k = 1, \dots, K - 1$, select the pair of polygons $h_{(i_k, j_k)}$ and $h_{(i_{k+1}, j_{k+1})}$, with (i_k, j_k) and (i_{k+1}, j_{k+1}) in $PATH$. Then, compute the cost of inserting a reachable polygon $h_{(i, j)}$ between those two such that $(i, j) \notin PATH$. After computing all these insertion costs, carry out only one insertion, namely that one with the minimum insertion cost. This will increase the path by one polygon, which may reduce the total cost (e.g. if a penalisation for crossing an obstacle in the original path is removed after the insertion). Even when the total cost of the resulting path becomes worse, this path can be stored as an alternative solution with the purpose of being useful e.g. in a metaheuristic environment, as we describe later in this section.
2. *Remove polygon from path.* Compute the new total cost of removing one polygon at a time from the current path. Then, the resulting path is obtained by removing the polygon that corresponds to the best total cost among all the others. Again, even if the total cost of the resulting

path is worse than that of the original path, it can be useful to store the resulting path to be later used in a metaheuristic environment.

3. *Replace polygons in path.* For each $(i, j) \in PATH$, compute the change in the total cost given by replacing the corresponding polygon $h_{(i,j)}$ by one of its neighbours in the grid, namely $h_{(i+r,j+s)}$ for $-w \leq r \leq w$, $-w \leq s \leq w$ and $w > 0$, such that $1 \leq i+r \leq n_R$ and $1 \leq j+s \leq n_C$. Then, after computing all the changing costs, replace the polygons that lead to a new path with the best total cost among all the others.

We can introduce randomness to these heuristics in a similar way as previously proposed for the constructive heuristics. This can be useful to provide better solutions, especially if we are able to call the heuristics several times. Indeed, to get the best of these heuristics we can combine all them together and use a metaheuristic algorithm to coordinate their interaction, as described in the next section.

4.3 Metaheuristics

There are many metaheuristics proposed in the literature, such as Tabu Search, Genetic Algorithms, Ant Colony Optimisation and Simulated Annealing [4]. These methods have been widely used in the last decades to find feasible solutions for many hard combinatorial optimisation problems [1], especially when modeling practical situations. Although having different motivations, metaheuristics have a common basic idea: they start with one or more initial solutions and then use a set of simpler heuristics to iteratively perturb these solutions, with the aim of obtaining better quality solutions at the end. These perturbations are important not only to improve the quality of the initial solutions, but also to get rid of local optima. This way, a metaheuristic is able to exploit several neighbourhoods of the feasible set, which increases its chances of finding an optimal solution of the problem (or getting closer to it) in relation to a standard heuristic.

All the constructive and perturbation heuristics described earlier in this section can be used to build a metaheuristic method. The constructive heuristics proposed in Section 4.1 can provide initial solutions to the method, especially when randomness is incorporated, so that many different solutions may be provided. Then, at each iteration of the metaheuristic, the path perturba-

tion heuristics proposed in Section 4.2 may be used to try to get paths that are better than those found so far as well as to diversify the current search space.

To exemplify the use of a metaheuristic in this context, we describe a Tabu Search (TS) method [5] that relies on the previously defined heuristics. There are different variants of TS and furthermore the algorithm described below may be implemented in different ways. The most appropriate variant/implementation depends on the type of the problem, heuristics used to construct and perturb solutions, data structures, computer environment, and many other characteristics. This way, to obtain a TS implementation that works well in practice, it is important to test with different strategies and parameter choices first and then stick to the best setting.

The main idea of TS is to keep a *tabu list* to indicate which changes are prohibited when the perturbation heuristics are called. This tabu list is dynamically updated through iterations, so that some prohibited changes remain in the list for a given number of iterations. After a change is removed from the tabu list, it becomes allowed again. This is the way that TS algorithms try to get rid of local optima.

Let $\mathcal{S} = \{PATH_1, PATH_2, \dots, PATH_P\}$ be an ordered set of solutions, using the total cost of each path as its ranking value. We denote by TL the tabu list that contains the perturbations (moves) that must be prohibited in the current calls to the path perturbation heuristics. This can be implemented as a simple circular list so that old entries are replaced by new ones through the iterations (which makes this tabu list dynamic). Finally, we define an ordered set \mathcal{C} that contains the paths that result from the path perturbation heuristics. Using this notation, the TS method is presented in Algorithm 4. The stopping criterion of the algorithm is not particularly defined, as it depends on the implementation. The usual criteria are given by reaching a maximum running time or a maximum number of iterations of the algorithm.

There are several open steps in Algorithm 4. One of them is how to choose index i in line 5. For instance, by choosing $i = 0$ we take the path with the currently best total cost. This may be a good choice for getting better solutions in the improvement heuristics, but other choices may be useful to generate solutions that allow to exploit other neighbourhoods of the feasible set. Hence, an implementation of the algorithm should consider different ways

Initialise set \mathcal{S} by calling the constructive heuristics;
 $TL = \emptyset$;
while *stopping criterion is not reached* **do**
 $\mathcal{C} = \emptyset$;
 Choose an index i such that $PATH_i \in \mathcal{S}$;
 Apply the improvement heuristics to $PATH_i$ allowing only the
 perturbations that are not in TL ;
 Add to \mathcal{C} all the paths resulting from the perturbation heuristics;
 Select K paths from \mathcal{C} , add them to \mathcal{S} and add the perturbation moves
 that originated them to TL ;
end
if ($\mathcal{S} = \emptyset$) **then**
 STOP: no solution found;
end
Return $PATH_1$ as the best solution found for the problem;
Algorithm 4: Tabu search method (TS).

of carrying out this step. These comments apply to other parts of the algorithm as well, such as the size of K and how to select the paths from \mathcal{C} .

5 A probabilistic viewpoint for further study

A classic topic in probability theory is *first passage percolation*, cf. Hammersley and Welsh [6]. This topic is generally concerned with the following basic question. Given some growing sequence of graphs — think of regular structures such as the two-dimensional grid or the complete graph — whose vertices or edges are given a random distribution of weights, what is a good asymptotic description of the lowest weight path (geodesic) between two specified vertices?

As an offshoot of the SWI problem, it could be natural and novel to consider a variant of first passage percolation that incorporates some aspects of power line routing. Although interesting, this is beyond the scope of this project and report. We only sketch a simplistic model problem which might be considered along these lines. The following is a variant of first passage bond percolation. (Note that more complex variants of first passage site percolation might model

the original problem more faithfully.)

The underlying graph is an $n \times n$ grid, embedded in the plane, whose nodes are given some random weights, according to a given probability distribution. Here we define a *power line route* to be a sequence of mutually distinct points of the grid, every consecutive pair of which is within distance k , according to a given norm. The calculation of the cost of a power line route takes into account two components, weighted separately. First, it incorporates the length of the polygonal curve (which accounts for the cost of the power line). Second, it includes the cumulative cost of all the nodes in the sequence (which accounts for the cost of the tower construction). The cost of each node is calculated as a product of the weight at that node and some given function of the lengths and angle θ of the incident line segments. (A particularly simple example of the last-mentioned function is, say, the product of the incident line lengths and $|\pi - \theta|$, though other simple choices could be more interesting or realistic.) The problem is to determine the asymptotic behaviour (as $n \rightarrow \infty$) of the lowest cost power line routing between $(1, 1)$ and (n, n) .

As far as we are aware, this type of model, particularly with the penalty cost for acute angles in the polygonal curve, has not been studied in the literature before.

6 Conclusion

In this report, we have addressed a real-life problem faced by the company NM Group, a problem called power line routing. We considered the problem at two different scales of discretisation, namely macro and micro levels. For macro level routing, we proposed a new algorithm that overcomes an issue faced by a previous algorithm used by the company. We also comprehensively addressed the more detailed micro level routing problem. We proposed several different algorithmic solution strategies, one of which is exact, the others of which are heuristic or approximate. In particular, we have encoded the problem in an auxiliary weighted directed graph, the shortest path of which corresponds exactly to a minimum cost solution for the micro level routing problem. This leads to an exact algorithm for determining the optimum that, while theoretically efficient, is unlikely to be computationally practical. So we also proposed several heuristics including path perturbation and metaheuristics, all of which

have decent hope of producing reasonable candidate solutions in reasonable time/space.

Acknowledgments

This research was done during the Study Group Mathematics with Industry 2015 (SWI) in Utrecht. We would like to thank the organisers of the study group for creating the wonderful atmosphere.

The authors thank NM Group representatives Paul Richardson and Chris Williams for presenting the problem and readily answering all of our questions about power line routing and mapping.

We gratefully acknowledge Aleksandra Stojanova, Dusan Bikov, Giorgi Khimshiashvili, Natasha Stojkovicj and Zlatko Varbanov for pleasant conversations during the study group and for their valuable comments and suggestions.

References

- [1] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.
- [2] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959. doi: 10.1007/BF01386390.
- [3] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987. doi: 10.1145/28869.28874.
- [4] F. Glover and G. A. Kochenberger. *Handbook of metaheuristics*. Springer Science & Business Media, 2003.
- [5] F. Glover and E. Taillard. A user’s guide to tabu search. *Annals of Operations Research*, 41(1):1–28, 1993.
- [6] J. M. Hammersley and D. J. A. Welsh. First-passage percolation, subadditive processes, stochastic networks, and generalized renewal theory. In

Proc. Internat. Res. Semin., Statist. Lab., Univ. California, Berkeley, CA,
pages 61–110. Springer-Verlag, New York, 1965.

- [7] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination*. AMS
Chelsea Publishing, 1999.

Patient-adaptive compressed sensing for MRI

Lee Gonzales (Vrije Universiteit Brussel), Koos Huijssen (VORTech), Rakesh Jha (Eindhoven University of Technology), Tristan van Leeuwen (Utrecht University), Alessandro Sbrizzi (UMC Utrecht), Willem van Valenberg (Utrecht University), Ian Zwaan (Eindhoven University of Technology), Johan van den Brink (Philips Healthcare), Elwin de Weerd (Philips Healthcare)

Abstract

The theory of compressed sensing (CS) promises reconstruction of sparse signals with a sampling rate below the Nyquist criterion by taking randomized measurements under certain assumptions about the structure of the signal. Current compressed sensing techniques applied to magnetic resonance imaging (MRI) require measurements concentrated heavily around the center of the Fourier space in order to yield somewhat usable results. PHILIPS Healthcare is interested in the potential time-saving benefits of compressed sensing applied to MRI, but requires robust and accurate reconstruction results. During the 106th European Study Group for Mathematics and Industry, PHILIPS challenged us to improve upon existing Fourier space sampling patterns for compressed sensing. The patterns could, for example, include patient dependent prior information. We demonstrate (experimentally) that current CS-MRI techniques lack a sufficient amount of the property called *incoherence*. Incoherence is a measure of correlation between the measurement matrix and the basis in which the signal is sparse. A compressed sensing method without sufficient incoherence results in sub-optimal performance in terms of scan-time and/or image quality. By introducing the necessary incoherence, biased sampling in the Fourier space is no longer necessary. Increasing incoherence while keeping the similar sparsity level in a practical setting is not as straightforward. In this paper, we demonstrate an off-line approach based on prior patient information and evaluate the results with the structural similarity index (SSIM) as a measure of image reconstruction quality. The developed strategy provides an improved signal basis such that both scan-time reduction and good image reconstruction are attained.

1 Introduction

Magnetic resonance imaging (MRI) is the imaging modality of choice for diagnosing a vast range of diseases, including multiple sclerosis and cancer. Compared to X-ray based techniques, such as computerized tomography (CT), MRI provides superior soft-tissue contrast and does not expose the patient to ionizing radiation. Unfortunately, a typical MRI exam can take over 30 minutes (as compared to 5 minutes for a CT scan). One way to reduce the scan time is to collect fewer measurements. Current image reconstruction techniques, however, require a sampling strategy which obeys the Nyquist criterion, in order to reconstruct a usable image. Recent developments in the field of *compressed sensing* (CS) promise a dramatic reduction of the number of measurements needed to reconstruct an image, as long as the measurement process is *incoherent* and the image can be represented in terms of relatively few basis terms (*sparsity*).

PHILIPS Healthcare is one of the major MRI scanner manufacturers in the world and it is interested in optimization of the CS-MRI paradigm. During the SWI 2015, PHILIPS would like to investigate how the CS-MRI experiment can be optimally performed, in the sense that an accurate image is reconstructed from data obtained within the shortest possible scan-time. Some information about the patient is available prior to the scan thus it can be used to achieve this goal. As PHILIPS suggests, the approach should be *patient-dependent*, that is, the existing data has to be quickly processed to determine the scanner setup for the new scan.

In this report, we discuss how we combine CS and MRI in order to develop reconstruction algorithms that exploit previously acquired patient-specific information. The report is organized as follows. First, we review the mathematics behind basic MRI image reconstruction and compressed sensing and identify a possible bottleneck for the straightforward application of CS theory to MRI imaging. Then, we delineate a strategy to solve this problem and we show some preliminary results we have obtained during the SWI 2015.

2 Preliminaries

2.1 Magnetic resonance imaging

The simplest mathematical model for the MRI imaging process can be stated as follows. We are interested in reconstructing the transverse component of the so-called *spin magnetization vector*, which is tissue-dependent, and thus reveals internal structures. We represent the transverse magnetization of a 2D slice by a function $u(x, y)$ with $x, y \in [0, 1]$. In practice, u is a complex function but for simplicity, we assume $u \in \mathbb{R}$. The work presented in this report can be easily extended to the complex case.

In MRI, we can measure the function u only in the Fourier domain, also called k -space. The 2D Fourier series of u is denoted by

$$\hat{u}_{kl} = \int_0^1 \int_0^1 dx dy u(x, y) e^{i2\pi(kx+ly)},$$

for $k, l \in \mathbb{Z}$. In practice, only the coefficients up to some maximum bandwidth $-B < k, l < B$ are measured, allowing us to reconstruct an $N \times N$ discrete representation of the transverse magnetization $u_{ij} = u(i/(N-1), j/(N-1))$ with $0 \leq i, j < N$. Given N , the Nyquist criterion determines the value of B which makes a correct reconstruction of u from its frequency coefficients possible.

By organizing all the coefficients in vectors, we can state the MRI measurement process as follows

$$\hat{\mathbf{u}} = F\mathbf{u},$$

where $\mathbf{u} \in \mathbb{R}^n$, $\hat{\mathbf{u}} \in \mathbb{C}^n$, $F \in \mathbb{C}^{n \times n}$ represents the 2D discrete Fourier transform and $n = N^2$ denotes total number of pixels. A schematic depiction of the process is shown in figure 1.

Since all the Fourier samples need to be measured sequentially, the time needed to acquire them scales quadratically with the required resolution. In the next section we will discuss an alternative sampling paradigm that promises a more favorable scaling assuming that \mathbf{u} exhibits some additional structure.

2.2 Compressed sensing

The basic idea behind compressed sensing is that we can uniquely solve an underdetermined system $A\mathbf{x} = \mathbf{b}$ given that the solution we seek is *sparse* (i.e., has only a few non-zero elements) and the matrix A satisfies the so-called *restricted isometry property* (RIP).

Definition 2.1. A vector $\mathbf{x} \in \mathbb{R}^n$ is k -sparse when it has at most k non-zero elements.

Definition 2.2. A matrix $A \in \mathbb{R}^{m \times n}$ satisfies the restricted isometry property RIP(k, δ_k) if for every k -sparse vector \mathbf{x} there exists a constant δ_k such that

$$(1 - \delta_k)\|\mathbf{x}\|_2 \leq \|A\mathbf{x}\|_2 \leq (1 + \delta_k)\|\mathbf{x}\|_2.$$

When the measurement matrix A satisfies the RIP and the solution x is sparse, or well approximated by a sparse solution, then one can solve the so-called *basis pursuit denoise* (BPDN) problem

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \|A\mathbf{x} - \mathbf{b}\|_2 \leq \sigma.$$

Here, $\sigma \geq 0$ is the noise level of the measurement \mathbf{b} .

With these definitions we can now state the following theorem by Candès [3] regarding the recoverability of a sparse signal from noisy measurements.

Theorem 2.1. *Let the matrix A satisfies RIP($2k, \delta_{2k}$) with $\delta_{2k} < \sqrt{2} - 1$, and $\mathbf{b} = A\mathbf{x} + \mathbf{n}$ for given signal \mathbf{x} and $\|\mathbf{n}\|_2 \leq \epsilon$. Then, the error between the solution $\bar{\mathbf{x}}$ of the BPDN problem and the true signal \mathbf{x} is bounded as follows:*

$$\|\bar{\mathbf{x}} - \mathbf{x}\|_2 \leq C_0\|\mathbf{x} - \mathbf{x}_k\|_1/\sqrt{k} + C_1\epsilon,$$

where C_0 and C_1 are positive constants and \mathbf{x}_k is the best k -sparse approximation to \mathbf{x} . Thus, if the given signal \mathbf{x} is k -sparse, we have $\|\bar{\mathbf{x}} - \mathbf{x}\|_2 \leq C_1\epsilon$

With overwhelming probability, certain types of random matrices (e.g., matrices whose elements are i.i.d. Gaussian) satisfy the required RIP property when

$$m \geq Ck \log(n),$$

where C is a problem-specific constant [3]. For our MRI problem, this would mean that the number of measurements is no longer driven by the resolution but by the sparsity of the signal. It is in practice not feasible to check whether a given matrix satisfies RIP. Instead, one typically considers the *coherence*.

In practical applications, the signal of interest is not sparse itself, but admits a sparse representation in some orthonormal basis $\Psi \in \mathbb{C}^{n \times n}$. Modelling the measurement process as taking inner products of the signal with m rows of an orthonormal basis $\Phi \in \mathbb{C}^{n \times n}$, we can express the sensing matrix as $A = R\Phi\Psi$, where $R \in \mathbb{R}^{m \times n}$ selects m rows at random. The resulting matrix A is a suitable RIP when the *mutual coherence* between Ψ and Φ is low.

Definition 2.3. The mutual coherence of two orthonormal bases Ψ and Φ is defined as

$$\mu(\Psi, \Phi) = \max_{1 \leq i, j \leq n} |(\Psi^T \Phi)_{ij}|,$$

Generally, the lower the coherence, the lower the RIP constant and the fewer measurements we expect to need in order to recover a given sparse signal. Note that for orthonormal bases we have $1 \leq \mu \leq \sqrt{n}$. In the remainder of the paper we will use the coherence as a heuristic to gauge how well a given pair (Ψ, Φ) is expected to perform.

2.3 Sparse recovery

There are a number of algorithms for solving the BPDN problem, most of which are based on one of two equivalent reformulations. The first is quadratic formulation of the problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \lambda \|A\mathbf{x} - \mathbf{y}\|_2,$$

and the second is the Lasso problem

$$\min_{\mathbf{x}} \frac{1}{2} \|A\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_1 \leq \tau.$$

For a given σ , there exists a unique λ and τ such that the solutions of all three problems coincide [5, 2]. The relation between these parameters is given by the *Pareto curve*. This is illustrated in figure 2.

Finding these parameters λ or τ is not trivial, however, and typically relies on some sort of continuation method. A very elegant way of finding a τ corresponding to a given σ is described by [5]. Essentially, they develop a root-finding method to traverse the Pareto curve. The Lasso subproblems are solved via a projected gradient algorithm and is suitable for large-scale problems and complex data.

3 Approach and results

We have seen that the two important ingredients in CS are sparsity and coherence. The image needs to be as sparse as possible while the coherence needs to be as low as possible. The goal is to leverage these results to reduce the number of measurements needed to recover the image \mathbf{u} from Fourier measurements

$$\mathbf{b} = R\mathbf{F}\mathbf{u},$$

where $R \in \mathbb{R}^{m \times n}$ with $m < n$ is a restriction matrix that subsamples the full Fourier measurements. Since we are interested in taking as few measurements as possible, we want m to be as small as possible. Introducing the subsampling ratio $\rho = n/m$, the potential speedup of the measurement process is proportional to ρ .

A typical image is not sparse in the natural pixel basis. Instead, we need to find a basis in which the image can be sparsely represented. A common choice is wavelets, denoted here by a matrix W . The sparsity is illustrated in figure 3.

The recovery problem now is to find wavelet coefficients $\mathbf{z} = W\mathbf{x}$ such that $R\mathbf{F}W^T\mathbf{z} \approx \mathbf{b}$. A problem with this approach is that the mutual coherence $\mu(\mathbf{F}, W^T)$ is quite high, as illustrated in figure 4. A remedy to this is to insert a random ± 1 diagonal matrix in the measurement process and collect measurements as $\mathbf{b} = R\mathbf{F}\mathbf{S}\mathbf{u}$. Here, $\mathbf{S} = \text{diag}(\mathbf{s})$ where the $s_i \in \{+1, -1\}$ are i.i.d. Rademacher random variables [1, 4]. The mutual coherence $\mu(\mathbf{F}\mathbf{S}, W^T)$ is much lower, as illustrated in figure 4.

Reconstructions with and without \mathbf{S} for homogeneous random sampling are presented in figure 5. We clearly see that incorporating the matrix \mathbf{S} spreads the information more evenly over the Fourier space so that uniform random sampling makes more sense. Comparing the reconstruction quality in terms of the Structural Similarity Index Measure (SSIM) [6] for different subsampling ratios in figure 6 we see that incorporating \mathbf{S} allows for a higher subsampling factor.

Unfortunately, it is not feasible in practice to incorporate \mathbf{S} in the sampling as it would entail randomly perturbing the object prior to taking the measurement. Therefore, we will take a slightly different view on the problem.

3.1 Breaking the coherence

Instead of changing the measurement process, we will aim to find a basis \widetilde{W} that reduces the mutual coherence. An obvious candidate is $\widetilde{W} = SW$, however, typical images are much less sparse in this new basis as illustrated in figure 7.

This leads us to two extreme cases: for $\mathbf{s} = \mathbf{1}$ we have a very sparse representation and a high coherence while for uniformly random $s_i \in \{-1, +1\}$ we have a low coherence and insufficient sparsity. This is shown in figure 8. The question is whether there exists a mask \mathbf{s} that achieves an “optimal” trade-off between the two extremes. To investigate this we take the s_i to be correlated Rademacher variables or random checkerboard patterns and vary the scale to obtain a natural continuation from one extreme to the other. A few examples are shown in figure 9. For these matrices, we compute the coherence and the sparsity and plot them. Figure 10 shows that they indeed trace out a trade-off curve as argued earlier. Figure 11 shows the reconstruction quality for these various masks. Unfortunately, the partially coherent masks perform only marginally better for small subsampling ratios than the other two.

3.2 Experimental design

The goal of this section is to see if we can do better when we explicitly design S to minimize the coherence while maintaining the sparsity of a reference image \mathbf{u} . We could formulate this problem as

$$\min_{\mathbf{s}} \mu(F, SW^T) \quad \text{s.t.} \quad \|WS^{-1}\mathbf{u}\|_1 \leq (1 + \kappa)\|W\mathbf{u}\|_1,$$

with κ small. The inequality constraint ensures that the application of the new transform WS^{-1} to \mathbf{u} gives a sparse representation.

If we take S to be a diagonal matrix $S = \mathbf{diag}(\mathbf{s})$ we can formulate this problem as

$$\min_{\mathbf{s}} \|A\mathbf{s}\|_{\infty} \quad \text{s.t.} \quad \|U(\mathbf{s})\|_1 \leq (1 + \kappa), \quad (1)$$

where A is the matrix representation of the linear operation $F^T \mathbf{diag}(\mathbf{s}) W^T$ (i.e., $A\mathbf{s} = \text{vec}(F^T \mathbf{diag}(\mathbf{s}) W^T)$) and $U(\mathbf{s}) = \frac{1}{\|W\mathbf{u}\|_1} W\mathbf{v}$ where $v_i \equiv 1/s_i$.

To assess the feasibility of this approach, we solve this optimization problem for a 16×16 reference image, shown in figure 12 and denoted by *True image*.

We use a black-box non-linear optimization routine in Matlab (`fmincon`) to solve the optimization problem. The starting value for \mathbf{s} is the vector whose components follow a Rademacher distribution as introduced above, that is: $s_i \in \{1, -1\}$. This is shown in figure 12. The algorithm is manually halted after 40 iteration, when the convergence reaches a plateau. The convergence history is shown in figure 13.

The resulting solution is shown in figure 12 and it is denoted by *Optimized S*. Interestingly, the result has a similar structure as the reference image. This can be understood as follows. The reference image is multiplied point-wise with \mathbf{s} . If we take $s_i \approx u_i^{-1}$, the resulting normalized image will be almost constant, thus allowing for a very sparse approximation using only the coarsest scale wavelets.

We perform CS reconstructions of the test image with $N = 16$ and reduction factor 2 for the sparsity transforms W (standard approach) and WS^{-1} , respectively. To appreciate the improvement obtained by the experimental design algorithm, we also consider the reconstruction with the starting value for S , that is, the Rademacher distributed values.

The reconstructed images are shown in figure 12, bottom row. We consider the relative error given by $\|\mathbf{u} - \mathbf{u}^r\|_2 / \|\mathbf{u}\|_2 \times 100\%$ where the superscript r denotes the reconstructed image and we report the error value under the corresponding plots. Note the drastic reduction in the error when the optimized S is used.

4 Discussion

We have seen that the coherence between the Fourier transform and wavelets leads to suboptimal performance of CS-type reconstructions and we have delineated a strategy to modify the Wavelet transform by means of the S matrix. The resulting transform maintains the sparsity and at the same time minimizes the coherence with the sampling operator RF . This trade-off solution gives excellent results in terms of CS reconstructions.

Note that the steps (experiment design and CS reconstructions) can be performed *off-line*, that is, after the end of the MRI exam. In this way, there is no surcharge of time for the clinical protocol, a major drawback for *on-line* design methods.

Our approach relies on the knowledge of a reference image. The question is, of course, how this approach will perform when the reference image is not the same as the true image. We expect minimal problems when the reference image is a previously acquired scan, a set of reference images, or other prior information. Alternatively, we could perform a first reconstruction by standard CS and use the resulting image as reference for designing S . This step could be repeated until no improvement is obtained with respect to the previously reconstructed image.

5 Conclusions

We have analyzed the pitfalls of CS applied to MRI and we have presented an innovative approach to improve the reconstructions. The large scale optimization problem can be performed off-line, making the way to the clinic potentially short.

References

- [1] N. Ailon and B. Chazelle. The Fast Johnson-Lindenstrauss Transform and Approximate Nearest Neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009. doi: 10.1137/060673096. URL <http://dx.doi.org/10.1137/060673096>.
- [2] A. Aravkin, J. Burke, and M. Friedlander. Variational properties of value functions. *SIAM Journal on optimization*, 23:1689–1717, Nov. 2013. ISSN 1052-6234. doi: 10.1137/120899157. URL <http://arxiv.org/abs/1211.3724><http://epubs.siam.org/doi/abs/10.1137/120899157>.
- [3] E. Candes and M. Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(March 2008):21–30, 2008. ISSN 1053-5888. doi: 10.1109/MSP.2007.914731.
- [4] F. Krahmer and R. Ward. New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. *ArXiv e-prints*, Sept. 2010.
- [5] E. Van Den Berg and M. Friedlander. PROBING THE PARETO FRONTIER FOR BASIS PURSUIT SOLUTIONS. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008. ISSN 1064-8275.

URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.9332&rep=rep1&type=pdf>.

- [6] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. Wavelets for Image Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

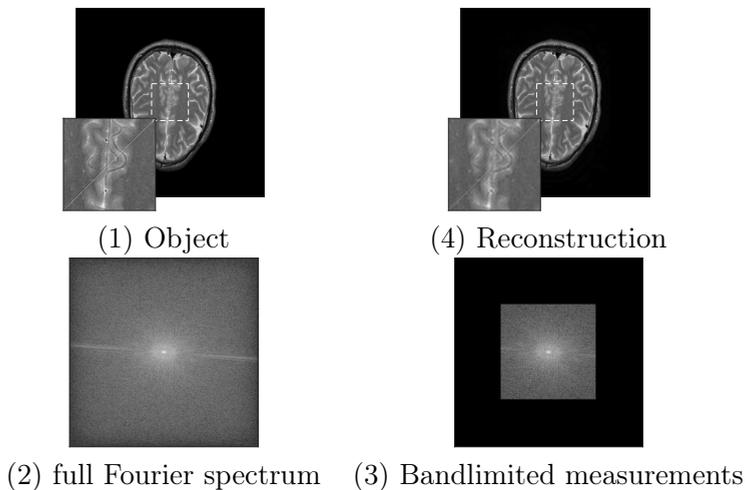


Figure 1: Schematic depiction of the MRI process. The ground object (1) is sampled in the Fourier domain (2), yielding a set of bandlimited measurements (3) from which we can reconstruct using a discrete inverse Fourier transform (4). Note the slight loss in resolution caused by the bandlimited nature of the measurements.

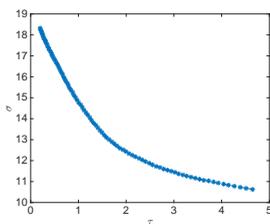


Figure 2: Schematic depiction of the Pareto curve, which relates the optimal solutions to the LASSO, BPDN and QP formulations of the sparse recovery problem. At a give (τ, σ) , the derivative of the curve is proportional to λ .

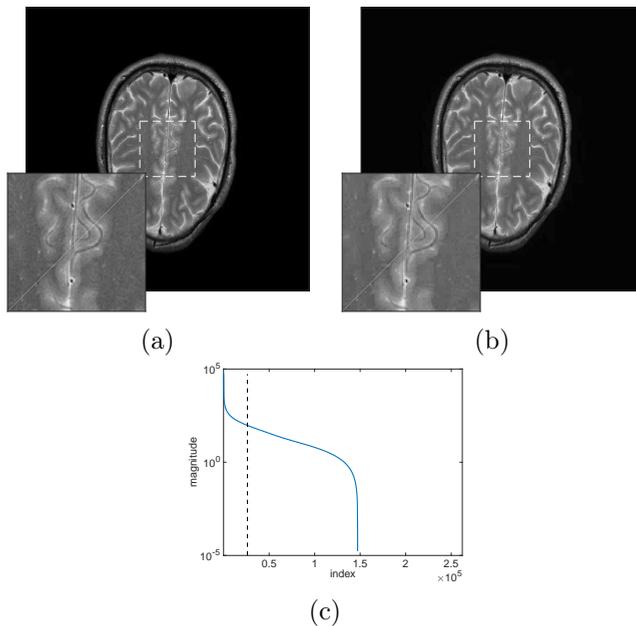


Figure 3: Sparsity in wavelets. (a) original image, (b) image using only 10% of the largest Wavelet coefficients, (c) magnitude of the Wavelet coefficients, the vertical line indicates the cut-off used to produce image (b).

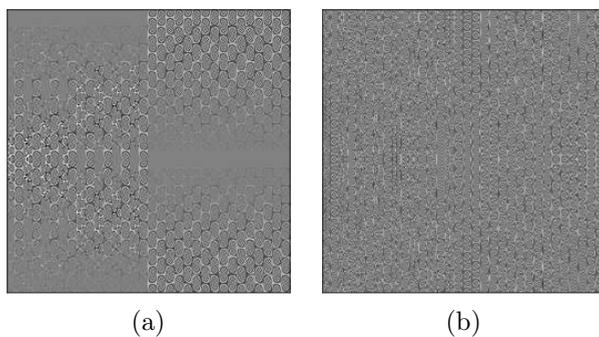


Figure 4: Coherence of (a) F and W^T and (b) FS and W^T . We see that the second matrix is much less coherent and hence is better suited for CS reconstruction.

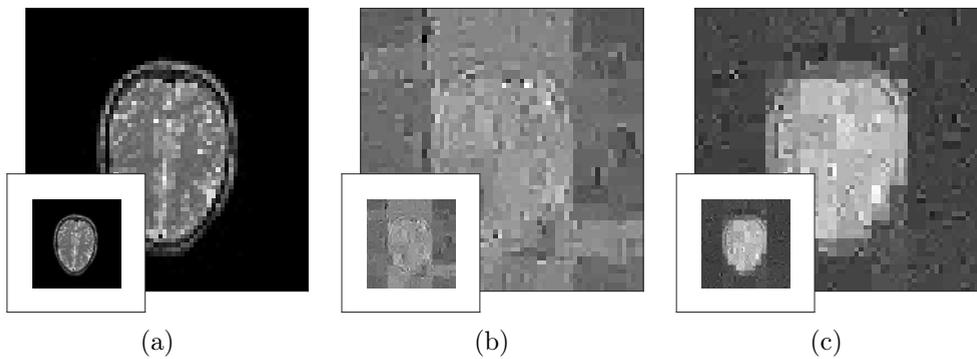


Figure 5: (a) original image, (b) reconstruction with $\rho = 8$ without S , (c) reconstruction with $\rho = 8$ with S . The latter clearly gives a much better reconstruction, illustrating the importance of including the matrix S .

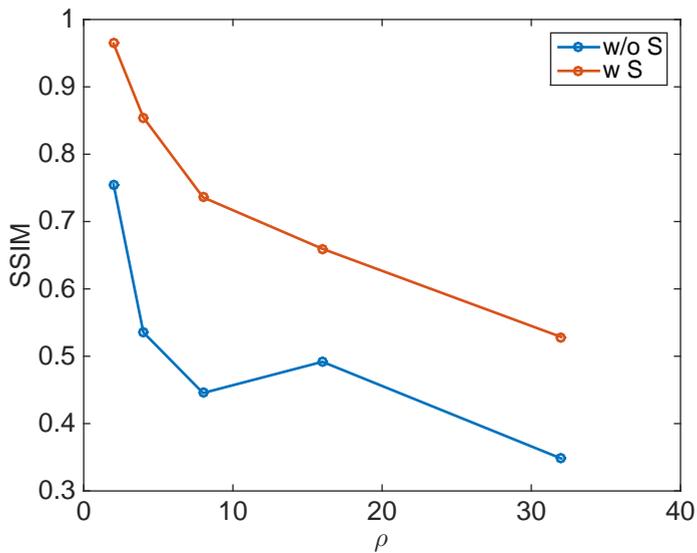


Figure 6: Reconstruction quality (in terms of the SSIM) for various subsampling ratios.

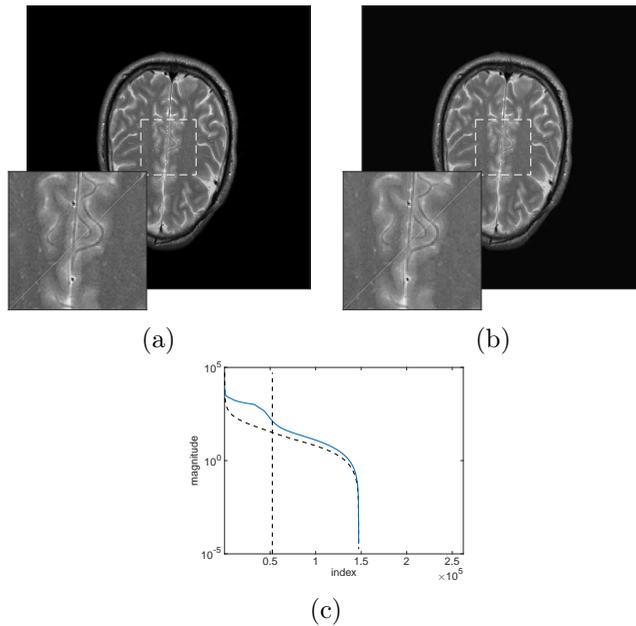


Figure 7: Sparsity in modified wavelets WS . (a) original image, (b) image using 20% of the largest Wavelet coefficients, (c) magnitude of the Wavelet coefficients. The dotted line indicates the magnitude of $W\mathbf{u}$ while the solid line indicates the magnitude of $WS\mathbf{u}$. The vertical line indicates the cut-off used to produce image (b). We see that the original image is less sparse in the modified wavelets.

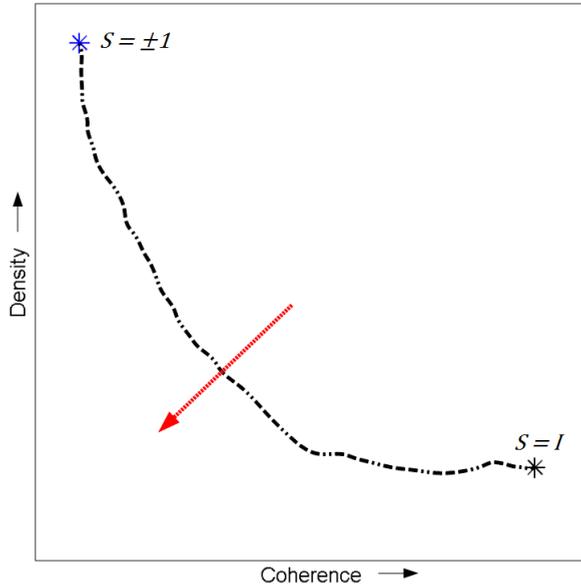


Figure 8: Schematic depiction of the tradeoff between sparsity and coherence.

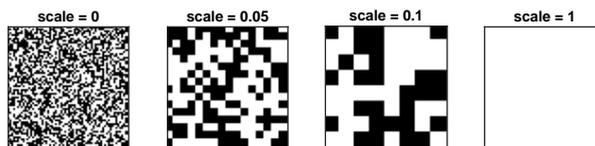


Figure 9: Examples of partial coherent masks, ranging from completely incoherent (left) to completely coherent (right).

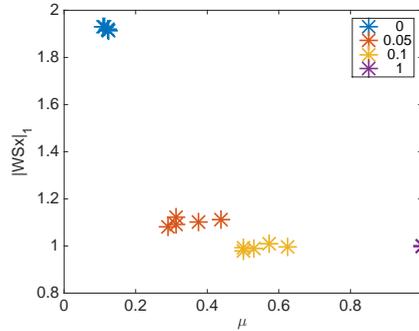


Figure 10: Tradeoff between sparsity and coherence for various image masks, ranging from completely incoherent (scale = 0) to completely coherent (scale = 1).

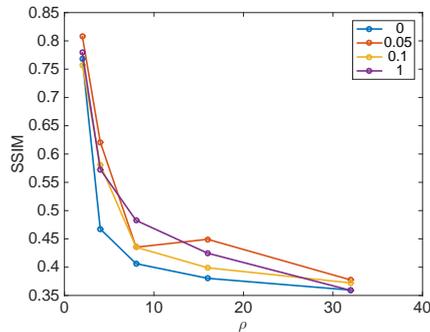


Figure 11: Reconstruction quality using modified wavelets with various image masks, ranging from completely incoherent to completely coherent. We see that the partially coherent masks with scales 0.05 and 0.1 perform slightly better than the other two.

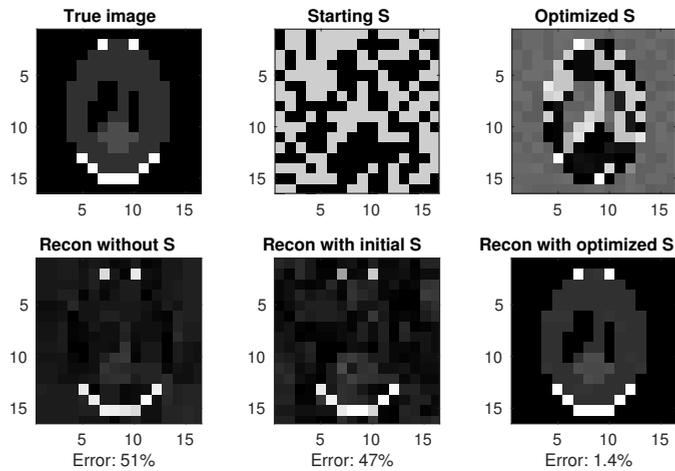


Figure 12: Experimental design. Top row: The ground truth image, the starting and optimized S , respectively. Bottom row: The three reconstructions, obtained without S , with the starting S and with the optimized S . Note the drastic improvement in the obtained image when the optimized S is employed.

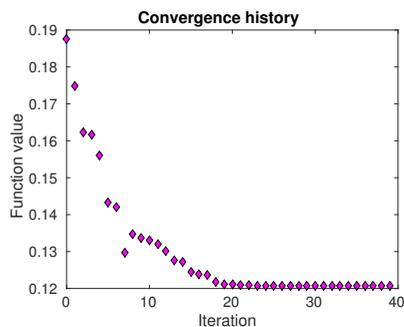


Figure 13: Experimental design. Convergence history for the design algorithm (Eq. 1).

A Matlab framework for rapid prototyping

To test the proposed algorithms, we used the SPOT toolbox, which allows us to define matrix-free linear operators. This toolbox allows us to use standard Matlab matrix-vector notation and manipulation while avoiding explicitly storing dense matrices. A 2D Fourier transform of a 2D signal u , for example, can be defined as follows.

```
F = opDFT2(n,n);  
ut = F*u;
```

Here, the Fourier operator F acts like a matrix, but upon multiplication it calls `fft2`. We can construct new operators by simply multiplying them together. The MRI measurement process, for example, is implemented as follows.

```
F = opDFT2(n,n);  
I = randperm(n);  
I = I(1:m);  
R = opRestriction(n,I);  
y = R*F*u;
```

For the sparse reconstruction we use `spgl1`. A complete reconstruction then, is done as follows.

```
F = opDFT2(n,n);  
I = randperm(n);  
I = I(1:m);  
R = opRestriction(n,I);  
y = R*F*u;  
W = opWavelet2(n,n,'Haar');  
z1 = spgl1(R*F*W',y,[],sigma);  
u1 = W'*z1;
```

Statistical Modeling of Mechanical Bearing Life Testing

Sébastien Blachère (SKF), Martin Bootsma (Utrecht University), Alessandro Di Bucchianico (Eindhoven University of Technology), Mike Keane (Delft University of Technology), Xinru Li (Leiden University), Andrea Roccaverde (Leiden University), Cristian Spitoni* (Utrecht University), Dong Yan (Leiden University)

Abstract

We investigate reliability test plans under different censoring schemes for estimating performance of bearings with different life characteristics. The test plans, which are based on Weibull distributions, should deliver estimates of performance characteristics with a specified precision. We present results on both a theoretical approach based on Fisher information and a simulation approach.

Keywords: bearing, lifetime, Weibull distribution, censoring, test plan, Fisher information

1 Introduction

1.1 About SKF

SKF is a global technology provider offering products and services related to bearings and units, seals, mechatronics and lubrication systems. Its headquarters are located in Sweden. The company has around 165 production sites in 28 countries. SKF has several research centres, including one in the Netherlands in Nieuwegein.

Mechanical bearings (see Figure 1) are an important product of SKF. They are mechanical elements that constrain motions to desired motions only, and at the same time reduce friction between moving parts. There is a wide range of applications of bearings, including bicycles, cars, manufacturing machines, trains, wind turbines and airplanes (see Figure 2). Sizes of bearings range from less than 10 mm to 14 m. Since bearings are essential for the proper and safe functioning of machines and equipment, it is essential for SKF to give customers reliable information on the performance. The performance of mechanical bearings is



Figure 1: Bearings.

*Corresponding author

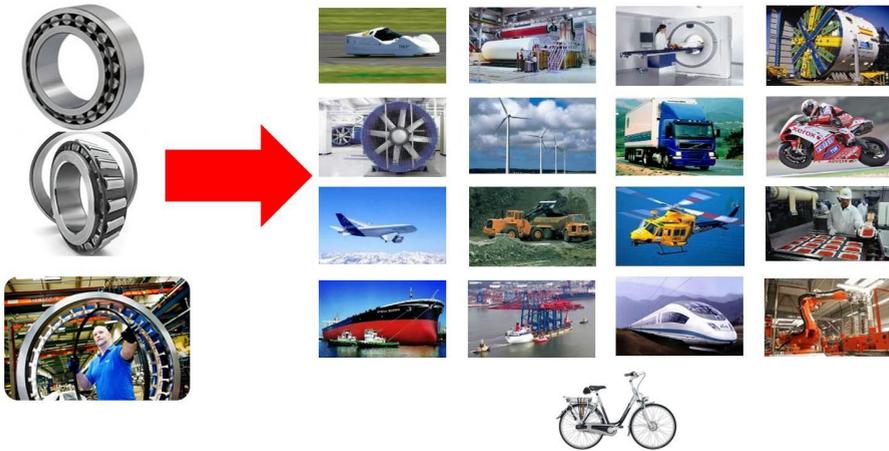


Figure 2: Examples of applications of mechanical bearings.

expressed through their life, i.e., the amount of time or number of revolutions that a bearing is capable to reach within nominal functioning. Bearing life depends on various parameters like the bearing type or size and the operating conditions (speed, load, lubrication, . . .). SKF uses an internal calculation tool based on physical models to assess bearing life. However, there is a need for life testing on actual bearings in order to validate these models, evaluate performance of prototypes and obtain insight in effect of design choices. Even identical bearings running under identical operating conditions may experience a wide dispersion in their life. The ratio of the longest to the shortest life may exceed 100 in large samples.



Figure 3: Test rigs used in bearing life tests.

1.2 Outline of the problem

Life tests consist of running a group of bearings under identical operating conditions until a stopping criterion is fulfilled. There exist three classical stopping criteria:

- **Type I.** The test is stopped when a preset time has been reached. This is illustrated in the left-hand side of Figure 4.
- **Type II.** The test is stopped when a preset number of failures has been reached. This is illustrated in the right-hand side of Figure 4.
- **Hybrid.** The test is stopped when either a preset time or a preset number of failures has been reached. This is illustrated in Figure 5 .

Life times of both failed and not yet failed bearings are recorded since both types of data contain information. Usually most bearings have not yet failed at the end of the life test. As usual in reliability engineering, the data is modelled using a Weibull distribution.

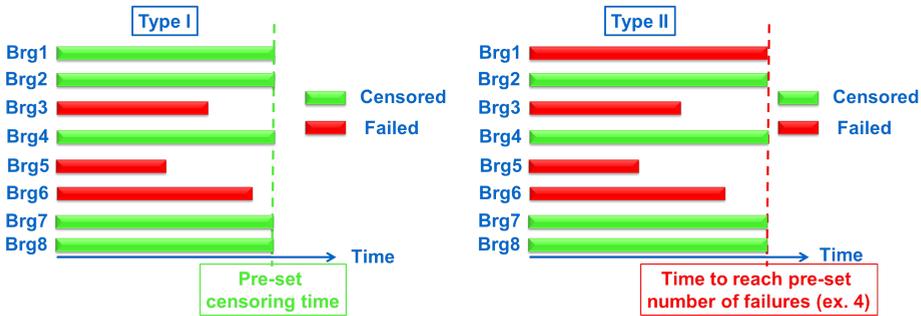


Figure 4: Type I and II stopping criteria.

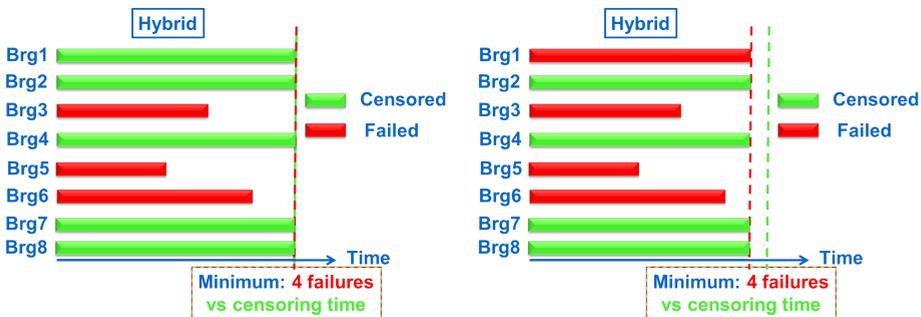


Figure 5: Hybrid stopping criterion.

The objective of life tests is the estimation of the Weibull parameters (see Subsection 2.1 for details). The precision of such an estimation depends strongly on the test strategy (number of bearings tested, test duration, number of observed failures, replacement policy). Therefore, obtaining precise estimation of the Weibull parameters

may require lengthy (up to several months) and thus costly tests. It is thus necessary to understand the link between the test duration and the precision of the estimation in order to optimize the test time while reaching the target precision. The precision of the estimation is traditionally defined as the length of the confidence intervals for each of the parameters (see Subsection 2.1 for details) on a logarithmic scale:

$$R_{10}(L_{10}) = \frac{L_{10,95}}{L_{10,05}} \quad \text{and} \quad R_{10}(\beta) = \frac{\beta_{95}}{\beta_{05}},$$

where $(L_{10,05}, L_{10,95})$ and (β_{05}, β_{95}) are the 90% two-sided confidence intervals for L_{10} and β , respectively.

Problem statement *Given N life tests (i.e., life tests with bearings of the same type and conducted under identical conditions) to be run within the same fixed test capability, what is the strategy to follow to minimize the total test duration of all N life tests with a given confidence to obtain parameter estimates with a preset precision?*

When designing a test strategy, the precision and the available bearings of a certain type are constraints. The degrees of freedom are the sample size, the type of stopping criterion, the replacement policy and the value associated to this stopping criterion (preset time and/or preset number of failures). Some extra degrees of freedom can be added like the replacement of failed bearings by new ones for instance, but they are not treated in this article.

In addition, both the test time and the test precision are depending on the test strategy in a stochastic way. Therefore, the specifications onto the precision and the time need to be expressed in terms of their distributions via the mean, standard deviation or some percentiles.

1.3 Approach

This report reflects our first attempt to tackle the general problem. We did so by concentrating ourselves on a simpler problem where we had only 1 type of bearing. For this case, we derived theoretical results based on Fisher information that we will allow us in the future to compare different testing strategies. We complemented these theoretical results by running simulations for different testing strategies based on a fast R code that we developed ourselves.

1.4 Outline of this article

This article is organized as follows. In Section 2 we provide details on the Weibull distribution (parametrizations, estimation for the different censoring schemes). Section 3 contains our theoretical approach to test plans based on the Fisher information. The results of our numerical simulations can be found in Section 4. We end our paper with conclusions and recommendations in Section 5.

2 Background on the model

2.1 Weibull Distribution

The Weibull distribution has been introduced in the setting of material strength by Waloddi Weibull (Weibull (1939)) and was later extended to a wide range of types of experimental data (Weibull (1951)). The Weibull distribution is one of the extreme value distributions and appears if one considers minima of random variables (which is very natural in material strength since material break when the weakest link fails). Another motivation for the use of the Weibull distribution is that the hazard rate is flexible since it is a power function and may thus model both increasing and decreasing failure rates (depending on the sign of the exponent).

The Weibull distribution is extensively used in reliability theory together with its special case, the exponential distribution. The Weibull distribution possesses two main forms, one with 2 parameters (with domain $(0, \infty)$) and one with 3 parameters (which has an additional location parameter so that the domain need not start at 0).

Here we will describe the background on the Weibull distribution (parameterizations, parameter estimation, censoring, ...). For a comprehensive overview on the Weibull distribution, we refer to the excellent monograph Rinne (2008).

The two-parameter Weibull distribution has a scale parameter and a shape parameter. The standard representation in term of the cumulative distribution function is

$$F(x|\alpha, \beta) = 1 - e^{-\left(\frac{x}{\alpha}\right)^\beta} \quad (1)$$

The parameter α is the scale parameter, while the parameter β is the shape parameter. The exponential distribution is included as the special case $\beta = 1$. Note that for $\beta > 1$ the Weibull distribution has an increasing failure rate. Here we will use another representation of the Weibull distribution which has the same shape parameter β , but a different scale parameter, L_{10} . In this representation, the cumulative density function has the form:

$$F(x|L_{10}, \beta) = 1 - \left(\frac{9}{10}\right)^{\left(\frac{x}{L_{10}}\right)^\beta}. \quad (2)$$

The parameter L_{10} has a clear interpretation; it is the 10%-quantile of the distribution, i.e. 90% of the bearings survive at least until time L_{10} . The parameters are linked to each other through the relation

$$L_{10} = \alpha (-\log(9/10))^{1/\beta}. \quad (3)$$



Figure 6: Weibull.

The Weibull distribution can also be described by the survival function

$$S(x|L_{10}, \beta) = \left(\frac{9}{10}\right)^{\left(\frac{x}{L_{10}}\right)^\beta}. \quad (4)$$

In this paper, we will consider L_{10} as a parameter of the Weibull distribution. However, if one wants to estimate the 10%-quantile of the distribution one could estimate L_{10} also non-parametrically.

The probability density function of the Weibull distribution in this parameterization is given by

$$f(x|L_{10}, \beta) = \frac{1}{L_{10}} \left(\frac{9}{10}\right)^{\left(\frac{x}{L_{10}}\right)^\beta} \left(\frac{x}{L_{10}}\right)^{\beta-1} \beta \log\left(\frac{10}{9}\right). \quad (5)$$

Note that in the remaining part we suppress in the notation for the cumulative distribution function and the probability density function sometimes the dependence on L_{10} and β to obtain clearer expressions.

2.2 Censoring

In real-life testing of bearings, it is impossible to test until all bearings have failed. This is also the case if one uses accelerated life testing, i.e. testing under more severe conditions than normal in such a way that using physical degradation laws one can connect life times under severe conditions to life times under normal conditions (see e.g., Meeker and Escobar (1998)). The data that we obtain from life tests will thus include data on non-failed items. Such observations are called *censored observations*. This terminology is used in engineering contexts (life tests of physical objects) as well as in medical contexts (clinical trials). Note that censored observations in the context of life tests do contain useful information on life times (one usually has a lower bound for the actual lifetimes). In order not to lose information, it is thus important to use statistical techniques that make use of both the censored and uncensored observations. This very much applies to the case of life tests for bearings, since typically the majority of observations is censored.

In order to include censored observations in statistical analyses, it is necessary to model the censoring mechanism at hand. In medical contexts (survival analysis) it is common to model censoring as a random variable independent of the lifetime distribution. In engineering context (reliability theory) it is more common to model censoring in a different way (see e.g., (Rinne, 2008, Section 8.3.1) for a detailed discussion about the different ways of modelling censoring mechanisms). We now consider in more detail censoring types that are common in reliability theory (they also appear in medical contexts under the name administrative censoring). We will restrict ourselves to the case of a single right-censoring. For multiple censoring schemes such as progressive censoring, we refer to the literature (see e.g., Ng et al. (2004) and (Rinne, 2008, Sections 8.3.3 and 8.3.4)).

Type I censoring

Suppose we run n bearings on n machines and we stop the experiment at a preset time T . The number r of failed items at time T is then the realisation of a random variable R . We have thus observed r failures at times $\{x_{(1)}, x_{(2)}, \dots, x_{(r)}\}$ and we have not observed the failure of the other $n - r$ bearings. With the notation $x_{(i)}$ we denote the i^{th} order statistic, i.e., the i^{th} smallest observed failure time. The joint probability density function to observe the failure times is given by:

$$f(x_1, \dots, x_n | L_{10}, \beta) = \frac{n!}{(n-r)!} \prod_{i=1}^r f(x_{(i)} | L_{10}, \beta) (1-F)^{n-r}(T | L_{10}, \beta). \quad (6)$$

for $0 \leq x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(r)}$ (where $r := \max\{i : x_{(i)} \leq T\}$) and equals 0 otherwise. Note that for this type of censoring (known as administrative censoring in the context of survival analysis) the number of failures r is random and the end time is deterministic.

Type II censoring

Suppose we run n bearings on n machines and we stop the experiment once we have $k \leq n$ failures. At the stopping time we have observed k failures at times $\{x_{(1)}, x_{(2)}, \dots, x_{(k)}\}$ and we have not observed the failure of the other $n - k$ bearings. The joint probability density function to observe the failure times is given by:

$$f(x_1, \dots, x_n | L_{10}, \beta) = \frac{n!}{(n-k)!} \prod_{i=1}^k f(x_{(i)} | L_{10}, \beta) (1-F)^{n-k}(x_{(k)} | L_{10}, \beta). \quad (7)$$

which is defined for $0 \leq x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(k)}$. Note that in this type of censoring the number k of failed items is fixed, while the end time is random (it equals the k^{th} order statistic).

Hybrid censoring

An alternative censoring scheme is possible by combining the stopping criteria of Type I and Type II censoring, i.e. we stop the experiment when either we reach the preset time T or the preset number r of failures. In other words, we stop the experiment at the random time $\min(T, X_{(k)})$. This type of hybrid censoring was introduced in Epstein (1954). It is called type-I hybrid censoring in Balakrishnan and Kundu (2013).

Type I censoring is appealing from a practical point of view, since it fixes the duration of the experiment. A mathematical drawback is that it is harder to analyse (one needs to take into account the random time between T and the last failure time before T). Type II is appealing from a mathematical point of view since it is easy to analyse, because the number of failures is deterministic. The practical drawback is that one has no control on the duration of the experiment. A possible drawback

of both Type I and hybrid censoring is that if T is chosen too small, that there are may be too few failures and the resulting estimates are poor. In order to overcome this drawback, Childs et al. (2003) introduced an alternative hybrid censoring scheme based on $\max(T, X_{(k)})$. However, this has the same practical drawback as type II censoring.

2.3 Parameter estimation

In this subsection we discuss estimation of the L_{10} and β parameters of the two-parameter Weibull distribution under the three censoring schemes mentioned in the previous subsection. We will assume that failed items are not replaced during a life test. Maximum Likelihood is the preferred way of estimating parameters in reliability theory, because it not only has well-known asymptotic optimality properties but as exemplified by Formulas (6) and (7) it can easily deal with censored data (unlike e.g. the method of moments). The literature mostly deals with estimation for type II censoring, under the assumption that type I censoring can be dealt with in a similar way by conditioning on the number of failures in the interval $[0, T]$ (cf. Remark 26 on page 438 of Rinne (2008)). An exception is Cohen (1965) which treats both types of censoring.

For Type II censoring the standard approach is to take derivatives of the loglikelihood equation with respect to the parameters and to note that after simplification of the equations one obtains $\hat{\lambda}$ through the following relation (Cohen (1965)):

$$\hat{\lambda} = \left(\sum_{i=1}^n x_i^{\hat{\beta}} \right)^{\hat{\beta}}. \quad (8)$$

The following equation determines $\hat{\beta}$ in case of type II censoring:

$$\frac{1}{\hat{\beta}} = \frac{\sum_{i=1}^n x_i^{\hat{\beta}} \log(x_i) + (n-k) x_{(k)}^{\hat{\beta}} \log(x_{(k)})}{\sum_{i=1}^n x_i^{\hat{\beta}} + (n-k) x_{(k)}^{\hat{\beta}}} - \frac{1}{k} \sum_{i=1}^k \log(x_{(i)}) \quad (9)$$

A similar relation holds in case of type I censoring. For hybrid censoring, Kundu (2007) describes that one has basically the same procedure where the form of the likelihood depends on which of the stopping criteria applies to the data set at hand. Existence and uniqueness of solutions of (9) are guaranteed unless all observations are equal (see e.g. Farnum and Booth (1997), Pike (1966)). Since the right-hand side of (9) can be proven to be an increasing function of β (see Farnum and Booth (1997)¹, numerical procedures like Newton-Raphson quickly yield numerical solutions. There are also exist explicit approximate ML estimators based on Taylor expansions of the logarithm of the Weibull distribution (so transforming the Weibull distribution into an extreme value distribution, see e.g. Kundu (2007)) for details).

¹The proofs in Farnum and Booth (1997) are only written down for the uncensored case, but it can easily be shown that a slight adaptation make them work for the censored cases as well

The ML estimators are biased. Corrections are possible by noting that the distribution of $\hat{\beta}$ is a pivotal quantity for β , i.e. the distribution of $\hat{\beta}/\beta$ only depends on n and depending on the type of censoring T or k (see e.g., McCool (1970) who evaluates this distribution by Monte Carlo simulation). For a complete discussion of pivotal quantities we refer to Bain and Engelhardt (1991) and McCool (1970).

In view of the Invariance Principle of Maximum Likelihood estimation (Zehna (1966)), it does not matter which parametrization of the Weibull distribution one chooses when one is only interested in the parameter estimates. However, it does make a difference when computing confidence intervals for the Weibull parameters. This is because the lack of formulas for exact confidence intervals necessitates to use asymptotic intervals for which the width directly depends on the asymptotic standard deviation of the parameter estimator. Explicit asymptotic confidence intervals for the Weibull parameters have been discussed in Meeker and Nelson (1976) and Kahle (1996). Both papers used observed Fisher information, but Meeker and Nelson (1976) do this for the logarithm of the Weibull distribution (so an extreme value distribution) since the asymptotic sampling distribution in that case converges faster. The formulas in both papers involve second derivatives of the incomplete gamma function, which can be expressed in terms of other special functions (see Geddes et al. (1990)). We followed the approach of Meeker and Nelson (1976) but used direct numerical integration to evaluate the integrals directly since there were no convergence problems.

3 Theoretical results

If we have a single bearing, we have a continuous decision process as long as the bearing has not failed: do we keep the experiment running or do we stop the experiment and replace the existing bearing with a new one. Which of the two options is most attractive depends on several factors: 1) the amount of information one is expected to get from each choice, 2) how fast this information is obtained and 3) the cost of replacing an existing bearing with a new one. Here we neglect the costs completely and we focus on the first two factors.

3.1 Type I censoring

Suppose we test a single bearing and we keep running the experiment till either the bearing fails or a fixed stopping time a has been reached. Suppose the lifetime of the bearing is distributed according to a Weibull distribution with parameters L_{10} and β . With probability $S(a|L_{10}, \beta)$ the bearing is still functioning at time a . The Fisher information $I_1(a)$ from this experiment can therefore be calculated as:

$$I_1(a) = -S(a|L_{10}, \beta) \frac{\partial^2}{\partial L_{10}^2} \log S(a|L_{10}, \beta) - \int_0^a \left(\frac{\partial^2}{\partial L_{10}^2} \log f(x|L_{10}, \beta) \right) f(x|L_{10}, \beta) dx \tag{10}$$

This can be written as:

$$I_1(a) = \left(\frac{9}{10}\right) \left(\frac{x}{L_{10}}\right)^\beta \frac{\left(\frac{a}{L_{10}}\right)^\beta \beta(1+\beta) \log \frac{10}{9}}{L_{10}^2} - \int_0^a \frac{\beta - \left(\frac{x}{L_{10}}\right)^\beta \beta(1+\beta) \log \frac{10}{9}}{L_{10}^2} \frac{1}{L_{10}} \left(\frac{9}{10}\right) \left(\frac{x}{L_{10}}\right)^\beta \left(\frac{x}{L_{10}}\right)^{\beta-1} \beta \log \left(\frac{10}{9}\right) dx \quad (11)$$

We do not have a closed-form expression for I_1 , but the numerical evaluation of this integral is fast. If n bearings are tested with type I censoring, the Fisher information I_n is

$$I_n = nI_1. \quad (12)$$

There are two important time measures for the duration of the experiment, 1) na , i.e., n times the fixed stopping time a and 2) the total time that machines are running, denoted by $T(L_{10}, \beta, n, a)$. The first measure is relevant to the situation that the machines on which a bearing fails before time a cannot be used for other purposes or if one has to pay for the time use of these machines even if they are not running. The second time measure is important if machines on which a bearing fails can be used for other experiments.

For the first time measure, the Fisher information per time unit equals I_1/a . For the second time measure we need to calculate the expected running time of a single-bearing experiment with type-1-censoring.

$$E[T(L_{10}, \beta, 1, a)] = \int_0^a x f(x|L_{10}, \beta) dx + aS(a|L_{10}, \beta) \quad (13)$$

and the Fisher information per time unit equals $I_1/E[T(L_{10}, \beta, 1, a)]$. For the relevant case that $\beta > 1$, the larger a , the higher the Fisher information per time unit for the second time measure (see Figure 7). For the first time measure, there is an optimum. There are not many failures before time a if a is small. On the other hand, if a is large, many machines are empty because the bearing on that machine has failed already. The optimum as function of β is plotted in Figure 8.

3.2 Type II censoring

The Fisher information I of this experiment with respect to the parameter L_{10} is given by:

$$I = -E \left(\frac{\partial^2}{\partial L_{10}^2} \log (f_{(n,k)}(x_{(1)}, x_{(2)}, \dots, x_{(k)}|L_{10}, \beta)) \right). \quad (14)$$

The second order derivative with respect to L_{10} of the logarithm of the probability density function is given by:

$$\frac{\partial^2}{\partial L_{10}^2} \log (f_{(n,k)}) = \frac{k}{\beta} + \frac{\beta(1+\beta)}{L_{10}^{2+\beta}} \log \left(\frac{10}{9}\right) \left((n-k)x_{(k)}^\beta + \sum_{i=1}^k x_{(i)}^\beta \right). \quad (15)$$

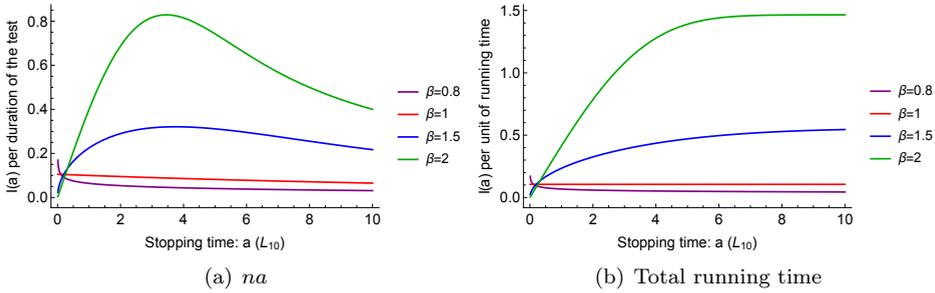


Figure 7: The Fisher information per time unit for type I censoring. In Figure a) the Fisher information is divided by na , in Figure b) the Fisher information is divided by the total running time.

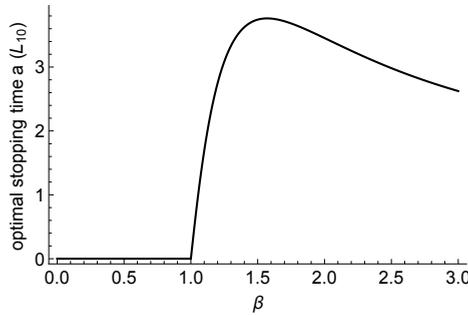


Figure 8: Stopping time for type I censoring that maximizes the Fisher information per time unit ($I_1(a)/a$).

Therefore, the Fisher information is given by:

$$I = - \int_0^\infty dx_{(1)} \int_{x_{(1)}}^\infty dx_{(2)} \dots \int_{x_{(k-1)}}^\infty dx_{(k)} f_{(n,k)} \frac{\partial^2}{\partial L_{10}^2} \log(f_{(n,k)}), \quad (16)$$

Luckily this k -dimensional integral can also be expressed as a double integral (Park (1996)). Let $f_{k:n}$ be the density function of the k^{th} order statistic in a sample of size n . We have the following expression for $f_{k:n}$ (see page 224 of Rinne (2008)):

$$f_{k:n}(x) = \frac{n!}{(k-1)!(n-k)!} F(x)^{(k-1)} S(x)^{(n-k)} f(x) \quad (17)$$

If we define $g(w) := g(L_{10}, \beta, w)$ as

$$g(L_{10}, \beta, w) := \int_w^\infty \left(\frac{\partial}{\partial L_{10}} \log \left(\frac{f(x|L_{10}, \beta)}{S(w|L_{10}, \beta)} \right) \right)^2 \frac{f(x|L_{10}, \beta)}{S(w|L_{10}, \beta)} dx \quad (18)$$

we can write the Fisher Information as:

$$I = n \left(\frac{\beta}{L_{10}} \right)^2 - (n - k) \int_0^\infty g(w) f_{k:n}(w) dw \quad (19)$$

It can be shown, e.g., by performing the substitution $y = \left(\frac{w}{L_{10}} \right)^\beta - \left(\frac{x}{L_{10}} \right)^\beta$, that the function $g(w)$ does not depend on w . We obtain that $g(w) = \left(\frac{\beta}{L_{10}} \right)^2$. Therefore we obtain as result that the Fisher information for Type II censoring has the following form:

$$I = k \left(\frac{\beta}{L_{10}} \right)^2, \quad (20)$$

i.e., each observed failure time provides the same amount of information, independent of the sample size or the order of the failure.

There are again two important time measures for the duration of the experiment, 1) the time till the k^{th} failure, which is given by the k^{th} order statistic $x_{(k)}$ and 2) the total time that machines are running, denoted by $T(L_{10}, \beta, n, k)$.

The total running time is given by:

$$T(L_{10}, \beta, n, k) = \sum_{i=1}^k x_{(i)} + (n - k)x_{(k)} \quad (21)$$

and the expected total running time can be calculated once we know the expected time of the k^{th} order statistic, i.e.,

$$E(T(L_{10}, \beta, n, k)) = \sum_{i=1}^k E(x_{(i)}) + (n - k)E(x_{(k)}) \quad (22)$$

The probability density function of the j^{th} order statistic is given by:

$$f_{j:n}(x) = \frac{n!}{(j-1)!(n-j)!} F(x|L_{10}, \beta)^{(j-1)} (1 - F(x|L_{10}, \beta))^{n-j} f(x|L_{10}, \beta) \quad (23)$$

and the expectation of the j^{th} order statistic is given by (see Formula (5.34) of Rinne (2008)):

$$E(x_{(j)}) = j \binom{n}{j} \frac{\Gamma(1 + \frac{1}{\beta})}{(\log \frac{10}{9})^{\frac{1}{\beta}}} L_{10} \sum_{i=0}^{j-1} \frac{(-1)^i \binom{j-1}{i}}{(n-j+i+1)^{1+\frac{1}{\beta}}}. \quad (24)$$

We propose as a measure to compare several test strategies the Fisher information per time unit, i.e., the Fisher information divided by the expected total running time of the machines, or the Fisher information divided by the time until the k^{th} failure.

If the total running time is relevant, the higher k the more information is obtained (see Figure 9(a)), simply because every failure gives the same amount of information and failures occur more rapidly when the bearing are ageing. If the time until the k^{th} failure is relevant, there is an optimal value of k , as can be seen in Figure 9(b). This optimal value of k can also be interpreted as an optimal value of the ratio k/n and this optimal ratio depends on β (see Figure 10).

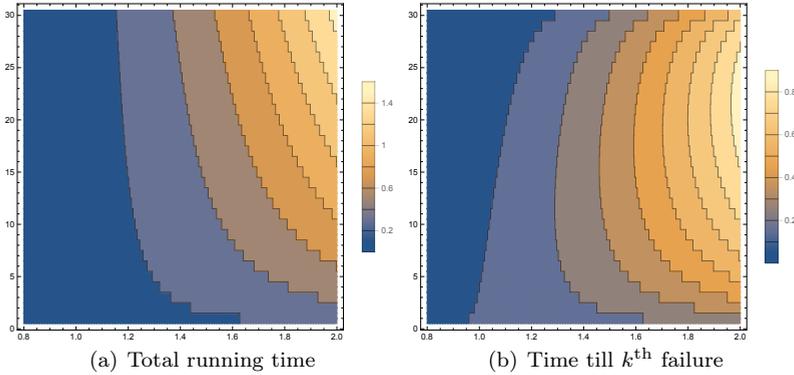


Figure 9: The Fisher information per time unit for Type II censoring with 30 machines. In Figure a) the Fisher information is divided by the total running time, in Figure b) the Fisher information is divided by the time of the k^{th} failure multiplied with the number of machines.

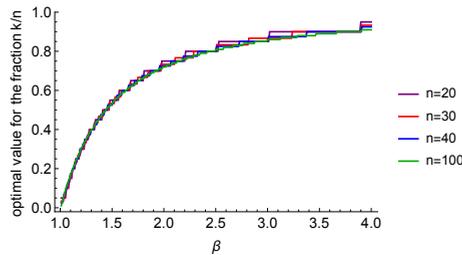


Figure 10: Optimal ratio for the fraction k/n if the time until the k^{th} failure is relevant.

4 Numerical results

In this section we study by simulation different test strategies in the simplified case when there is only one type of bearing and failed items are not replaced. The test strategies all involve hybrid censoring. In order to have practical relevance, all test strategies should meet the condition that $R_{10;0.90}^\beta < 12$. The optimality criterion is to minimize to 80%-percentile of the TTT (Total Time on Test) statistic, i.e. the sum of failure times for the failed items and the testing period for the items that did not fail before the end of the test. The failure times were sampled from a Weibull distribution with $L_{10} = 100$ and shape parameter $\beta = 1.1$. In the hybrid censoring testing strategies we varied the number of items on test between 20 and 30, the number of failed items between 4 and 10 and the maximum testing period between 400 and 600 (with steps of size 25). In order to obtain accurate values, we used 30,000 replications for each setting. The irregular shapes in the contour plots are

interpolation artifacts caused by the integer values for the number of bearings.

4.1 Results Type I Censoring

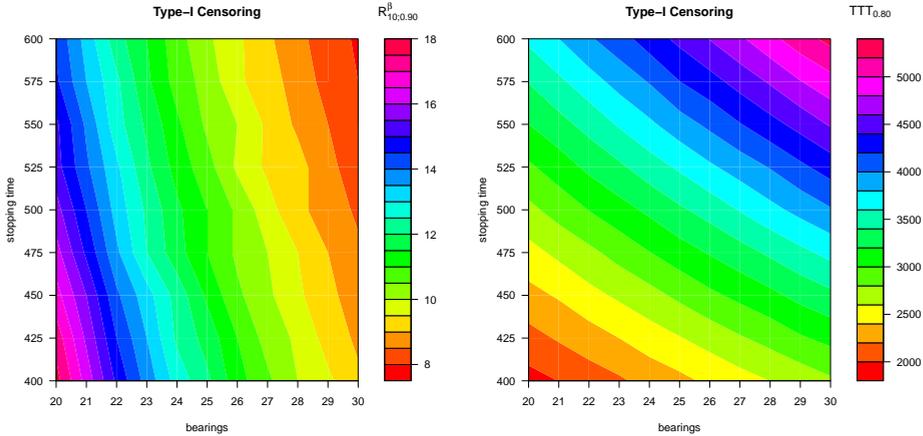


Figure 11: Type I censoring.

bearings	20	21	22	23	24	25	26	27	28	29	30
trunc	850	725	625	550	475	425	375	325	275	275	250

Table 1: Minimum required number of failures under type I censoring.

For type I censoring we see in Figure 11 that the condition $R_{10;0.90}^\beta < 12$ is met by combinations of stopping times and number of bearings that are to the right of the straight line that goes from 23 bearings and stopping time 475 to 25 bearings and stopping time 425. It follows from Table 1 that the required stopping time decreases linearly with the number of bearings (approximately 50 to 75 hours per bearing). It follows from Figure 11 that the optimal choice is to stay exactly on the straight line to obtain a minimal TTT value.

4.2 Results Type II Censoring

For Type II censoring we see artifacts in the plots in spite of the 30,000 replications in the simulation. Therefore we also present the minimum number of failures to ensure $R_{10;0.90}^\beta < 12$ in Table 2. Note the sharp drop when going from 22 to 23 bearings. We note that the TTT value does not change much for a fixed number of failures when we increase the number of bearings. This means that the increase in TTT for a fixed number of bears is fairly constant in the range of bearings that we considered: every extra failure in the stopping criterion causes an increase of 1000 in the TTT value.

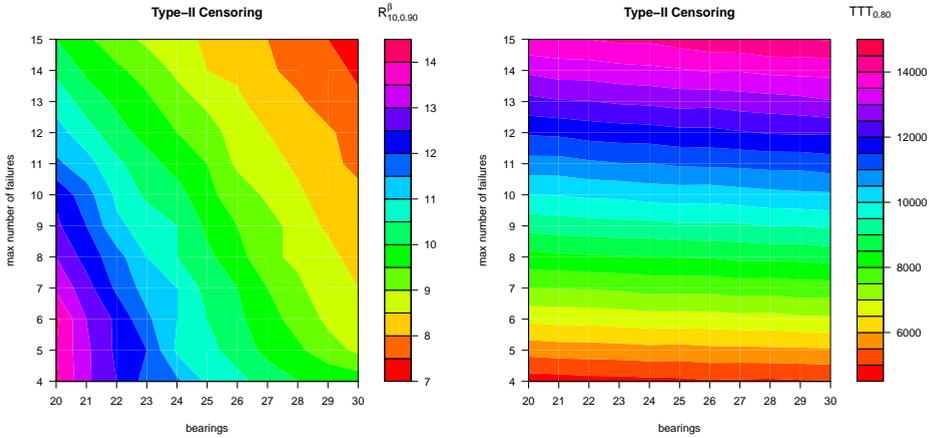


Figure 12: Type II censoring.

bearings	20	21	22	23	24	25	26	27	28	29	30
r	11	9	8	4	4	4	3	3	3	3	3

Table 2: Minimum required number of failures under type II censoring

4.3 Results Hybrid Censoring

As discussed in Subsection 2.2, the idea of hybrid censoring is to have a beforehand fixed maximum testing time (due to the type I censoring mechanism) which is very important from a practical point of view (scheduling of testing facilities), but at the same time have the option to stop the testing earlier if the test results allow sufficiently precise estimates (due to the type II censoring mechanism). However, this is more complicated than it looks at first sight. For example with type I censoring and 20 bearings we need a stopping time of 850. If we add any corresponding type II censoring stopping criterion, then this means that the value of $R_{10;0.90}^{\beta}$ will increase since we may stop too early. For example, if we perform hybrid censoring by naively combining the type I stopping time of 850 with the type II criterion for 20 bearings (i.e., stop after 11 failures), then $R_{10;0.90}^{\beta} = 12.1$. In order to meet the $R_{10;0.90}^{\beta} < 12$ condition we need to increase either the type I or the type II criterion. For 20 bearings, this means we could choose the stopping time to be equal to 850 and number of failures to be equal to 12 (with TTT approximately 5900) or choose stopping time 875 and number of failures 11 (with TTT approximately 6150). It is thus better to fix the type I criterion and increase the type II criterion when applying hybrid censoring.

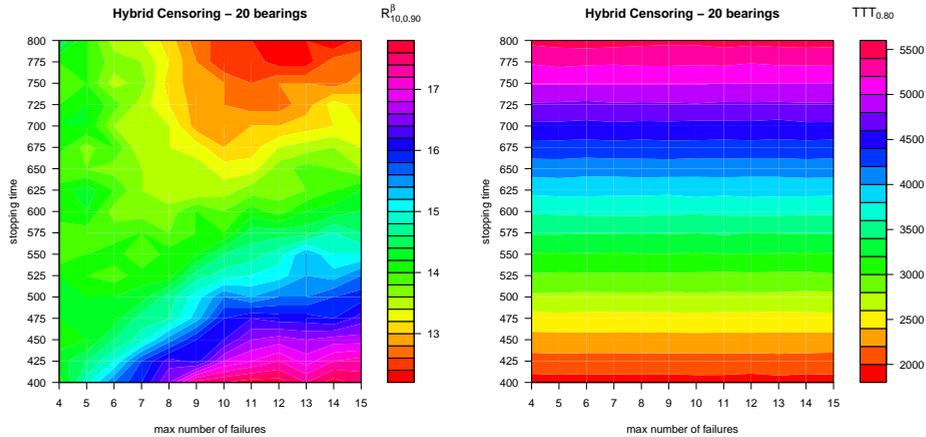


Figure 13: Hybrid censoring - 20 bearings.

5 Discussion

In this section we first summarize the conclusions that we may draw from our results. We then present recommendations to SKF for further lines of research.

5.1 Conclusions

In this paper we studied a simplified case of the testing problem posed by SKF. The simplification consisted of considering only one type of bearing and no replacement of failed items. We approached the problem through both a theoretical approach based on Fisher information and a numerical approach based on simulation.

A key insight of the approach based on Fisher information is to continue testing as long as there is a positive rate of contributing information. Information per test time leads to optimal values for the stopping time (Type I censoring) or the number of failures (Type II censoring) expressed as a function of β .

Simulations help to minimize tests time along the Pareto front of strategies. For Type I censoring there seems to be a linear dependence of R_{10}^β on the truncation time. For Type II censoring we see drastic changes for smaller values of bearings in the number of required failures in order to satisfy a constraint on R_{10}^β . The corresponding values for the TTT, however, seem to be fairly constant as a function of the number of bearings. For hybrid censoring one cannot simply combine the criteria of Type I and Type II censoring. In order to get optimal TTT values, one should fix the Type I criterion of a given number of bearings and increase the Type II criterion.

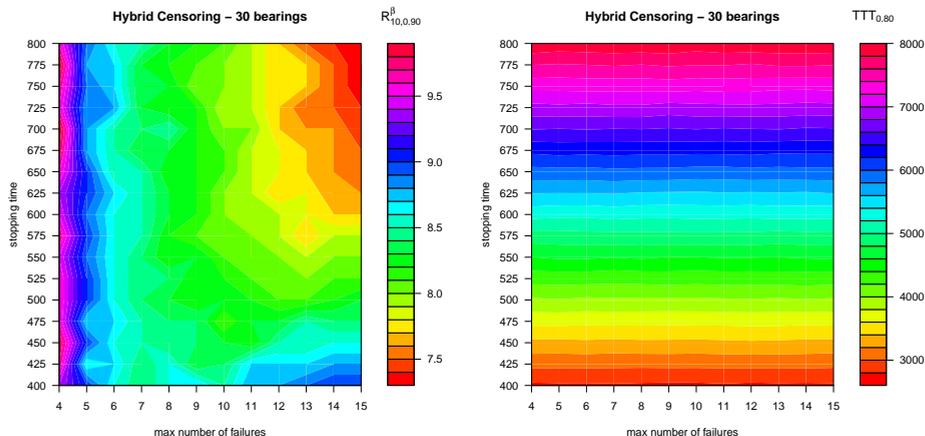


Figure 14: Hybrid censoring - 30 bearings.

5.2 Recommendations for Future Research

Future research is needed to extend the results of the current paper to more realistic situations with different types of bearings and more complex replacement strategies (not only replacement of failed items individually, but also in pairs as testing devices usually combines bearings in groups of 2 or 4 bearings).

For the simulations we recommend to use larger simulations or develop variance reduction techniques like importance sampling in order to obtain more stable results. Since the simulations were performed for only one set of values of the Weibull parameters L_{10} and β , it is recommended to study the influence of L_{10} and β .

The approach based on Fisher information should be extended to include tests with more than one type of testing as well as hybrid testing. It is also recommended to perform a sensitivity analysis.

A final idea is to explore the idea of approximation the Weibull distribution with an exponential distribution when $\beta \approx 1$.

Acknowledgements

We thank SKF for supplying us with the necessary background information for this interesting problem. S. Blachère wishes to thank Mr. Alexander de Vries, Director SKF Group Product Development, for his kind permission to publish this article.

References

L. Bain and M. Engelhardt. *Statistical Analysis of Reliability and Life Testing Models - Theory and Methods*. Marcel Dekker, New York, 2nd edition, 1991.

- N. Balakrishnan and D. Kundu. Hybrid censoring: Models, inferential results and applications. *Comput. Stat. Data An.*, 57:166–209, 2013.
- A. Childs, B. Chandrasekar, N. Balakrishnan, and D. Kundu. Exact likelihood inference based on Type-I and Type-II hybrid censored samples from the exponential distribution. *Ann. Inst. Stat. Math.*, 55:319–330, 2003.
- A. Cohen. Maximum Likelihood estimation in the Weibull distribution based on complete and on censored samples. *Technometrics*, 7(4):579–588, 1965.
- B. Epstein. Truncated life tests in the exponential case. *Ann. Math. Stat.*, 25:555–564, 1954.
- N. Farnum and P. Booth. Uniqueness of Maximum Likelihood estimators of the 2-parameter Weibull distribution. *IEEE Trans. Rel.*, 46(4):523–525, 1997.
- K. Geddes, M. Glasser, R. Moore, and T. Scott. Evaluation of classes of definite integrals involving elementary functions via differentiation of special functions. *Applicable Algebra in Engineering, Communication and Computing*, 1(2):149–165, 1990.
- W. Kahle. Estimation of the parameters of the Weibull distribution for censored samples. *Metrika*, 44:27–40, 1996.
- B. Kundu. On hybrid censored Weibull distribution. *J. Stat. Plann. Inf.*, 137:2127–2142, 2007.
- J. McCool. Inference on Weibull percentiles and shape parameter from Maximum Likelihood estimates. *IEEE Trans. Rel.*, R-19(1):2–9, 1970.
- W. Meeker and L. Escobar. *Statistical Methods for Reliability Data*. Wiley, New York, 1998.
- W. Meeker and W. Nelson. Weibull percentile estimates and confidence limits from singly censored data by Maximum Likelihood. *IEEE Trans. Rel.*, R-25(1):20–24, 1976.
- H. Ng, P. Chan, and N. Balakrishnan. Optimal progressive censoring plans for the Weibull distribution. *Technometrics*, 46(4):470–481, 2004.
- S. Park. Fisher information in order statistics. *J. Amer. Statist. Assoc.*, 91(433):385–390, 1996.
- M. Pike. A suggested method of analysis for a certain class of experiments in carcinogenesis. *Biometrics*, 22:142–161, 1966.
- H. Rinne. *The Weibull Distribution: A Handbook*. CRC Press, 2008.
- W. Weibull. Statistical theory of the strength of materials. *Ing. Vetenskaps Akad. Handl.*, 151:1–47, 1939.

- W. Weibull. A statistical distribution function of wide applicability. *J. Appl. Mech.*, 18(3):293–297, 1951.
- P. Zehna. Invariance of Maximum Likelihood estimators. *Ann. Math. Statist.*, 37(3): 744–744, 1966.

