Railway - Spare Inventory & Maintenance Scheduling model development

D. Sommers

Delft University of Technology

TU Delft

Railway - Spare Inventory & Maintenance Scheduling model development

for railway maintenance management

by

D. Sommers

Student number: 4858530

A thesis to fulfill the degree of

Master of Science
in Transport, Infrastructure & Logistics

at the Delft University of Technology,
to be defended publicly on Friday July 23, 2021 at 16:00.

Supervisors:	Prof. dr. R.M.P. Goverde,	TU Delft CEG
	Dr. N. Bešinović,	TU Delft CEG
	Dr. S. Fazi,	TU Delft TPM
	Drs. M. Grashoff,	ProRail

Preface

This report is the second-to-final achievement in a 9+ year educational track.

Over those years there were quite a lot of trials and tribulations which caused me to switch gears a few times. Starting out in mechanical engineering in Eindhoven, switching to another institution, switching back to the TU/e to continue in industrial engineering; switching to Delft to start a degree in transport, infrastructure & logistics (which is hereby completed) and finally also taking on another degree in science communication. I'm not listing this in order to boast about the wealth of knowledge I may have gathered through all of this (although I am very grateful for that as well) but to illustrate that there is not one single path which brings everybody straight from A to B (nice inclusion of a transportation metaphor if I do say so myself). Instead that path is very badly visible while you're on it, sometimes causing doubts about whether you've completely lost your way all together. Looking back you can see more clearly what you did and why your path followed the curves it has.

The perseverance I needed to follow that path towards this point obviously was not fully (if at all) my own, it has required many people a lot of time and effort to push and drag me all the way to the end of this masters degree. Specifically during this research, that help came from my committee. First of all obviously Rob as committee chair to guide me in finding a research direction. Thank you for nudging me and pretending that I found the answer to your questions all by myself. Also Nikola, with his unlimited enthusiasm for research and deep understanding of the craft. I hope to some day find a job in which I am half as happy as you are in yours. Your upbeat attitude really helped me through some pretty low points in this graduation process. I also want to thank Stefano for his contribution specifically to the modelling and the encouragement to just give it a try. Diving in the deep end is quite scary and you knew exactly when to give me that little push off the diving board. Finally Mark, I want to thank you for your kindness throughout this process. At some points students have an irrational tendency to feel like the whole world is against them, but you were always the one in my corner.

While the TU Delft is a great institution and ProRail a great company to graduate in, my personal environment is probably even more important in aiding me to reach this milestone. I want to thank my parents first and foremost, for supporting me throughout my 24-year spanning education and always encouraging me to take full advantage of my potential. Also my big brother and sister, who were a massive inspiration when I was little and they were big, and still are today. I also quickly want to shout-out the Paelmannen from Eindhoven for teaching me about the finer things in life outside of my academic achievements, as well as the friends I made in Delft who made me feel at home and taught me the Delft way of life. And of course my fantastic partner Suzanne, I am convinced that without you I'd most likely live under a bridge somewhere and sell pretty rocks I found for a living. You give me purpose in life and are an incredible example to live up to.

While there is still a big challenge ahead before I'm officially done being a student, this thesis marks the point I can at least start to call myself an Engineer and I can not find the words to express how proud I am of that.

Daniël Sommers
Delft, July 2021

Summary

Planning of maintenance operations in large systems like a railway network becomes is non-trivial.

To aid in governing the maintenance process, mathematical models have proven very effective. Statistical methodologies have been developed and applied to analyse failure of components in technical systems but also to forecast other stochastic processes like lead times of spare parts or the success of a repair action. Most of these models however favour industrial settings in which products are produced or on the other end of the spectrum fully disconnected systems like fleets of trucks.

The research before you adds to the scientific body of modelling maintenance processes by expanding it towards the railway infrastructure domain. This specificity makes the problem at hand differ from the traditions of production line analysis because of less dependence of failure dynamics between parts, moreover the addition of minimal maintenance options in the model are not a custom in the traditional models. A Monte Carlo simulation is developed which estimates costs made for maintenance operations in a railway system. This simulation is subsequently included in a modelling framework where surrogate modelling is used to optimise three decisions: when to replace parts based on their condition, when to buy spare parts and how many spare parts to buy each time.

To this end a model should be able to take in data regarding a specific part (i.e. a specific type of railway switch) which is installed multiple times in the entire system, and produce optimised values for the three decision variables. The research is governed by a single research question:

How can spare inventory control decisions and maintenance scheduling decisions be modelled for the application in railway networks to minimise operational cost?

To simulate both spare inventory control and maintenance scheduling, overarching strategies are chosen to fit the problem situation. For the sake of maintenance scheduling, a highly advanced level of scheduling should be adhered to. The railway network after all has a high demand of reliability. The most advanced methods of maintenance scheduling strategies are the *condition based maintenance* (CBM) strategies. In which parts are preventively replaced whenever possible. This involves checking the parts' condition and replacing that part when its deterioration exceeds a threshold L_p which is based on knowledge about its failure processes (preventive in-place repair is not considered as an option as a result of the reality of the railway network operations). The inspections of the condition of a part are in this research considered to happen periodically, while some parts in the railway system do have sensor technology allowing for continuous data on their condition, these are very rare and have less need for the developed model.

Spare inventory control can be steered through strategies that specify how the stock levels are reviewed as well as how to buy new parts. Because the warehousing of spare parts in these large systems is often outsourced to specialised third-party operators, continuous review of stock levels is the most applicable choice. The ordering regime can then take the form of an *(s,nQ)* system where n units of quantity Q are ordered each time the stock level falls below s such that the stock level is restored to above s. Alternatively the *(s,S)* system can be used, where every time the stock level drops below s enough spare parts are ordered to replenish the stock level back to S. For the sake of this research only the more generalised form of *(s,S)* is considered.

To fulfill the research goal, the developed model should thus optimise L_p, s & S to suit the minimisation of the operational cost. The final model consists of two parts: a Monte Carlo type simulation which predicts an estimated cost based on part data and values for the three decision variables and an optimisation algorithm which optimises the three decision variable values through surrogate modelling. Together these form the modelling framework (see figure 1) of the functional Railway - Spare Inventory & Maintenance Scheduling (R-SIMS) model.

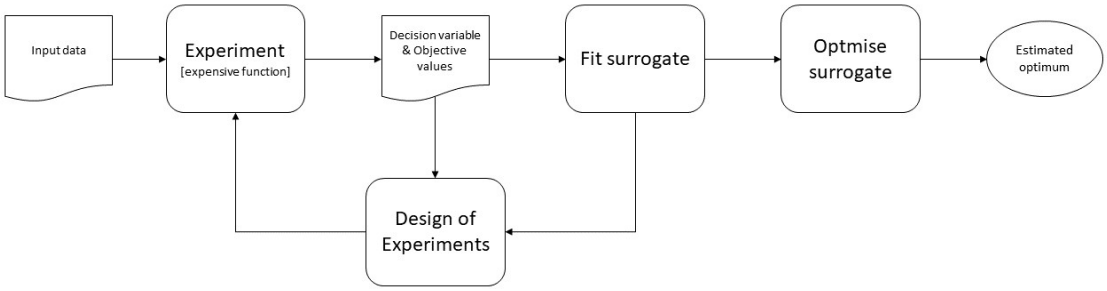

Figure 1: modelling framework of the R-SIMS model

The simulation was developed using a base model from literature. Other than general alterations of the base simulation which increased accuracy and applicability in railways, the new simulation was equipped with possibilities for minimal maintenance in a case of stock-out after an unexpected failure of a part.

To optimise the decision variables, a suitable optimisation technique is found in surrogate modelling, a discipline of optimisation which deals specifically with derivative free optimisation (DFO) of expensive black-box functions (functions which have no closed form objective function available like the simulation in this case). Four surrogate modelling algorithms are identified as potential fits for the problem at hand. These algorithms are focused on problems with a single objective and integer decision variables. Moreover they are able to deal with problems which would have a solution close to the edges of a search space.

To ascertain which of the four candidate surrogate modeling algorithms, experiments are performed to test each on the measures of performance, speed and stability. Performance being the quality of the solution found by the algorithm in a certain amount of time, speed being defined by the time per iteration as well as the time required by an algorithm to reach a certain objective value and stability being the variations of results found by a single algorithm when running the same experiment for the same duration multiple times. The propagation of the best found objective over the 200 iterations performed in this experiment are shown in figure 2.

Results show that for performance, the MVRSM and HyperOpt algorithm score equally well (average objective values found of 37.42 and 38.20 €/part/period respectively and a pair-wise t-test significance of 0.6986 over the results of both algorithms). On the speed metric, HyperOpt and Randomsearch show the lowest time per iteration with averages of 10.72 and 10.87 seconds per iteration respectively. Moreover HyperOpt requires the least amount of time to reach the objective value goal of 64.32 €/part/period with 2.68 minutes required on average. In terms of stability HyperOpt also out-performs MVRSM both in case of the variation in the found objective (coefficient of variation of the objective of 0.07 for HyperOpt and 0.14 for MVRSM) as well as for the found values of the decision variables (combined coefficient of variation of 1.55 for HyperOpt versus 3.63 for MVRSM). This leads to HyperOpt being the most favourable algorithm for optimisation of the studied problem.

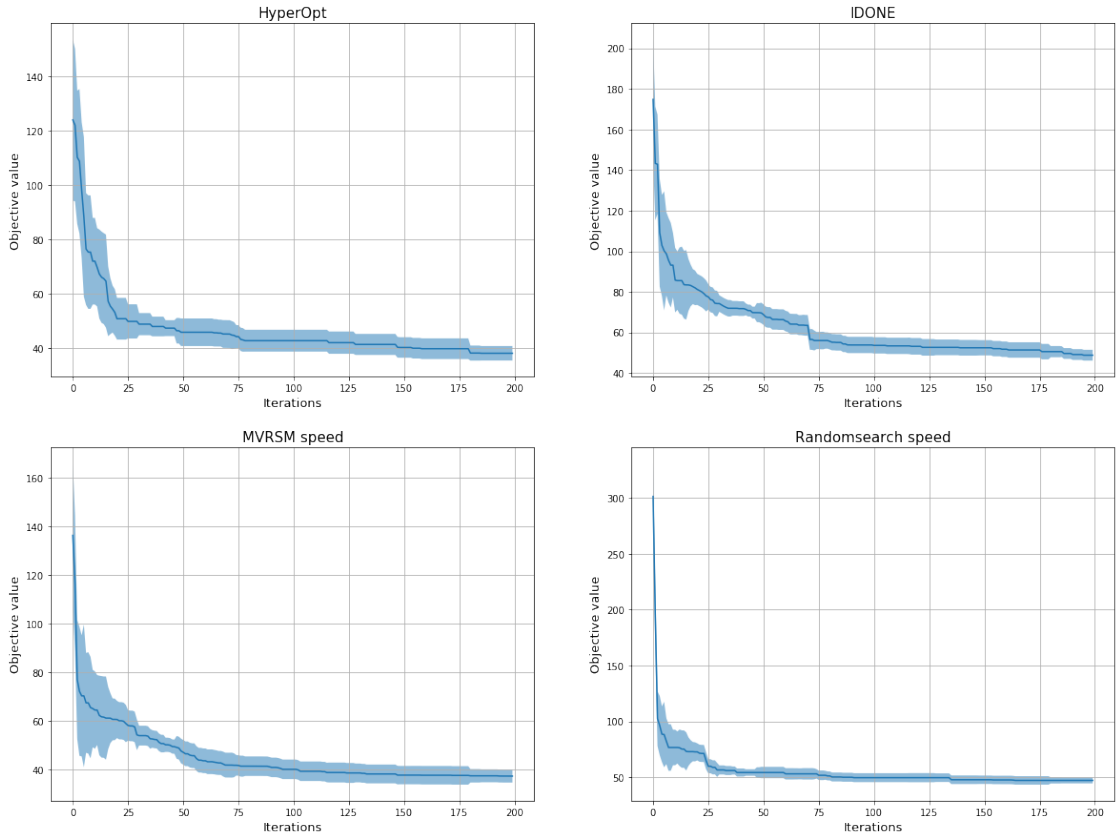

Figure 2: Performance runs averaged over the 10 repetitions of the experiment with a shaded area indicating one standard deviation of the results above and below the average.

A test-case is subsequently ran through the entire model to give a practical example of its overall functioning. The 200 trials of the model resulting from this test-case are visualised in figure **??**. Where the test-case shows that the full model is operational and yields useful results, some caveats regarding its use can be described. The downtime calculation currently happens in a manner which is not in line with reality. Because of the time granularity of the simulation, downtime is only recorded in full time-steps. When a user then uses a larger time-step (i.e. a month or year) to reduce computational power necessary, downtime becomes increasingly unrealistic. To combat this problem another method of determining downtime could be used (possibly estimating a probability distribution for the downtime length within a single time-step). Another problem which relates to the previous one is the computational effort involved in solving the model. The use of high-end computation systems would help to solve this but the inherent evaluation of the problem by the model resists parallelisation of the calculations. This is a result of the sequantiality of the problem, each time-step has to be processed after the other because decisions made in one time-step can influence stock levels and possibilities for maintenance actions in the next. Moreover during each time-step each part installed in the system has to be handled after the other as well because of the same reason. Finally the data gathering necessary to use the model is quite significant. A lot of data on operational cost of for instance replacement of parts or inspection are required as inputs for the model and these costs often have complex underlying financial and operational foundations. If a user does not poses data sets which fit the required data fully, they will have to spend much effort on gathering the right data in the right fashion before they would be able to use the model fully.

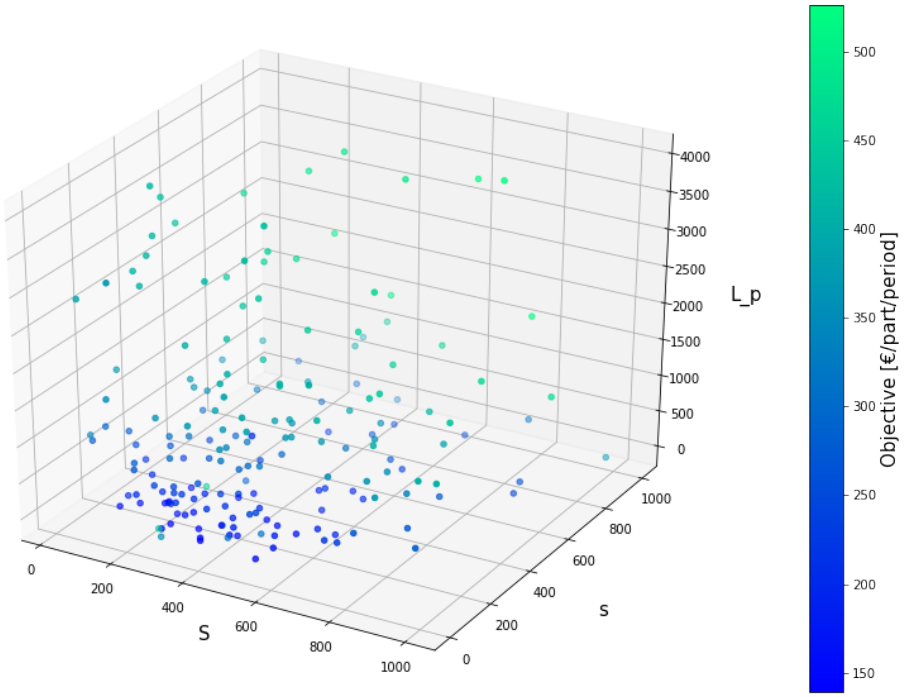

Figure 3: 4D plot of objective and decision variables found by R-SIMS model

To gain a deeper understanding of the newly developed simulation model, a sensitivity analysis was performed studying the effect of nine of the used parameters on the objective *EC*, the operational cost of a part per period. This analysis showed a large influence of the downtime cost (average correlation = 0.999), the spare part procurement cost (average correlation = 0.999) and the holding cost (average correlation = 1.000) on the final objective. The lead time shows a significant correlation (0.863) with the objective only in a low demand setting and inspection interval has a strong significant correlation (0.970) with the objective in the medium demand setting.

The developed model satisfies the research goal, it optimises estimated costs per part per period and returns values for the three decision variables which govern both spare inventory control (reorder level s and order up-to level S) and maintenance scheduling (CBM threshold L_p). By choosing proper elements to build the simulation (e.g. the CBM and (s,S) policy and adding minimal maintenance) this model better suits the railway infrastructure context than the models available in literature.

Future research on this topic could focus on:

More research could be done to gain insights into best practices when it comes to data selection. The data required to utilise operational models is often very detailed and widespread throughout an organisation or over stakeholders. The aggregation of data sources and control of access is an entire field of data science which would be applicable to this research. Secondly, alterations to allow the model to include parts which have continuous monitoring capabilities would be helpful. Because large systems like the railway infrastructure often involve different sets of parts. Some of these parts in practice have their own dedicated continuous monitoring systems which allow the owners to implement another maintenance scheduling strategy. Making the model more modular in such a way that the user could run these parts through as well would be a great increase in value for the model. Thirdly, including parts which have an *(s,nQ)* inventory control strategy. Similarly to

the previous point, some part sets have a different inventory control strategy from the one used in the R-SIMS model. By again making the model more modular to accommodate for these strategies it would be more widely applicable in practice. Fourth, better methods for estimating downtime costs should be developed. In the developed model, because of the granularity of time it is very difficult to balance computational needs with an accurate estimate of downtime cost. Especially in infrastructure systems like the railways where downtime is extremely costly this factor is important. Further work on the implementation of the downtime costs would therefore be a great contribution to the R-SIMS model in particular. Finally, possible expansion of the application outside the railway management domain. While one of the biggest contributions of this research is the development of a combined spare inventory & maintenance scheduling model specifically suited for railway infrastructure, the literature review showed that in general these models rarely include functionalities for minimal maintenance. This functionality is particularly valuable in the railway setting but could also be applied in specified other applications which for instance are hard to reach with spare parts. Research in this wider application would therefore increase the benefit industry would be able to gain through the use of the R-SIMS model.

Contents

List of Figures

List of Tables

1 Introduction

When technical systems like factories, coal mines or a railway network increase in size, maintenance increases in both complexity and importance exponentially. One of the challenges of maintaining large technical systems is getting a grip on the failure dynamics of parts. Failure can be linked to many factors; environmental factors like weather, differences in how users treat objects and unexpected accidents can all influence the timing of part failure, this makes failure very hard to predict. Nonetheless, statistic analysis can be used to determine the best courses of action when maintaining technical systems and installations. The planning of repair operations as well as the consideration of procurement of spare parts are important functions for a system maintenance manager. Where smaller systems like a farmer's tractor can be managed on a more ad-hoc basis, larger ones like a railway network require standardisation and a logical reasoning for the choices made. Typical decisions that have to be made in maintenance management are: when to repair or replace parts in the system, or how many spare parts to keep. These decisions are usually aided by mathematical modelling and statistics. While modelling these decisions separately can be appealing from the modellers' perspective because of its simplicity, an integrated approach in which both decisions regarding the actual maintenance scheduling strategy and spare inventory control strategy are modelled generates more value to the user. This research therefore aims to develop such a modelling framework which is specifically tailored towards the railway infrastructure domain.

1.1. Problem at hand

Maintenance scheduling and spare inventory control are usually managed using defined strategies. These strategies dictate what actions to take at what point in the maintenance and warehousing cycles. A maintenance strategy could contain rules which specify when to replace parts or when to inspect them. A spare inventory control strategy could contain rules regarding the number of spares to keep, at which points to check the number of spares left and how many spares to buy in each order.

Many of the models made to aid in decision making for maintenance operations have mainly been focused on applications in production operations or single machines. However, for large and interconnected systems like the railway network operational decisions in maintenance work differently. This is a result of particularities in larger systems that smaller systems do not share: firstly the inter-dependence of parts in the system is less influential in larger systems like railways. When in a production line one part breaks, other parts will quickly be unable to function because the entire system is only operating on a single product. Therefore, most decision making models in maintenance management consider parts to be dependent on each other (Van Horenbeek &

Pintelon, 2015; J. Wang & Zhu, 2021; Zhang, Liao, Zeng, Shi, & Zhao, 2021). In larger systems this problem is less prevalent. A railway network handles many products at the same time and when single parts break there is a lot of opportunity to circumvent the failure.

Secondly smaller systems contain very particular parts which are made with a high degree of precision. This leaves little room for improvisation when parts have to be replaced but no proper spare parts are around. For instance, when this happens with a car engine, often the best option is to refrain from using that engine until the proper spare parts are available. However, when a support beam for overhead lines in railways is starting to fail, one could still try to install part of an old (possibly decommissioned) beam or try welding the cracks shut if they do not have the proper spare part available. This process is called minimal maintenance and is rarely included in decision support models for maintenance management (Keizer, Teunter, & Veldman, 2017; L. Wang, Chu, & Mao, 2009; J. Wang & Zhu, 2021). An extensive review study by Van Horenbeek, Buré, Cattrysse, Pintelon, and Vansteenwegen (2013) found less than 10% of models reviewed considered any type of maintenance other than perfect repair/replacement.

The modelling framework which is developed in this thesis will address only single types of parts at a time. Because of unmanageable complexity in the case of multiple part-types being modelled simultaneously this is common in spare inventory control & maintenance scheduling models.

Addressing a single type of part at a time means that the model is able to make the studied decisions for a collection of identical parts which are installed in the system. If the system in this case would be represented by a chair, a single part type could be a chair leg. There are four legs in the system but because they are all identical they can share a pool of spares. This makes the legs a single part type. The model would then be able to determine for instance the number of spare legs to warehouse or the moment when new spare legs should be procured. The model could do the same for the two arm rests or the back rest but not for all at once since these are different part types. When translated to the railway infrastructure domain one could regard a specific type of railway switch as a part type of which many could be installed in the system. This method reduces the necessary computing power as well as the detailed data which the user is required to provide.

To fill the need for these types of models which are built specifically for the railway domain, this research develops such a modelling framework and uses it to address several relevant decisions in maintenance management of railway networks.

The main contribution of this research is the development of a Monte Carlo simulation which is capable of estimating operational cost as a result of the maintenance decisions taken. Moreover this developed simulation is paired with an existing surrogate modelling algorithm to minimise the estimated costs by finding the best values for three decision variables. The simulation and the surrogate modelling algorithm together form the developed modelling framework.

An example of a use case of this model would be that of a specific type of railway switch. One could consider the Dutch railway network in which this type of switch would be installed 200 times, spread over the system. A user of the developed model would then be able to enter data regarding that type of switch; cost of procurement of the part, cost of replacement operations but also data on its failure frequencies and probability of downtime when the part fails unexpectedly. The model would then process this data and suggest the estimated financially best values for decisions made in maintenance management of that type of railway switches.

Another example of an industrial need for such a model is presented by the case of ProRail. ProRail is the Dutch infrastructure manager for all rail infrastructure and the commissioner for this thesis. ProRail is also responsible for the maintenance of all rail infrastructure in the Netherlands. These maintenance operations, including spare parts inventory, are outsourced to expert

contractors. This reduces the understanding ProRail has of the maintenance processes and those processes' influence on the service ProRail is supposed to provide. By means of the proposed model ProRail could get an indication of what the best values are regarding the three studied decisions. This could then serve as an indication of their contractors' behaviours in the maintenance process and subsequently be used to evaluate the relationship with- and division of responsibilities between ProRail and its contractors. Alternatively the developed model could be used by the contractors directly to make operational decisions in their maintenance processes.

1.2. Goal

The goal of this thesis is to develop a model which aids in making better decisions in terms of financial returns regarding spare inventory control and maintenance scheduling. This model is specifically suited for application in railway systems. The parts which are handled by the model should be seen as independent, meaning that they do not influence each others' failure behaviours. The model takes into account a wide array of costs for maintenance operations as well as spare part warehousing and procurement. The model also includes options for minimal repair.

This goal is translated into the following research question and sub-questions:

RQ: *How can spare inventory control decisions and maintenance scheduling decisions be modelled for the application in railway networks to minimise operational cost?*

1. SQ1: *What strategy of maintenance scheduling best suits the railway domain?*
2. SQ2: *What strategy of spare parts inventory control best suits the railway domain?*
3. SQ3: *How can costs of spare inventory and maintenance scheduling be modelled?*
4. SQ4: *How could decision variables in the spare inventory control strategy and maintenance scheduling strategy be determined to minimise operational cost?*
5. SQ5: *How could this type of model add value to railway infrastructure management?*

1.3. Research outline

To provide a theoretical basis and to develop a starting point for the modelling, a literature review is performed in section 2. This literature review starts off with describing basic concepts surrounding maintenance scheduling and inventory control. Further it describes the status quo in literature regarding the combined modelling of spare inventory and maintenance scheduling. Thereafter it shortly introduces methods of optimisation of such models and surrogate modelling in particular. This results the specification of the scientific gap this research aims to fill in an overview of the most recent models and an estimation of the fit of those models to the railway management context.

Section 3 presents a detailed overview of the complete problem including all relevant concepts and their interactions. Proper maintenance scheduling and inventory control strategies will also be presented to fit this problem.

The developed predictive simulation model is presented in section 4. The model uses that simulation to analyse entered scenarios and predict the estimated cost. A form of surrogate modelling is used to optimise the three decisions (when to replace parts based on their condition, when to buy spare parts and how many spare parts to buy) to find the lowest cost prediction. Figure 1.1 shows a schematic overview of the elements of this research and their interactions.

Finally in section 5 a number of numerical experiments are presented to test the selection of surrogate modelling algorithms and to motivate a definitive choice for the best fitting one. This forms a feedback loop towards the development of the model as the experiments determine the optimisation algorithm used in said model. Moreover some sensitivity analysis is preformed on the

simulation to provide important feedback on the influence of several parameters in the result of the simulation. Finally section 7 describes conclusions in terms of answering the research questions and provides insights regarding the added value of this model to both science and industry.

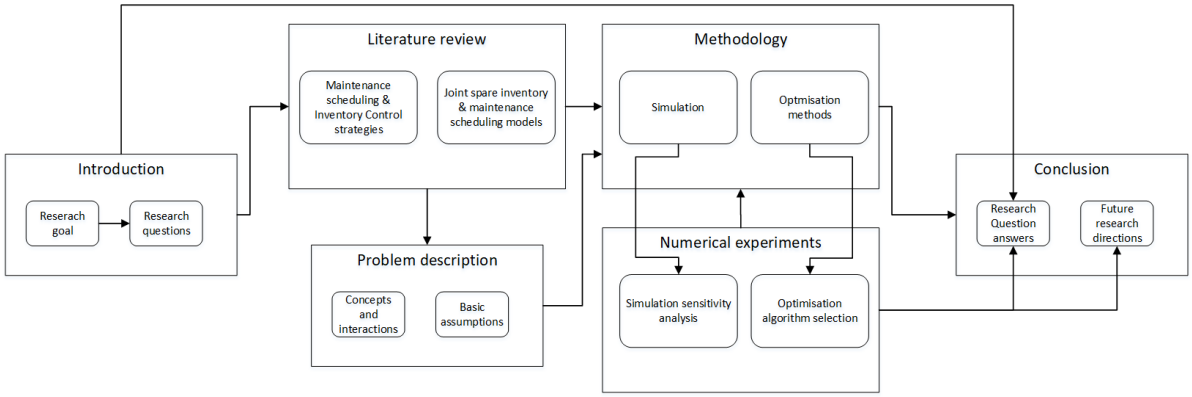

Figure 1.1: Outline of the research

2

Literature Review

This research aims to build a model that aids decision making regarding maintenance scheduling and spare part inventory control in a railway infrastructure setting. The development of this model involves the conceptual and mathematical modelling of business processes (i.e. the proper maintenance and spare inventory control strategies), the consequences of the decisions taken and theoretical optimisation of costs. These models vary widely in their approaches and techniques. The objective of most of these types of models is optimisation of values for inspection- or order-intervals, order quantities and replacement conditions. All to serve the lowest possible operational cost of a system. This chapter first outlines some background on maintenance scheduling strategies and inventory control strategies. Thereafter joint spare inventory and maintenance scheduling models from literature are discussed and a review of optimisation in general and surrogate modelling in particular is provided. This literature review results in a defined scientific gap which the research aims to fill.

2.1. Background

2.1.1. Maintenance scheduling strategies

When looking at maintenance operations and scheduling thereof, multiple strategies are used in practice. Two main types of maintenance strategies are recognised: *Reactive* and *Proactive*. Reactive strategies, also named *corrective maintenance*, describe the best ways to react to a failure, action is only taken after a part or machine has already failed. Proactive maintenance strategies on the other hand, include maintenance actions being undertaken before a part or machine has failed. Proactive strategies can be dissected further into *Preventive* and *Predictive maintenance*. The former is mainly schedule based. Here one would, often based on historical data, calculate the optimal periodical time to service or replace the part in question. The most advanced methods of maintenance are the *Predictive* or *Condition based maintenance* strategies. These strategies take the condition of the part into account when calculating the optimal service or replacement moment. To determine the condition of the part one has to monitor it, either continuously or periodically. Continuous monitoring, if not done digitally, is extremely laborious and even if it is assisted by sensor technology can still come with high costs investment. Condition Based Maintenance with Periodical inspection is already more manageable, but is accompanied by more uncertainty and may require more corrective maintenance than a continuous monitoring policy would. Figure 2.1 shows the structure of this maintenance strategy typology as proposed by Huang, Chen, Chen, and Jiang (2012).

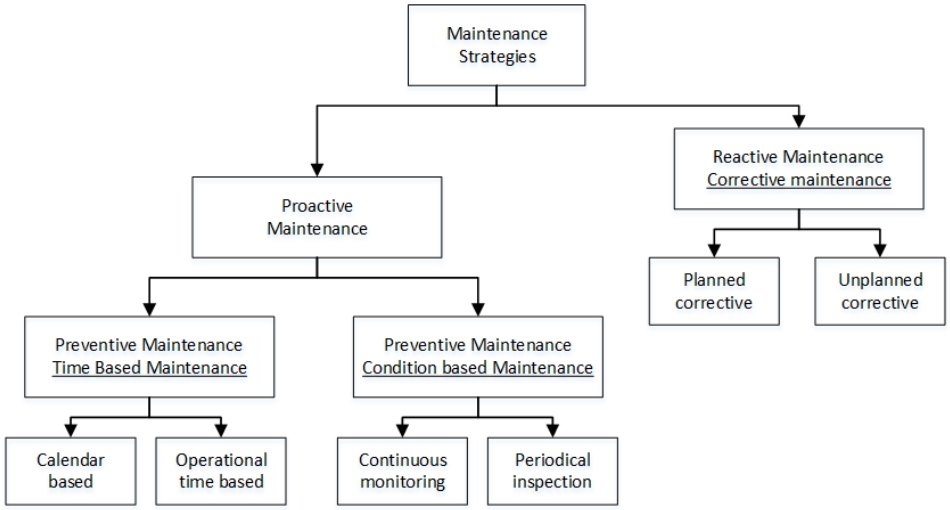

Figure 2.1: Representation of different maintenance strategies (Huang et al., 2012)

From all strategies displayed in figure 2.1, the most advanced is *Condition Based Maintenance* (CBM) with *Continuous monitoring*. This strategy yields the highest accuracy and the lowest chance of unpredicted corrective action being necessary. It is also the strategy that includes the highest investment in monitoring. While better methods are being developed for dealing with the extreme data production that results from continuous monitoring, many large systems in practice are still largely governed by periodical monitoring strategies (Jamshidi et al., 2018).

2.1.2. Spare inventory control strategies

To carry out maintenance operations often spare parts are necessary to replace parts when they have failed or are deteriorating. A *part* is considered a very broad term. What constitutes a part can range from nuts and bolts to larger components like a fuse box or entire installations like complex railroad switches. In order to replace these parts in the system when they have failed or are deemed unfit to function, spare parts are necessary. Without spare parts failing parts in the system could lead to costly downtime. The control of this spare part inventory describes the number of spare parts to keep, when to buy new ones, the number of new spare parts to buy and when to check the number of parts left in stock. Strategies for spare inventory control are highly influenced by the surrounding environment. For example, the packaging in which the parts are delivered as well as the lead times, the review times and the amount of warehousing space available. All these can be of impact on the spare stocking strategy.

Two important concepts in stock keeping are 1) the stock level, also called the Inventory on Hand (IOH) and 2) the Inventory Position (IP). The IOH represents the actual physical inventory of spare parts that are available at a certain moment. This is the quantity that can be checked in the warehouse. The IP is the total number of spare parts in possession. This includes ordered parts as well. For example, if a lead time (time between ordering a part and receiving it) is four days, the IP will be updated as soon as an order is placed while the IOH will only be updated once the ordered parts actually arrive four days later.

Inventory control strategies are described in many different ways in literature. Based on Silver, Pyke, and Peterson (1998) this research uses the following categorisation of inventory control strategies: *(R,s,nQ)*, *(R,s,S)*, *(s,S)* and *(s,nQ)*. The relationship of these strategies is depicted in figure 2.2. In this notation the R stands for a review cycle, an R of 4 for instance would indicate that every 4th period (month/week/day) the IOH is checked to see how many spares are left. The s is the reorder level, this level determines when new spares are ordered. Once the IOH becomes equal to- or lower

than s, new spares should be ordered. S and nQ then determine the number of spares that are ordered. The S this is the order up-to level, when ordering one would always order the number of spares which restores the IP to that level. S is therefore also often called the max stock level. The nQ refers to the batch-size. Part manufacturers often package their products in multiples making it hard or impossible to buy single pieces. In this case Q indicates the batch-size (e.g. a box of 50 bolts would have a Q of 50) and n stands for the number of batches to order at every order. The categorisation used here distinguishes four strategies by having periodic review (including an R) or a continuous review (without R indicated) policy on the one hand and having a specific pre-specified order quantity (nQ) or working with an order up-to level (S) on the other hand.

	Periodic review	Continuous review
Fixed base replenishment quantity	(R, s, nQ)	(s, nQ)
Variable replenishment quantity	(R, s, S)	(s, S)

Figure 2.2: Classification of inventory control systems (Donselaar & Broekmeulen, 2017)

In the *(R,s,S)* strategy, the IOH will be monitored periodically with R time units between each monitoring. If, at monitoring the IOH is below the value s, an order will be placed immediately which will be of such value that the IP will be returned to S. Note that this order may in some cases be larger than $S - s$ as the IOH may have dropped below s between the last two sequential review periods.

The *(s,S)* strategy includes no periodic review but a continuous one and will therefore always order directly at the moment that the IOH falls below the s level. This means that the order size is always $S - s$.

In the *(s,nQ)* strategy, the IOH is also monitored continuously. If at any point the inventory position falls below s, a new order is placed of nQ units. Q being the standard order size of that product and n being the minimum integer for which the order size will return back to or above s. In this case the s therefore functions as both the reorder and order up-to level.

Finally, the *(R,s,nQ)* strategy functions in a similar way to the *(s,nQ)* but uses periodic monitoring. This means that the orders may be larger due to an even lower IOH at the moment of ordering.

2.2. Joint spare inventory and maintenance scheduling models

Maintenance scheduling and spare inventory control are often seen as separate functions because they seem like two different sets of operations. Maintenance scheduling deals with maintenance workers having to inspect and/or service parts in the field, while spare inventory control deals with warehousing and procurement of materials and equipment. More so, viewing these two aspects as separate greatly increases the ease of modelling and subsequent decision making. Many models for maintenance operations scheduling (Lidén, 2015; H. Wang, 2002; Al-Turki, Duffuaa, & Bendaya, 2019) have been developed as well as models detailing spare inventory control strategies (Hu, Boylan, Chen, & Labib, 2018; Basten & van Houtum, 2014). While these models are widely applied in practice, some real-world situations require both aspects to be addressed simultaneously. This holds especially true for components and systems that want to balance uptime with costs. When the uptime of a system is incredibly important as compared to the costs (e.g. production lines) a system owner is usually more concerned with maintenance scheduling. After all, when one component fails in a production line, the entire system shuts down. Spare parts for those facilities however might be quite cheap to procure and/or warehouse. On the other hand, when regarding a

large fleet of bicycles the fleet owner may be less concerned with being extremely accurate in maintenance scheduling. Instead they may concern themselves more with the vast number of spare parts that might have to be kept to service all those bikes.

To develop a maintenance decision support model for application in railway infrastructure management, a joint model that takes into account both spare inventory control and maintenance scheduling would be the most suitable form. A base model was selected from literature to build upon. This model is extended in order to be better applicable to the railway infrastructure domain that is being studied in this research. Hwang and Samat (2019) reviewed joint models but focused on models that included production planning which makes them unsuitable for this research. Van Horenbeek et al. (2013) have reviewed many spare inventory and maintenance scheduling models based on several characteristics. While they were very thorough, some extension is necessary for the current research. Especially because of the developments in these models over the past decade.

2.2.1. Model requirements

To find a model in literature which is suitable to use as a base for the development of the model which is the product of this research a preliminary selection of models for further review has to be made. This selection is performed using three basic requirements regarding 1) the type of maintenance strategy, 2) the type of inventory control strategy and 3) the system which the model applies to:

Using the typology of Huang et al. (2012) explained in section 2.1.1, the most advanced strategies in maintenance are forms of Condition Based Maintenance. Because of the importance of proper and timely maintenance action in the large systems that railway infrastructure entails, one would strive for the best possible maintenance strategy. On the other hand these large systems are hard to monitor. Installing sensors on each and every part of a national railway network would create a vast web of electronics and an even more unwieldy stream of produced data. Note that the digitisation of condition monitoring does already occur but only in specific part categories like electronics. Therefore, for the sake of this research the new model to be developed will be aimed towards parts which have not already been implemented with means of continuous monitoring. As a result, *condition based monitoring with periodical inspections* is deemed the most suitable strategy for the new model to be developed. Consultation with actual maintenance providers (MPs) and infrastructure managers (IMs) in the field also showed that this corresponds with the actual operational strategy in practice.

The next strategic requirement for a model to function as the base for the new model to be developed, regards the inventory control of spare parts. Ebeling (2004) describes a host of methods for predicting demand and optimising stock keeping decisions like reorder levels and order quantities. These decisions can have a large effect on the overall maintenance cost of a part. Rausch and Liao (2010) implement one of these inventory control strategies in a maintenance operation. While their research concerned manufacturing equipment, the same conditions (minimised downtime and production costs while maximising reliability) apply to the rail system. Rausch and Liao (2010) also studied a CBM based maintenance system and incorporated an (S,s) inventory policy. Again, consultation with an IM in the field showed that this corresponds with what is the standard in practice. Because MPs often outsource the warehousing of parts to a third party, the actual warehousing and keeping track of parts is done very accurately by a specialised operator. This means that all in- and outgoing stock is recorded which, according to the typology by Donselaar and Broekmeulen (2017) would indicate that some form of Continuous review policy is used (either (s,S) or (s,nQ)). Whether a fixed base replenishment quantity is used or not is highly dependent on the part itself. For the sake of this research *both continuous review strategies* are deemed suitable for the railway infrastructure context.

Finally the base model which is used for the development of the new model has to be applicable to *Multi-unit systems*, meaning that the model considers multiple but identical installed parts, e.g. a specific type of bearing which is used multiple times in a machine (yet, no other parts than that type of bearing should be considered in a single run of the model). In the railway infrastructure domain systems grow excessively large, very few items are installed singularly. Most items are installed multiple times across the system. A single pool of spare units will thus serve multiple installed units of equipment.

2.2.2. Available models

Based on the three requirements developed in the previous section, the literature is reviewed to find the models which are potentially suitable as base models for this research. A previous literature review of joint spare inventory and maintenance scheduling models is utilised as well as further exploration of literature that was not included in that review. The review by Van Horenbeek et al. (2013) studied 23 spare inventory control and maintenance scheduling models. Eight of these used a periodic instead of a continuous review inventory policy. Of the remaining 15, two were single unit systems and only three of the 13 models left used a condition based maintenance system (L. Wang, Chu, & Mao, 2008; Xie & Wang, 2008; L. Wang et al., 2009). These three models as well as four others are explained in some more detail below for the sake of comparison. A special aspect of all of the models and methods described in this section is that they all involve granulating time. Meaning that the system is only reviewed, and decisions only being taken, at discrete time steps. By decreasing the length of these time steps continuous time could be approached but would also highly increase computational effort in optimising the system. One of the downsides of granulating time is that no conclusions can be drawn about what happens within a single time step. This translates into uncertainty regarding downtime and its impact on the system as a whole. Yet, because of practical constraints to computation time which outweigh the need for certainty and accuracy, all models considered will make use of granulated time.

The model by L. Wang et al. (2008) considers a system in which multiple identical parts deteriorate over time using Markov-Chains. This allows for modelling of several different states between an as-good-as-new part and a failed one. While this method allows for a higher level of customisability of the model, thereby allowing the users to approach the real-world situation very closely, a disadvantage is that it requires a large amount of data to base the failure dynamics on. Moreover, it creates a higher computational requirement which is not applicable or necessary for railway infrastructure purposes. Another drawback of this model for railway application is that it does not include options for minimal repairs or consider component inter- dependence.

Xie and Wang (2008) combine elements of condition based and time-based preventive maintenance. In their proposed model parts are replaced preventively according to a set time schedule, however, if the parts' deterioration crosses a certain threshold before that time, it will be preventively replaced based on that condition. This approach is very unique but their CBM strategy also includes sensors to continuously monitor the parts' condition. As described this would make it unfit as a base model.

L. Wang et al. (2009) use a continuous deterioration process instead of a step-wise one. This approach is somewhat more accurate to the real-world process and is facilitated by using an analytical approach to their model. The CBM approach used also includes perfect, instantaneous replacements and no minimal repairs. Lifetimes of each part, as determined by the failure threshold of deterioration, is stochastically modelled for each lifetime. An (s,S) inventory policy is used in which no more than one order of spares can be placed at one time. Because the model only handles a single component type, no interdependence between the failure behaviour of parts is taken into account. The simplicity of the model by L. Wang et al. (2009) lends itself well for

expansion and alteration. The optimisation method being a genetic algorithm however lends itself only to smaller problems involving only few components installed.

Next to the models reviewed by Van Horenbeek et al. (2013), four more potential base models are reviewed here.

J. Wang and Zhu (2021) modelled a situation in which parts degrade over time. They include stochastic degradation but discretised the deterioration into stages to differentiate a failed or working condition of a part. Components are considered not to self-announce failure, meaning that a part can be failed without the MP noticing directly. In case of a CBM policy with periodic review, which is the case here, this would mean that the failure would only be detected upon the next inspection of the failed part. As a logical carry-through of this un-announced failure aspect, maintenance is only performed in time-steps which also include an inspection. One big caveat to the situation studied by J. Wang and Zhu (2021) is that it concerns a *'k-out-of-n:F system'*. In which a system is considered to contain a total of n components and when k out of those are in a failed state simultaneously at some point, the system will fail. While the step-wise deterioration of each individual part is not linked to other parts, the subsequent maintenance decisions in this case obviously are linked by the condition of multiple parts. This part-interdependence is one of the aspects that in the vast infrastructure systems is less applicable. J. Wang and Zhu (2021) also only consider replacement of parts as their only maintenance action and thus include no minimal repairs. To actually model and optimise the maintenance scheduling and inventory control, a Markov Decision Process (MDP) is used in this model.

Keizer et al. (2017) also consider a multi-unit system. Again, the deterioration of a part is approached as progressing in discrete states. The first of those states is the as-good-as-new condition and the last is the failed state. This model, like many others, assumes the repair time itself to be instantaneous. This because the repair time, which rarely takes longer than a few hours, often pales in comparison with the lead time and the component's lifetime, which often encompass weeks/months and years respectively (Keizer et al., 2017). Repair is not only assumed to be instantaneous but also perfect, meaning that when a part is repaired/replaced it is always a successful repair or replacement and the part is immediately as-good-as-new. The step-wise deterioration for each part is modelled using a Poisson distribution and parts have no influence on other parts' deterioration processes, in other words, there is no part interdependence. Notable is that Keizer et al. (2017) define the cost of operation and replacement as a function of the deterioration of a part. Thereby representing that a more deteriorated part functions less efficiently and is harder to replace than a less deteriorated one. They use an (s,S) inventory policy and the process of operational decision making in each time step is approached with an MDP formulation.

Van Horenbeek and Pintelon (2015) describe a system which includes multiple different parts. This approach is often seen in production settings where multiple parts depend on each other and a single machine/function containing a limited number of different parts can be seen as an isolated system. This however indicates the interdependence of parts. Once the system breaks down due to a single or a few parts failing, all other parts stop working and the deterioration patterns are influenced as a result. To optimise their model, Van Horenbeek and Pintelon (2015) produces a sequential optimisation model. This approach entails the optimisation of the repair/replacement time for components separately, at first ignoring inter dependencies. Thereafter, the maintenance actions are compared and grouped to produce an optimal savings on maintenance costs while minimising the risk produced by deferring from the optimal repair/replacement time. The spare inventory policy used is rather unorthodox. One influential constraint is that at most one spare can be held for each non-identical component in the system. Moreover, they base spare procurement decisions and -timing on information about the predicted Remaining Useful Life (RUL) of a part.

Finally, W. Wang (2012) uses an (R,s,nQ) inventory control model. While this disqualifies it as a potential base model as described earlier in this chapter, the model itself is still valuable to review in terms of inspiration for possible extensions on another possible base-models. The model does include a continuous deterioration process, but no component inter dependencies. Also W. Wang (2012) assumes the duration of repair operations or inspections to be negligible while still including costs for both. An interesting aspect is the modelling of the failure behaviour of parts. This model assumes large numbers of parts installed in a system and therefore takes a Homogeneous Poisson Process to approximate the arrival of defective parts. Rather than modelling parts separately, this model thereby sees the entire set of identical installed parts as a queue from which parts stochastically progress to the failed state.

2.2.3. Scientific gaps

The literature review of joint spare inventory and maintenance scheduling models has combined the work of Van Horenbeek et al. (2013) and a review of the more recent research published in this field. An overview of the most important aspects of each of the reviewed models as well as the model developed in this research are displayed in table 2.1.

Table 2.1: Literature review table *: Keizer et al. (2017) used a continuous-review CBM policy, therefore each failure is noticed immediately without having to be announced between inspection periods. **: J. Wang and Zhu (2021) use a special form of the (s,S) policy where a variable order up-to level is used which depends on the system condition.

Authors	Staged deterioration	Self-announced failure	(s,S) or (s,nQ) Inventory policy	Markov Decision Process	Component inter-dependence	Minimal repairs
L. Wang et al. (2008)	x		x			
Xie and Wang (2008)			x			
J. Wang and Zhu (2021)	x		**	x	x	
Keizer et al. (2017)	x	*	x	x		
Van Horenbeek and Pintelon (2015)			x		x	
L. Wang et al. (2009)		x	x			
W. Wang (2012)		x				
This paper		x	x			x

Table 2.1 shows the unique contribution that the new model has to the current literature. The studied literature has not yet shown an example of a joint model which:

- Employs continuous deterioration process, making it more accurate to the true state of the part.
- Includes self-announced failure which allows maintenance engineers to perform maintenance between inspections, increasing desired responsiveness in the infrastructure setting.
- Uses an analytic decision process, thereby decreasing complexity and increasing use-ability as compared to ones which employ an MDP.
- Takes into account no component inter-dependence, thereby suiting infrastructure application better than ones that do.
- Includes minimal repair, again making it more suitable for infrastructure as a system that remains in a failed state for longer is extremely undesirable in those cases.

Because of the characteristics of the model to be developed and the close resemblance to the model by L. Wang et al. (2009), this model is chosen as a base model to develop further into the sought after model in this research.

2.3. Optimisation

The main function of the previously described joint maintenance scheduling and inventory control models is to estimate costs for a certain entered scenario. This entered scenario includes data about for example, inspection intervals, failure behaviours, cost factors and many more. Once the model is able to accurately estimate a cost as a result of the entered variables, one could try to find the values for specific variables which produce the most ideal (which in the case of this research means lowest) cost. This process is called *optimisation*. The field of mathematical optimisation is very broad and actively researched. The versatility however indicates many different avenues for ones research to specify in. To reduce the number of options, some aspects of the studied problem can be taken as demands for an optimisation technique to satisfy. These characteristics, presented as in the topology of (Mueller, n.d.):

- Decision variables *integer*
- Objective function *simulation*
- Number of objectives *single*
- Type of constraints *simulation*

These four aspects make the model computationally expensive to evaluate. Moreover, they make a closed-form analytic description of the objective function very hard if not impossible to find. This type of problem lends itself very well to optimisation using *Surrogate modelling*.

2.3.1. Surrogate modelling

Surrogate modelling is a technique used in optimisation which uses experimental data of decision variable values and accompanying objective values to estimate a relationship between both. This estimated relationship is called the *surrogate* (Jones, 2001).

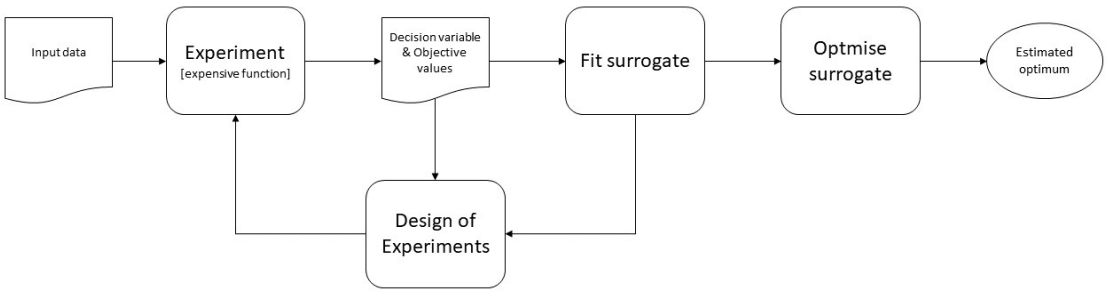

Figure 2.3: Modelling framework

Figure 2.3 shows the modelling framework for this research. This details the way in which surrogates are used in optimisation: the center of a surrogate modelling methodology is the *experiment*. The experiment represents the central problem which is the subject of optimisation. This experiment usually takes some input data, part of which is formed by the *decision variables*. This input data is subsequently used to output an objective value. The surrogate modelling method then uses the 'objective value - decision variable value' pairs as a basis to estimate a function that connects both elements together mathematically ($(objective\ value) = f(decision\ variable\ values)$). This estimated function is the actual *surrogate*. To further improve the accuracy of the surrogate more data is generated through repeating the experiment. The values of the decision variables to use in the input for the repeated runs of the experiment can each time be based on the estimated surrogate and the previously

gathered data through a *Design of Experiments* (DoE). If the surrogate is deemed satisfactory to the user or computational resources are depleted, the experimentation cycle can be terminated. The final surrogate is now a known function which can be optimised using conventional optimisation techniques which yields the estimated optimal values for the original decision variables.

The specific elements of this process can take many forms to serve different strategies. Each combination of DoE, types of surrogates fit and methods for fitting those surrogates, together form a *surrogate modelling algorithm*. To find the most appropriate algorithm, aspects like the nature of the problem itself, the resources available and the expertise of the user all have to be taken into account.

As described in the topology of Bhosekar and Ierapetritou (2018), surrogates can be used for three purposes: 1) prediction and modelling, 2) derivative-free optimisation (DFO) and 3) feasibility analysis. In this research the use for surrogates is DFO of an objective function which is expensive and therefore lacks derivative information necessary for conventional optimisation. While some other forms of DFO exist that are not based on surrogates (Hooke & Jeeves, 1961; Nelder & Mead, 1965), these do not suit complex functions well. Surrogate model based derivative free optimisation (Model-based DFO) is then again subdivided into local and global methods. Local methods focus on finding local optima with great precision (often assisted by indicating a small region in which an optimum is likely to be located) while global methods try to indicate the region in which a global optimum would reside. Because in the studied situation the expensive function is completely unknown and approached as a black-box no prior indication can be given regarding the location of the optimum. Moreover because the model will most likely be used mainly to aid in managerial decisions, a global optimum would be more beneficial than a local one, even though it would be less precise. The types of surrogate algorithms most suitable for this research are therefore, according to Bhosekar and Ierapetritou (2018), Model-based global DFO algorithms.

All the considerations described here were used to find candidate algorithms that might be suitable for the problem studied. A selection of algorithms available in literature and software were reviewed to select the ones which were used in experimentation.

Randomsearch is one of the most basic algorithms available (Bergstra, Yamins, & Cox, 2013b). It randomly selects values for the decision parameters and tests the core problem (the experiment) to find the value of the objective. It skips fitting a surrogate and its DoE can be described as taking a random pick for every decision variable from a uniform distribution over each variable's respective range. Surprisingly this method has proven quite useful in certain situations. Hence, it is wise to include this algorithm as a base-line for the performance on one's surrogate modelling approach. *SMAC* uses random forests (Hutter, Hoos, & Leyton-Brown, 2010), an approach which is deemed very well suited for categorical or discrete variables. *HyperOpt* (Bergstra et al., 2013b) uses the Tree Parzen Estimators (TPE) algorithm. In practice *SMAC* and *HyperOpt* produce very similar results. SMAC however is less usable in this instance in terms of software integration. *COMBO* or COMmon Bayesian Optimization uses a continuous surrogate (Ueno, Rhone, Hou, Mizoguchi, & Tsuda, 2016). Similarly, *Bayesianopt* and *pyGPGO* (Jiménez & Ginebra, 2017) also use a continuous surrogate as does *DONE* (Bliek, Verstraete, Verhaegen, & Wahls, 2016), which has specifically shown a distortion effect at the edges of the search space, which is advisable to keep in mind when selecting an algorithm to fit ones specific problem situation. *CoCaBo* is tailored to combine categorical and continuous inputs (Ru, Alvi, Nguyen, Osborne, & Roberts, 2020). *IDONE* which is an extension on the *DONE* algorithm, uses a continuous surrogate as well. Though, it is adjusted in a way that its final optimal solution is always an integer. (Bliek, Verwer, & de Weerdt, 2020a). It thereby subverts the distortion around the edges of the search space. One of the drawbacks of

IDONE is that it has a limited exploration parameter. This means that it is more local-leaning. While none of the algorithms described can fully assure that a global optimum is reached, one might consider expanding the exploration parameter somewhat to increase the search area of the *IDONE* algorithm. This is exactly what is effectively done in the *MVRSM* algorithm (Bliek, Verwer, & de Weerdt, 2020b), apart from also accommodating for continuous variables rather than only integer ones. Finally *Submodular Relaxation BOCS* (Deshwal, Belakaria, & Doppa, 2020), being an extension of the better known classical *BOCS* (Baptista & Poloczek, 2018), might also be suitable for this problem although practical concerns regarding the computational effort of this method could be considered in the choice to include it in the roster of algorithms to be tested.

2.3.2. Algorithm comparison

To compare algorithms in terms of their suitability for the studied problem they are scored on four aspects. All algorithms are described in section 2.3.1 and scores can be found in table 2.2. Scores can be – (very negative), - (negative), x (neutral, not applicable or unknown), + (positive), or ++ (very positive).

Table 2.2: Surrogate Modelling algorithm comparison

	Level of computational complexity	Capacity to work with discrete variables	Continuous surrogate	Local-leaning	Overall
Randomsearch	++	++	x	++	++
SMAC	–	+	+	x	x
HyperOpt	++	+	-	+	++
COMBO	x	–	–	x	–
Bayesianopt	x	–	–	x	–
pyGPGO	x	–	–	x	–
DONE	x	–	–	x	–
CoCaBo	x	-	x	x	-
IDONE	+	++	x	-	+
MVRSM	+	+	x	++	++
BOCS	–	+	x	x	-
submodular Relaxation BOCS	–	+	x	x	-

These four aspects represent both theoretical constraints and practical considerations. All of which are of influence in selecting the algorithms that could be most applicable to the particular problems studied.

Firstly, the decision variables (order up-to level, reorder level and CBM threshold) are integer variables. Depending on the type of surrogates used and the method for optimising those surrogate models, not all algorithms are well capable of performing optimisation tasks for integer valued decision variables. Secondly, one of the most interesting cases, and the seed for this research is the ProRail case described in section 3.1. In this case ProRail is interested in finding those parts for which a contractor might, for financial reasons, decide to keep little or no spare inventory. If this were the case for a part, the optimal value for the order up-to and reorder levels would lie close to or on 0. This is also the edge of the search space. Meaning that this is close to the lower bound of one or more of the decision variables. Depending on the search strategy of an algorithm, it might be able to better or worse deal with searching at the edge of search-spaces. One of the most indicative aspects of the algorithms regarding their distortion at the edges of the search space is whether they use a continuous surrogate. To fit a continuous function as a surrogate, the algorithm is often required to analyse both sides of a discrete value of the decision variable. If this has to happen at the edge of the search space, this is impossible, causing the distortion effect. Thirdly, an algorithm can be more- or less local-leaning. Meaning that it may be more or less inclined to get stuck analysing a local minimum of the objective value. For the sake of this research,

a less local-leaning algorithm is considered to be better fitting. This because very little is known about the actual function between the objective and the decision variables. Finally, the last aspect is the practical ease of computation. Because all algorithms are intended to be used in practice, they have to be practical in their application in terms of computational requirements and the technical expertise necessary to work with them.

The comparison in table 2.2 shows that *Randomsearch*, *HyperOpt*, *IDONE* and *MVRSM* are the four most suitable algorithms for the problem being studied and the only algorithms with a positive overall score.

3

Problem description

To describe the problem, and more specifically the model that represents the problem, first general explanation and definitions of the model are described in this chapter. Thereafter sets of assumptions are presented which pertain to *parts*, *maintenance operations* and *spare inventory control*. The operational consequences of these assumptions and mechanics are visualised in the *action sequence* and finally the objective of the entire model is described, the *cost*.

3.1. Modelling definitions

This research models a railway infrastructure system in which parts are installed, break down over time, and are replaced by spare parts. This model serves to fill a gap in literature (see chapter 2), but also a need in practice. As manager of an infrastructure system, ProRail values proper systematic insight into their processes and the processes of their contractors. Mathematical models can be helpful to better understand those processes. An example of a use case is finding parts in the system for which a contractor might, for financial reasons, hold little or no spare inventory. In this case, ProRail could argue that at least some inventory should be held for the sake of the reliability of the railroad system. Through running a mathematical model which simulates the contractors' maintenance scheduling and spare inventory control decisions, ProRail might be able to pinpoint those parts that have a high likelihood of being under-served in terms of spare parts due to financial considerations by the contractor. In this case an accurate spare inventory control and maintenance scheduling model which is specifically tailored to the railway infrastructure domain would be very beneficial for ProRail as a company and for all users of the rail infrastructure system.

The two main processes of spare inventory control and maintenance scheduling are modelled with the aim to determine the cost-optimal values of three decision variables: the order up-to level, the reorder level and the condition based maintenance threshold. To describe the system, figure 3.1 displays a hypothetical example. This example contains a part (a certain type of fuse box for instance) which is installed two times in the system (seen in the top two plots of the figure). Each of these two parts can be seen to deteriorate over time represented by the increasing level of deterioration on the y-axis. Eventually each part is replaced by a new part which leads to a deterioration level of 0 or as-good-as-new condition. Meanwhile, every time a part is replaced, the spare stock level (seen in the third plot in the figure) gets depleted by one part. Once the spare stock falls below a certain level (the reorder level) a new order is placed for spare parts which arrive a few periods later (order cycle is displayed in the bottom plot of the figure).
Some basic concepts which seem obvious can sometimes be cause ambiguities later on, some of these concepts are therefore introduced here.

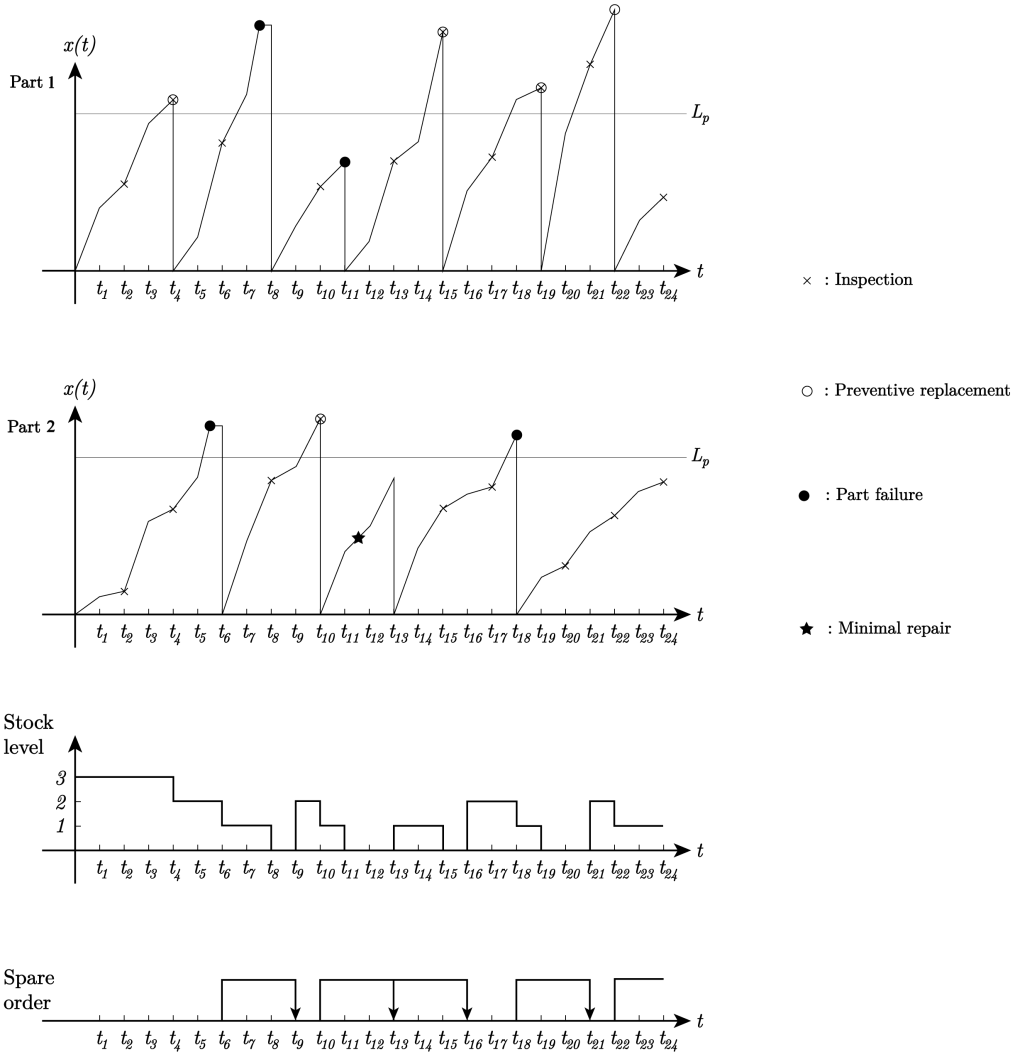

Figure 3.1: Behaviour of part failure and maintenance decisions on stock levels and system functioning (based on L. Wang et al., 2009)

Parts can be any field-replaceable sub-section of the system that is being studied. This means that a part might be a single, non-divisible thing, for example a piece of railroad track. Or a more complex sub-system like electrical relays.

Deterioration can be interpreted as any form of wear on a part. It can be caused by friction, heat or other physical abuse. Deterioration can take many different forms, it can be measured in the number of millimeters of wear on a piece of railroad track or the resistance of a train axle counting sensor.

Maintenance is the overarching term, used to indicate inspection (to determine the amount of deterioration of the part), replacement (installing a new or as-good-as-new part) or repair (restoring a failed part to working condition just before the part failed) actions of a part.

Spare stock describes the amount of spare parts that are available to perform maintenance actions with. Spare parts are identical to the parts installed in the system but are still as-good-as-new.

Corrective replacement is performed when a part has already failed. It replaces a failed part by a new or as-good-as-new part.

Condition Based Maintenance threshold denotes the level of deterioration at or above which a

preventive maintenance action is desired. The basis of CBM lies in the balance between preventive and corrective action. By setting the CBM threshold to a certain level one can influence this balance to fit the objectives of the policy.

Preventive replacement would preferably be performed every time a part's deterioration is observed to exceed the CBM threshold, in this case a part which is still operational will be replaced by a new one. While preventive replacement is operationally similar to corrective replacement (it takes the same amount of resources and the result, a newly installed part, is the same) the costs may be vastly different.

More concepts that are included in the studied systems are described in this chapter along with their accompanying assumptions and interactions. In subsection 3.2 parts and their deterioration process are explained, subsection 3.3 handles the maintenance actions and subsection 3.4 details the actions controlling inventory of spare parts.

3.2. Parts

The studied system is modelled for n identical parts. Regarding these parts, the following is assumed:

- Each part, upon installation, is as-good-as-new, meaning that the deterioration of the part, described by $x(t)$ is 0. After installation, the part immediately starts to deteriorate incrementally over time at a rate of $\frac{\Delta x}{\Delta t}$. Where Δx is randomly picked from a pre-determined probability distribution for every part for every time-step. This indicates that $x(t) = x(t-1) + \Delta x$. Since Δx is always positive, the deterioration will only increase linearly through time.
- Each part gets assigned a *failure threshold* X_f for their lifetime. Once the deterioration of a part exceeds this threshold, the part is considered *failed*. This threshold is picked from a pre-determined probability distribution similarly to the step-wise deterioration. Every time a new part is installed this threshold is re-picked and this threshold is always positive.
- All parts in the considered system are identical. All failure behaviour therefore is described by the same distributions throughout the system. This means that local differences in environmental conditions or different levels of use of a part compared to another is not taken into account in this model.
- Part failures are assumed to be self-announced. This means that in a period which does not include an inspection, failed parts are still noticed and replacement decisions can be made as a result outside the inspection cycle.

3.3. Maintenance operations

Three different maintenance operations are considered: inspection, replacement and minimal repair. Replacement can be done either in a preventive or corrective fashion. All of these operations have a certain cost. Where performing inspections is governed by pre-determined time intervals (T), both the minimal repair as well as the replacement decisions are based on the condition of the part (Condition Based Maintenance). The specific condition can either be the level of deterioration in comparison with a CBM threshold or whether part has failed or not. The following assumptions apply:

- Inspection is performed on a part every T time steps. Meaning that if the inspection interval T for instance equals 4, inspections are performed in time step 4, 8, 12, 16, ... until the part is replaced or failed. After replacement of a part, the cycle restarts, this means that if a part is replaced at $t = 17$, the new inspection cycle will look like 21, 25, 29, ...

- Each inspection is instantaneous and non-invasive, meaning that it does not influence the deterioration level of the inspected part. Upon inspection the part's condition is perfectly determined.
- A part can only be replaced if spare part stock is available.
- Replacement as well as minimal repair are considered to be instantaneous.
- If a part is in a failed state, the system will try to correctively replace said part as soon as possible.
- If a part is inspected, and the deterioration is shown to exceed the CBM threshold L_p, a part will be replaced preventively as soon as possible.
- A part that can not be replaced correctively due to stock-out of spare parts will be immediately subjected to a minimal repair, meaning that the part will exit the failed state and will be restored to the condition directly before failure. The deterioration level will be reset to the level before failure and a new failure threshold X_f will be picked such that it is higher than the current deterioration level. The part will subsequently be scheduled to be replaced correctively as soon as spare stock allows.
- A part that can not be replaced preventively will be scheduled to be replaced preventively as soon as spare stock allows.
- If a part has failed or exceeded the CBM threshold but can not be replaced due to stock-out the 'no-more-inspection' clause will come in to effect. Meaning that the part will not be inspected any more until it is replaced.

3.4. Spare inventory control

Inventory of spare parts is controlled using an (s,S) policy in which $s < S$. This policy entails continuous monitoring of the spare IOH. The stock is assumed to always be available for operations and all stock keeping actions are instantaneous. Further assumptions are:

- Only a single order can be outstanding at any one time. If the order marker $OC = 1$ no orders can be placed. Only once those ordered parts are delivered a lead time (t_s) later can another order be placed.
- The spare stock level (the physical amount of spares available) SL can never exceed S or become lower than 0.
- If spare stock equals reorder level s, new spares will be ordered if no order is outstanding at that time. Spare orders are always of the quantity $S - s$ in order to bring the Inventory Position back up to the order up-to level S.
- Every replacement action decreases SL by exactly 1.
- Spares that remain in stock incur a holding cost each period that they are kept in stock. Parts can be kept in stock indefinitely and will not deteriorate while kept in stock. They have an infinite shelf-life.

3.5. Action sequence

To illustrate decisions made and subsequent actions during each time-step and for each part, figure 3.2 illustrates a decision tree.

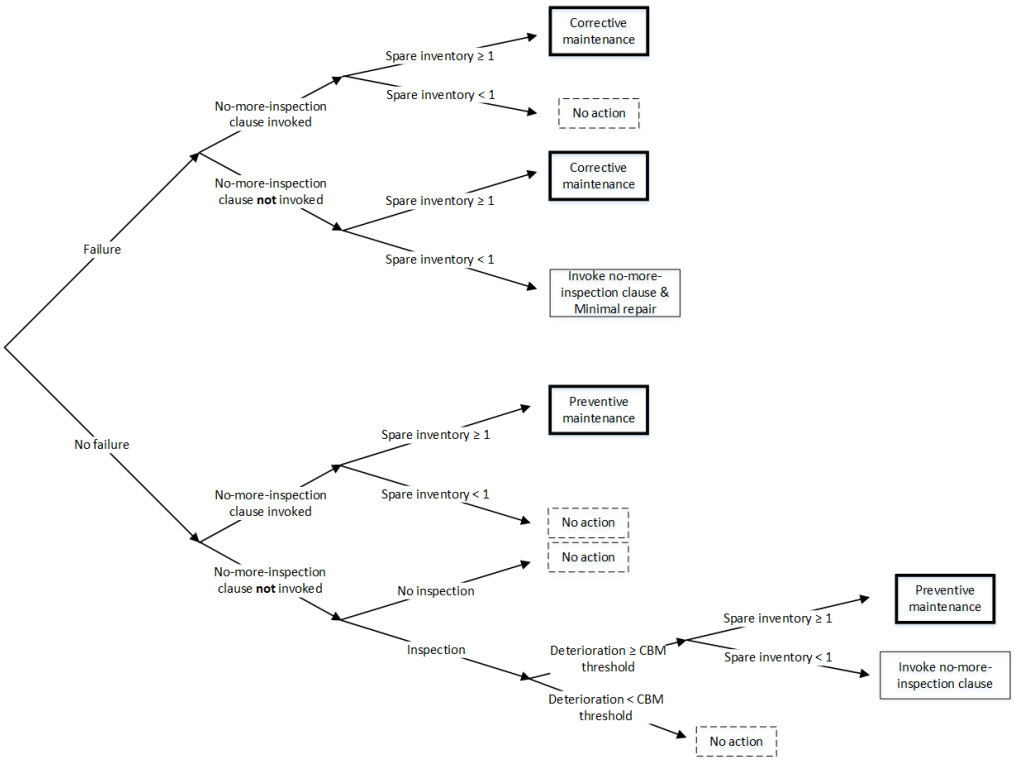

Figure 3.2: Representation of the actions/choices made for each part in each time step in the proposed model

3.6. Cost

All actions and decisions in the described model result in incurred monetary cost. Included cost parameters are Cost of inspection C_i [€/inspection], Cost of preventive replacement C_p [€/preventive replacement], Cost of corrective replacement C_c [€/corrective replacement], Holding cost per unit per time-step C_h [€/part/period], Cost of procurement of a spare part C_s [€/part], Cost of downtime per unit per time-step C_d [€/period] and Cost of minimal repair C_m [€/minimal repair action]. The simulation will simulate time steps to the amount of the time horizon t_m [time-steps]. Each time-step the simulation will run cost simulations for each part, n [parts] being the number of parts considered. If for example the time horizon equals 10 years, the time steps are 1 month in length and a specific type of railway switch is entered which is installed 20 times in the system, the simulation will simulate each of the $n = 20$ parts, for every $t_m = 120$ time steps.

All of these costs combined form the basis for the simulation's output which is the expected cost per part per time period EC [€/part/period]. This output is calculated by adding all costs together and dividing them by the time horizon multiplied by the number of parts included. All costs are tracked using counters, N_i [inspections] being the counter for the number of performed inspections, N_p [preventive replacements] the counter for the number of preventive replacements, N_c [corrective replacements] the counter for the number of corrective replacements, N_h [parts/period] the counter for number of parts held in stock, N_s [parts] the counter for the number of spare parts procured, N_d [periods] the number of time-steps of downtime and N_m [minimal repair actions] the number of minimal maintenance actions performed.
This all can be represented using Eq. (3.1):

$$EC = \frac{C_i N_i + C_p N_p + C_c N_c + C_h N_h + C_s N_s + C_d N_d + C_m N_m}{t_m \cdot n} \tag{3.1}$$

These costs can thus be calculated and are the objective to be minimised in this research.

4

Methodology

To further detail the developed model and its components, this chapter presents the general framework of the entire model as well as the simulation and optimisation steps in more detail.

4.1. Modelling framework

The Railway - Spare Inventory & Maintenance Scheduling model (R-SIMS) developed in this research is built using a simulation-optimisation framework. The simulation model is based on the existing model by L. Wang et al. (2009) and the optimisation is performed using surrogate modelling. Figure 4.1 shows the overview of the complete modelling framework. The first main element of the framework is the simulation model (further elaborated in section 4.3). This simulation model is able to take as input values for all parameters which constitute the part information (cost parameters, parameters describing lead times and inspection intervals etc.). Among the required parameters are the three decision variable values (reorder level, order up-to level and the preventive maintenance threshold). All parameters are described in section 4.2. The simulation model can then yield an estimate of the objective value, the estimated cost per part per period. The second main element is the surrogate modelling algorithm, four candidates for which are described in section 4.4. This algorithm uses three inputs: the simulation itself, the output of the simulation (the estimated cost) and the search space for each decision variable. It uses these inputs to estimate optimal values for the decision variables. Each iteration, it can use these values as inputs for the simulation model to further optimise the solution. Finally, when the algorithm is finished (cut-off by the user) it outputs the definitive estimated optimal decision variable values.

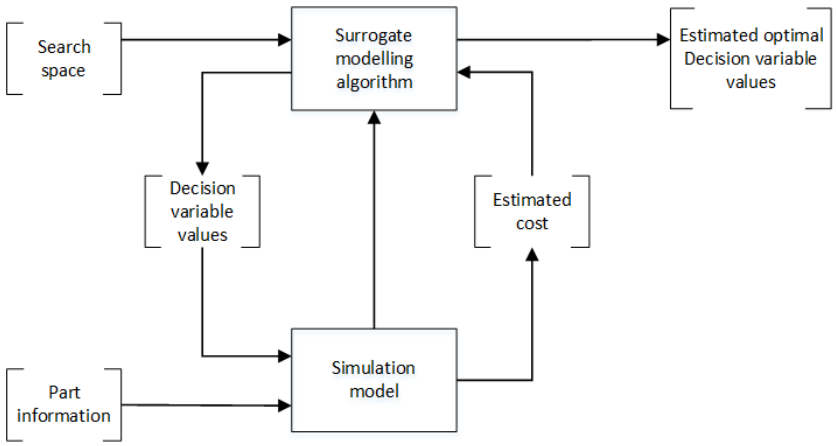

Figure 4.1: Overview of elements and interactions in the modelling framework

4.2. Parameters

The R-SIMS model is aimed at optimising three decision variables to fit a single objective. The decision variables are the order up-to level (S), the reorder level (s) and the CBM threshold (L_p) and the objective is the Expected Cost (EC). An estimate of the objective considering a single value for each of the three decision variables is gained through running the simulation part of this model. Some parameters relate directly to the objective as shown in section 3.6 in equation 3.1. These cost factors (C_i etc.), the simulation horizon (t_m), lead time (t_s), inspection interval (T) and the number of parts (n) are considered given and constant throughout the simulation. The counters (of type N_i) are initialised at 0, are discrete and progressively increase throughout the simulation duration in accordance with the maintenance decisions taken. Next to these constants and counters, the model includes the stochastic variables X_f and Δx which determines $x(t)$, which are both determined by their probability distributions $Pr\{\Delta x = x\} = f^D(x)$ and $Pr\{X_f = x\} = f^F(x)$. Further the model contains a set of markers, which are binary variables that indicate important decision aspects. These markers are F for indicating a failed part, I for indicating whether a part needs to be inspected in the current period, IN to indicate whether a part needs to be inspected for the rest of its lifetime, ICR and IPR to indicate whether a part is correctively or preventively replaced in the current period. IR to indicate a replacement regardless of the type, OC to indicate whether an order is currently outstanding and IM to indicate minimal repair being performed on a part. To store the necessary (non binary) values in the model, four memory variables are used as well: $LastOT$ which takes the value of the last time-step in which spare parts were ordered, NA to indicate how many spare parts are delivered in the current period, SL to keep track of the inventory on hand and ST to remember the start of the current lifetime of a part.
To indicate, for each part-specific parameter, which part exactly it refers to, the index j is used. The failure threshold for component j is described with $X_{f,j}$, F_j indicates failure of component j and so on. Index j therefore is an index from a set that runs from 1 to n, $j \in \{1, ..., n\}$.

4.3. Simulation model

The simulation uses the Monte Carlo method to represent the stochasticity that is present in the real-world. The failure of parts as well as the incremental deterioration are considered to be stochastic. To simulate the railway parts over a certain period of time, time-steps are used. This means that the simulation sequentially makes decisions for the system for each time-step after another. The length of a time-step is abstract and can be assumed any value by the user of the model. The simulation uses the Monte Carlo method to produce a reliable estimation of the objective when considering a certain value of the decision variables. The idea behind Monte Carlo simulation is to run multiple iterations in order to deal with the stochasticity of the model. To determine the amount of iterations to run the method of Driels and Shin (2004) is used. The number of iterations is based on the desired accuracy of the outcome (the expected cost EC).

$$MC = \left(\frac{100 z_{\alpha/2} S_x}{E \bar{x}} \right)^2 \tag{4.1}$$

Where MC is the number of iterations, S_x is the estimated standard deviation of the outcome of the simulation, E is the desired percentile deviation from the mean, $\bar{x}$ is the estimated mean and $z_{\alpha/2}$ is the confidence coefficient. S_x and $\bar{x}$ are usually gathered through preliminary testing of the simulation.
If for instance a desired accuracy would be to have the outcome of the simulation not deviate more than 5% from the mean in 95% of cases (with a sample mean of 50 and standard deviation of 7), the number of runs would be $MC = \left(\frac{100*1.96*7}{5*50}\right)^2 \approx 30$

The nomenclature of all used variables and parameters used is given in table 4.1 and pseudo-code is presented in Algortihm 1.

Nomenclature

$x_j(t)$	Deterioration level of part j at time-step t
$X_{f,j}$	Failure threshold for deterioration of part j
$f^F(x)$	Probability Density Function for the Failure threshold
Δx	Incemental deterioration
$f^D(x)$	Probability Density Function for the Incremental deterioration
L_p	Condition Based Maintenance threshold for deterioration
S	Order up-to level
s	Reorder level
EC	Expected Cost per part per time period over the simulation horizon
EC_{MC}	Cumulative Expected Cost per part per time period
n	Number of parts installed
T	Inspection interval
t_s	Lead time
I_j	Marker for whether part j should be inspected in the current time-step
F_j	Marker for whether part j is in the failed state in the current time-step
IPR	Marker for whether part j is replaced preventively in the current time-step
ICR	Marker for whether part j is replaced correctively in the current time-step
IR	Marker for whether part j is replaced in the current time-step
IM	Marker for whether part j is minimally repaired in the current time-step
IN	Marker for whether part j requires inspection in the future
OC	Marker for whether an order for spare parts is outstanding or not
IMC	Counter of number of minimal maintenance actions per period
$LastOT$	The last time a new order was placed for spare parts
NA	The number of spare parts that arrive in the current time-step
SL	The Inventory On Hand for spare parts
ST_j	The time-step in which part j was installed (start of new lifetime)
C_i	Cost of inspection
C_p	Cost of preventive replacement
C_c	Cost of corrective replacement
C_d	Cost of downtime (per part per period)
C_s	Cost of procurement of spare part
C_h	Holding cost of spare stock (per part per period)
C_m	Cost of minimal maintenance
N_i	Number of inspections performed
N_p	Number of preventive replacements performed
N_c	Number of corrective replacements performed
N_d	Number of periods of downtime (cumulative)
N_s	Number of spares procured
N_h	Number of spares in warehouse (cumulative)
N_m	Number of minimal maintenance tasks performed
j	Index, j = 1,2,...,n
t	Current time-step
t_m	Simulation horizon
MC	Number of Monte Carlo iterations

Table 4.1: Nomenclature for simulation model

Algorithm 1 Simulation Model for finding the objective value (estimated cost)

Require: C_i, C_p, C_c, C_h, C_s, C_d, C_m, t_s, T, n, t_m, L_p, S, s, MC ▷ ***Data entry***
Ensure: EC

Count := 0
Cost := 0
while Count ≤ MC **do**
  $LastOT := 0$, $OC := 0$, $NA := 0$, $SL := S$, $t := 0$ ▷ ***Initialization***
  $N_i := 0$, $N_p := 0$, $N_c := 0$, $N_m := 0$, $N_s := 0$, $N_d := 0$, $N_h := 0$
  for $j = 1$ to n **do**
    $F_j := 0$, $I_j := 0$, $ICR_j := 0$, $IPR_j := 0$, $IR_j := 0$, $IM_J := 0$, $ST_j := 0$, $IN_j := 1$
    Failure threshold := $X_{f,j}$ ▷ $P[X_{f,j} = x] = f^F(x)$
    Deterioration level := $x_j(t)$
    $x_j(t) = X_{f,j} \cdot U(0,1)$
  end for
  while $t \le t_m$ **do**
    if $t - LastOT = t_s$ and $OC = 1$ **then** ▷ ***Spare order handling***
      $NA = S - s$
      $OC = 0$
    else
      $NA = 0$
    end if
    $SL := SL + NA$
    $q := \{1, 2, ..., n-1, n\}$
    while $q \neq \{\}$ **do**
      $j \in q$
      $q := q - j$ ▷ ***Deterioration and failure***
      $x_j(t) := x_j(t-1) + \Delta x$ ▷ $P[\Delta x = x] = f^D(x)$
      if $x_j(t) \ge X_{f,j}$ **then**
        $F_j := 1$
      else
        $F_j := 0$
      end if
      if $IN_j = 0$ **then** ▷ ***No-more-inspection clause***
        if $F_j = 1$ and $SL > 0$ **then**
          $ICR_j := 1$
        else if $F_j = 0$ and $SL > 0$ **then**
          $IPR_j := 1$
        end if
      else
        if $F_j = 1$ and $SL > 0$ **then** ▷ ***Corrective replacement or minimal repair***
          $ICR_j := 1$
        else if $F_j = 1$ and $SL \le 0$ **then**
          $IN_j := 0$
          if $IM_j = 0$ **then**
            $IM_j := 1$
            $IMC := IMC + 1$
            $F_j := 0$
            while $x_j(t) > X_{f,j}$ **do**
              $X_{f,j} := \{X \mid P[X = x] = f^F(x)\}$
            end while
          end if
        else
          if $ST_j \neq t$ and $(t - ST_j) \mod T \equiv 0$ **then** ▷ ***Inspection and preventive replacement***
            $I_j := 1$
            if $x_j(t) \ge L_p$ **then**
              if $SL > 0$ **then**
                $IPR_j := 1$
              else
                $IN_j := 0$
              end if
            end if
          else
            $I_j := 0$

Algorithm 1 Simulation Model for finding the objective value (estimated cost) (continued)

```
61:                 end if
62:             end if
63:         end if
64:         if IPR_j = 1 or ICR_j = 1 then                    ▷ Installation of new part
65:             IR_j := 1
66:             x_j(t) := 0
67:             ST_j := t
68:             X_{f,j} := {X | P[X = x] = f^F(x)}
69:             IM_j := 0
70:             IN_j := 1
71:             SL := SL − 1
72:         else
73:             IR_j := 0
74:         end if
75:         if SL ≤ s and OC = 0 then                         ▷ Spare ordering decision
76:             OC := 1
77:             LastOT := t
78:         end if
79:     end while
80:     N_i := N_i + Σ_{j=0}^{n} I_j                          ▷ Counting, resetting and advancing time
81:     N_p := N_p + Σ_{j=0}^{n} IPR_j
82:     N_c := N_c + Σ_{j=0}^{n} ICR_j
83:     N_s := N_i + NA
84:     N_d := N_d + Σ_{j=0}^{n} F_j
85:     N_h := N_h + SL
86:     N_m := N_m + IMC
87:     for j = 1 to n do
88:         F_j := 0, I_j := 0, ICR_j := 0, IPR_j := 0, IMC := 0, IR_j := 0
89:     end for
90:     t := t + 1
91: end while
92: Cost = Cost + C_i N_i + C_p N_p + C_c N_c + C_h N_h + C_s N_s + C_d N_d + C_m N_m    ▷ Cost calculation
93: Count = Count + 1
94: EC_MC := EC_MC + Cost/(t_m · n)
95: end while
96: EC = EC_MC / MC;
```

The simulation can be described in several steps which are illustrated in the pseudo-code:
Data entry {Require}
This step entails the entry of all the necessary parameter values by the user. Note that the three decision variables are also required as the optimisation step lies a level beyond the simulation in the surrogate modelling algorithm. The output of the simulation is also defined as the estimated cost.
Initialization {L4 - L11}
All variables that do not pertain to a single part are initiated. As well as all values which are part-specific. Most notably every part gets assigned a failure threshold for the first lifetime and a deterioration level which is randomly assigned for a value between 0 and the failure threshold.
Next the first time-step begins, each time-step has the same sequence of actions and these are all processed in ascending order until the entered time horizon is reached.
Spare order handling {L13 - L19}
At the start of each period newly delivered orders are accepted. If an order was outstanding and the order has been placed a lead-time ago, this order will arrive and be added to the current inventory

position. Because of the (s,S) inventory control strategy, orders always amount to $S - s$ parts. If an order is accepted, the order marker can also be reset to 0.
Next each part is handled separately. The order in which the parts are handled is randomised.
Deterioration and failure {L24 - L29}
The part's deterioration level is updated by Δx. This value is picked from its appropriate probability density function (PDF). This new deterioration level can then be checked against the failure threshold to determine whether the part has failed or is still operational.
No-inspection clause {L30 - L35}
Instrumental in this model, is the option that parts are no longer inspected after they may have failed or exceeded the CBM threshold in earlier time-steps. If this is the case ($IN_j = 0$) the part will have to be replaced if the spare stock allows. The part will then be correctively replaced if it has failed and preventively be replaced if it is still operational.
Corrective replacement or minimal repair {L37 - L48}
If a part has failed and stock levels allow, it should correctively be replaced. If it has failed and stock levels are not sufficient, the part will be set to no longer be inspected, after all, the failure is now known and the part should be replaced as soon as possible. If it indeed has failed and no spare parts are available to replace, the part can be minimally repaired, but only if it has not been before. If it is minimally repaired, the marker is adjusted accordingly and its failure status is reset to operational as well. Finally the part's failure threshold has to be reset to a value, again picked from its appropriate PDF but in this case higher than the current deterioration level.
Inspection and preventive replacement {L50 - L63}
If the part has not failed at all, it will be considered to possibly be inspected. This inspection takes place periodically (with T as inspection interval) so if the current period in the part's lifetime is a multiple of T it shall be inspected. If upon inspection the part is found to have a deterioration level greater than or equal to the CBM threshold and enough spare stock is available, the part will be preventively replaced. If no stock is available, the part is set not to be inspected any longer and therefore to be replaced as soon as the spare stock allows.
Installation of new part {L64 - L74}
If a part is replaced, at this point this is recorded. This means that a new part is installed and some parameters have to be reset. The deterioration is set back to 0, indicating an as-good-as-new part. The start of the lifetime of the part is set to the current time-step. The failure threshold is renewed (again from its PDF). The minimal maintenance marker is reset as well as the marker for the no-inspection clause. Importantly, the spare stock is decreased by 1.
Spare ordering decision {L75 - L78}
Once the IOH of spares is updated, it is compared to the reorder level, if it has dropped to the reorder level and no order is outstanding, a new order will be placed, thereby resetting the last order time to the current time-step.
After all parts have been cycled through, each time-step ends with administrative steps.
Counting, resetting and advancing time {L80 - L90}
Every action or cost factor is marked and counted by its respective marker or memory variable. These values are all added to the values of the counters from the previous time-step. After all these actions are counted, the necessary markers are reset before the start of the next time-step. Finally time is advanced and a new time-step can be started.
Cost calculation {L91 - L95}
After the time-horizon has been reached, the final total costs can be calculated through multiplying all cost counters by their cost factors. The Expected Cost per part per time period over the simulation horizon is calculated by dividing the total costs by the time horizon multiplied by the number of parts. Finally the definitive estimate of the cost is calculated through dividing the cumulative Expected Cost by the number of Monte-Carlo iterations

4.4. Surrogate Modelling

Section 2 describes the selection of four possible surrogate modelling algorithms with which to optimise the result of the simulation. These four algorithms are here described in more detail before they can be thoroughly tested to find a best fitting one in chapter 5.

4.4.1. Randomsearch

Randomsearch is the most basic of the possible methods (Bergstra & Bengio, 2012). Randomsearch technically does not make use of an entire surrogate modelling algorithm. Every iteration, the methodology pics a random value for each of the decision variables from their respective ranges. It thereby assumes a uniform distribution over the entire search space for each of the decision variables. Rather than storing the values of the decision variables and the objective for each iteration, the method only stores these values for the best iteration so far.

4.4.2. HyperOpt

HyperOpt itself is a python based library created by Bergstra, Yamins, and Cox (2013a) which in itself gives options for utilising multiple different algorithms for optimisation of expensive functions. The algorithm most prominently featured by the creators as a well-performing one is the Tree-structured Parzen Estimator approach (TPE) (Bergstra, Bardenet, Bengio, & Kégl, 2011). This algorithm is compared to Randomsearch by Bergstra as: "The TPE algorithm is like taking Randomsearch and then slowly refining it to not choose values for hyper-parameters that are strongly correlated with terrible performance" (Enthought, 2013). While the algorithm in this case is applied directly to parameters of our simulation the same holds true. HyperOpt in this research therefore indicates the use of the HyperOpt library (Bergstra et al., 2013b) to optimise with the application of the TPE algorithm.

The TPE algorithm itself functions not through estimating a surrogate model to describe y, the objective value, as a result of x, the decision variable values; resulting in a probability of an objective value given a certain set of decision variable values $p(y|x)$. But by estimating the reverse (which decision variable values would be probable to fit a certain value of the objective): $p(x|y)$. It does so by taking the history of observed decision variable values and accompanying objective values and splitting it over an arbitrarily chosen desired performance y^* and estimating for each decision variable a probability density function for y. This results in:

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \tag{4.2}$$

The algorithm is then trained to try new parameters which are more likely to have a higher density in $g(x)$ and a lower density in $l(x)$. This is done through selecting new experiment parameters (the DOE) based on optimising the Expected Improvement (EI). The EI is defined as the expected difference between the desired performance and the actual performance given a certain value of the decision variable:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p(y|x) dy = \int_{-\infty}^{y^*} (y^* - y) \frac{p(x|y)p(y)}{p(x)} dy \tag{4.3}$$

The way in which y^* is determined is through the user choosing the ratio of the observations they would like to fall above or below the threshold. This ratio can be defined as $\gamma = p(y < y^*)$. In other words, say $\gamma = 0.8$ then 80% of observations would be determined to be less optimal than the desired performance. By implementing the expressions for γ, $l(x)$ and $g(x)$ into Eq.(4.3) one gets the final expression for EI (a more elaborate derivation is available in Appendix D):

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x) + (1-\gamma) g(x)} \quad \propto \quad \left(\gamma + \frac{g(x)}{l(x)} (1-\gamma) \right)^{-1} \tag{4.4}$$

Eq.(4.4) shows that in order to maximise EI, the algorithm will have to propose values of the decision variables x^* which have a higher likelihood in $l(x)$ than in $g(x)$. After each iteration the experiment is ran again with the new values of x^*, the results are taken into the history and used together with the previously ran experiments to develop new PDFs and new estimated optima.

4.4.3. IDONE

IDONE stands for Integer Data-based Online Nonlinear Extremum-seeker (Bliek et al., 2020a), which is an extension of the *DONE* algorithm (Bliek et al., 2016). IDONE, different from DONE, uses a linear combination of Rectified Linear Units (ReLU) as its surrogate. The surrogate model itself is indicated by $g(x)$ and consists of D basis functions.

$$\begin{aligned} g(\mathbf{x}) &= \sum_{k=1}^{D} c_k \max\{0, z_k(\mathbf{x})\} \\ z_k(\mathbf{x}) &= w_k^T \mathbf{x} + b_k \end{aligned} \tag{4.5}$$

Where $\mathbf{x} \in \mathbb{R}^d$ are the d decision variables. By fixing the parameters w_k and b_k, the model becomes linear in its parameters c_k. The specific choice of the basis functions makes this model have local minima exactly on its corner points (Bliek et al., 2020a). Each iteration, this model is fit to the available data. The available data consists of pairs of decision variable values and their corresponding objective value ((x_i, y_i) being the (decision variables, objective value) pairs for iteration i). The model fit is performed by solving the regularized linear least squares problem:

$$\min_{c_N} \sum_{n=1}^{N} (y_n - g(\mathbf{x}_n, c_N))^2 + \lambda \| c_N - c_0 \|_2^2 \tag{4.6}$$

To solve this problem the recursive least squares algorithm is used (Madisetti, 1997). This algorithm uses the optimum of the fitted surrogate as its DOE for the next iteration. This means that at each iteration the surrogate described in Eq.(4.5) is fit to the data using Eq.(4.6) and subsequently optimised using the BFGS method described by Wright and Nocedal (1999). This optimisation yields a single set of decision variable values $\mathbf{x}^*$ which is then used as the new experiment to run in order to produce new data. However, to avoid getting stuck in a local minimum, first a small alteration is added to the values: $\mathbf{x}_{N+1} = \mathbf{x}^* + \delta$ where $\delta \in \{-1, 0, 1\}$. This cycle is repeated for as many iterations as the user desires after which the last optimum calculated will form the suggested optimum of the original problem.

4.4.4. MVRSM

Where the *IDONE* algorithm was designed specifically to deal with integer valued decision variables, the Mixed-Variable ReLUbased Surrogate Modelling (MVRSM) algorithm is an adaptation of IDONE designed to deal with a mixture of integer and continuous variables (Bliek et al., 2020b). The adapted surrogate used in MVRSM distinguishes continuous ($\mathbf{x}_c$) and discrete ($\mathbf{x}_d$) decision variables:

$$\begin{aligned} g(\mathbf{x}_c, \mathbf{x}_d) &= \sum_{k=1}^{D} c_k \max\{0, z_k(\mathbf{x}_c, \mathbf{x}_d)\} \\ z_k(\mathbf{x}_c, \mathbf{x}_d) &= \begin{bmatrix} v_k^T & w_k^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_d \end{bmatrix} + b_k \end{aligned} \tag{4.7}$$

By then choosing v_k, w_k and b_k at the start and fixing them the model becomes linear in its parameters c_k again which lends it to be optimised using linear regression. The method of choosing these parameters influences the ability of the algorithm to deal with both continuous and integer valued decision variables by having any local minimum be located on intersections of

decision variables where the integer constraints are respected for those variables which require them and these constraints are relaxed for the continuous variables. The fitting of the surrogate to the data happens exactly in the same manner as in IDONE, using the recursive least squares algorithm. Also similar to IDONE, MVRSM uses the optimum of the fitted surrogate for its DOE for the experiments in the next iteration. However, the Limited-memory BFGS (L-BFGS) algorithm is used to optimise the fitted surrogate rather than the BFGS. The L-BFGS, as the name suggests, uses less memory because it assigns progressively more weight to iterations as it is running (Wright & Nocedal, 1999). The optimum of the surrogate found by the L-BFGS algorithm $(\mathbf{x}_c^*, \mathbf{x}_d^*)$ is then used to construct the seed for the experiment in the next iteration. This requires the final step of adding a deviation in order to avoid getting stuck in a local minimum through $(\mathbf{x}_c^{N+1}, \mathbf{x}_d^{N+1}) = (\mathbf{x}_c^*, \mathbf{x}_d^*) + (\delta_c, \delta_d)$ where $\delta_c \in \mathbb{R}^{d_c}$ and $\delta_d \in \mathbb{Z}^{d_d}$. Note here that the perturbations made to the optimal values are real values for the continuous variables and integer values for the discrete variables. Moreover this method leaves room for the user to choose a mix of possible values to use as perturbation which exceed the confines of those used in the IDONE algorithm. This allows MVRSM to use larger exploration parameters.

This chapter described the main contribution of this research, the development of a Monte Carlo simulation which simulates the expected costs of maintenance operations of parts in a railway infrastructure. This simulation includes a Condition Based Maintenance strategy with periodic inspections and an (s, S) inventory control strategy. Moreover this simulation includes options for minimal maintenance and extended options to prevent unnecessary inspections in the system.

Further this chapter described in more detail the four most suitable existing surrogate modelling algorithms which could be used to optimise the developed simulation. The next chapter reports on experimentation which serves the choice for a single most fitting surrogate modelling algorithm as well as experiments which increase understanding of the simulation and the complete model (simulation combined with an optimisation algorithm).

5

Numerical experiments

This chapter reports on the experimentation on elements of the developed model. First all experimental setup is discussed after which the results of those experiments is presented.

5.1. Experimentation strategy

5.1.1. Motivation

First, the simulation has to be accompanied by an optimisation algorithm to finish the whole model, to that end the four candidate algorithms which are proposed in chapter 4 are tested in *Experiment 1: algorithm choice* to see which is the best fit for the developed simulation. Once the model is completed by the identification of a suitable optimisation algorithm, the full workings of the model are displayed through *Experiment 2: test case*. This real world test case also yields practical insights regarding the use of the model. Finally, to provide more information to the user about the influence of some of the input parameters on the objective of the developed Monte Carlo simulation, *Experiment 3: simulation sensitivity analysis* is performed. The user could utilise this information to make decisions about the rigidity necessary for their data acquisition for the model.

5.1.2. Build of experiments

To perform all experiments, the simulation algorithm developed in chapter 4 is written in functional code using Python. This program is able to predict the expected cost for an input set of parameter values. The simulation is subsequently implemented in the EXPObench benchmark library (Bliek, Guijt, Karlsson, Verwer, & de Weerdt, 2021). This library, also written in Python, is able to take in problems with a determined set of parameter values and indicated decision variables such as the developed simulation. The library provides options for optimisation of the entered problem using different surrogate modelling algorithms, including the four algorithms studied in this chapter. All experiments are ran on a Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz with 8 GB of RAM system or comparable consumer-oriented systems.

5.2. Experimental setup

5.2.1. Experiment 1: Algorithm choice

After the considerations in section 2, *literature review*, four different algorithms for optimising the proposed simulation model through surrogate modelling were deemed potentially suitable. To test these algorithms an experiment is performed, the results of which are analysed based on three metrics: performance, speed and stability. These metrics are designed to score the algorithms on user-focused characteristics. Users are assumed to value firstly the final result: the objective value. The value of that objective should be as low as possible. The measure of *Performance* is therefore

used, here defined as the magnitude of the objective found by an algorithm in a given amount of time. All algorithms iterate their parameters in order to reach more and more accurate results. Through iteration these results (should) grow closer to optimal as the algorithm runs. An algorithm would then be considered to have a better performance when it reaches a solution of the underlying problem which is closer to optimal, given a certain amount of time, than another. To test the performance, algorithms are ran for a certain amount of time after which the progression of the best found result for each iteration can be compared between algorithms.

Secondly, the *Speed* of an algorithm may be of interest to a final user. While speed by itself is not fully indicative of an algorithm's superiority over others, when combined with the performance it can be of value in determining the best fitting algorithm to the studied problem. Speed is determined in two ways, firstly the time per iteration and secondly the time to reach a certain pre-set goal value for the objective. This pre-set goal value is determined at 150% of the average final objective value found in the *Performance* analysis. Both these measurements of speed can inform the user of which algorithm to choose in cases where computational time is limited and the best result is sought after in the shortest amount of time.

The final metric is then designed with the practical application of this model in mind. If a maintenance provider in railway infrastructure is looking for the best values of the three decision variables to use in their operations, they may want to make sure that these values are stable. Meaning that they do not vary much when the user runs the model two times using the same data. Otherwise, upon re-checking their results or re-evaluation of their operational strategy, they may risk a high likelihood of getting a different result from the same model. For this purpose *Stability* is used as a metric. The stability of an algorithm is defined in this research as being the variation in the definitive result of an algorithm. The nature of the studied problem and the method used for solving it includes the fact that the end result can never be guaranteed to be the actual optimum. Also, the inherent randomness in all algorithms subsequently potentially deliver different results when running the same algorithm twice for the same problem. This difference could be very slight or it could be very broad, and the magnitude of that difference is what here defines the stability of an algorithm. The stability of an algorithm is quantified by comparing the coefficients of variation of the results of each algorithm.

The actual experiment involves a single hypothetical part being ran through all four algorithms for 200 iterations. Preliminary tests showed little to no further improvement of the found objective after 200 iterations for all algorithms. To aptly compare results, the experiment is ran 10 times for each algorithm. The problem presented to the algorithms is a fictional part which has parameter values that are picked randomly from the ranges as conceived in section 5.2.3. Table 5.1 shows the parameter settings used for the algorithm choice experiments. To determine the amount of Monte Carlo repetitions to perform in the experiment Eq.(4.1) is used. The values for α and E are set to 0.25 and 15 respectively. Meaning that the result of the simulation would be expected not to deviate more than 15% from the mean in 75% of cases. This combined with the estimates for the mean and standard deviation as gathered from a preliminary set of 800 runs of the simulation results in a number of Monte Carlo iterations of 25. This number of Monte Carlo iterations is used for all runs in the algorithm choice experiments.

Table 5.1: Algorithm choice experiments parameter levels

C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	$f^F(x)$	$f^D(x)$	t_m	n
50	224	140	832140	5500	100.83	448	6	12	*Weibull(871.984, 23.406)*	*Gamma(1.667, 0.6)*	1690	30

5.2.2. Experiment 2: Real-life test case

With the best fitting algorithm identified, the model is complete. As a means of practical testing, the model is subsequently used to optimise the decision variable values for a specific railway infrastructure part. This part is the receiver of a track circuit sensor installation called a RUTA. A track circuit is used for train detection in railway systems. Its basic operation relies on sending an electrical current through a circuit which uses both rails of the track. When a train passes that particular piece of track its wheel bases short the circuit thereby signaling to the system that the track is occupied. The RUTA unit is the receiver in this system, it receives the current when a rail block is not occupied and receives no current in case the rail block is occupied. In both cases the RUTA unit can relay that information and subsequently all necessary measures in terms of audio-visual signaling as well as operational measures can be taken.

Figure 5.1: RUTA unit installed in the field

To run the developed model for this particular part, all parameter values are to be gathered as accurately as possible. All parameter values for this RUTA are gained from data provided by ProRail and its contractors. Information about the cost of inspection, replacement and maintenance actions are gathered by inquiring maintenance engineers about their work. The lead time and cost of the part were stored in supplier information available at ProRail, the holding costs are calculated as a percentage of the part costs. the downtime costs were calculated through averaging rail section values then the average downtime per failure is determined from failure data sets from ProRail. Finally the inspection interval is provided by the contractor's FMECA analysis. The final test data can be found in table 5.2.

Table 5.2: RUTA test case data

C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	$f^F(x)$	$f^D(x)$	t_m	n
7.50	12.50	12.50	389,367	-	122.82	146.25	10	24	Weibull(1306.416, 1.764)	Gamma(1.667, 0.6)	1774	1009

Parameter values

To serve confidentiality of pricing information, the details of the parameter values are redacted from this public version of this thesis. Further information can be requested through the author.

5.2.3. Experiment 3: Simulation sensitivity analysis

This sensitivity analysis serves to test the influence of nine parameters on the outcome (the expected cost EC) of the Monte Carlo simulation of the developed model. As this simulation is newly developed in this research, a sensitivity analysis can increase understanding of its functioning and the effects parameters have on the result of the simulation. The sensitivity analysis includes 9 of the input parameters (see section 4.2 for all parameters). These parameters are the six cost parameters (C_i, C_p, C_c, C_d, C_h, C_s, C_m), the lead time (t_s) and the inspection interval (T). The decision variables are not included because they do not have to be provided by the user in the final model and therefore would not yield useful information. The number of parts (n) is not included either because this is assumed to be clearly available to the user and would therefore require little extra information for the user to determine the input value for. The PDFs for the failure threshold and the step-wise deterioration ($f^F(x)$ & $f^D(x)$) are not analysed as parameters of influence but are included to give context to the information gathered. The PDFs together determine the demand of spare parts in the system, therefore a low- medium and high demand settings are constructed to see whether the influence of the nine parameters considered, changes over different demand settings. Finally the simulation horizon (t_m) is not included as a parameter to be analysed because it is the determinant of the accuracy of the model and can be adjusted by the user to

provide more accuracy at the cost of more computational resources. More testing of the model is advised to quantify the influence of the simulation horizon on this accuracy and speed of the model but these test are not included in this research. In this sensitivity analysis the simulation horizon parameter is set to twice the average lifetime of the parts considered.

The sensitivity analysis is performed using the 'One-at-a-time' (OAT) approach by Saltelli et al. (2008). This entails that for each separate selected parameter values are varied over their respective ranges. This yields information about the relative influence of these parameters on the simulation output (i.e. the estimated cost).

This is done through determining a realistic range for each of the 9 parameters and dividing these ranges over 10 values, these values form possible *levels* of those parameters to choose from. A collection of 1 level for each of the 9 parameters forms an experiment *scenario*. To this scenario, standard values for the excluded parameters are added to form a run of the simulation. Table 5.3 shows the determined levels for each of the included variables.

Table 5.3: Parameter levels

	C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T
Level 1	€10	€1	€1	€72,360	€10	€0.18	€2	1 month	3 months
Level 2	€20	€133	€131	€241,200	€11,120	€203.87	€267	2 months	11 months
Level 3	€30	€267	€261	€410,040	€22,230	€407.55	€533	3 months	18 months
Level 4	€40	€399	€389	€578,880	€33,340	€611.23	€798	5 months	26 months
Level 5	€50	€532	€521	€747,720	€44,450	€814.92	€1,063	6 months	34 months
Level 6	€60	€665	€650	€916,560	€55,560	€1,018.60	€1,330	7 months	41 months
Level 7	€70	€798	€780	€1,085,400	€66,670	€1,222.28	€1,595	8 months	49 months
Level 8	€80	€930	€910	€1,254,240	€77,780	€1,425.98	€1,860	10 months	57 months
Level 9	€90	€1,036	€1,040	€1,423,080	€88,890	€1,629.65	€2,127	11 months	64 months
Level 10	€100	€1,196	€1,170	€1,591,920	€100,000	€1,833.33	€2,392	12 months	72 months

These levels represent the range of values each parameter is likely to take in a real-world setting based on exploratory data gathering in the Dutch railway network. The standard values for the excluded parameters are based on the same data search:

The *Inspection cost* range is based on the same hourly rate as used in the test case, multiplied by two workers. he duration of an inspection could be very quick in case of a visual inspection of a well reachable part and in some cases somewhat more laborious in case of hard to reach parts which may include a lot of travel time. Therefore a duration range between 0,2 and 2 hours is included. This leads to a range of €10 to €100 for the C_i parameter. The *Preventive replacement cost* range is based on a data set containing failure and repair time data of 3818 repair operations from ProRail. With an observed minimum of 1 minute and a maximum of 1435, multiplied by the hourly rate of two engineers result in the range of €1 to €1,196 used for the C_p parameter. The *Corrective replacement cost* range is gained from the same data set as the preventive one. However the set is filtered on critical repair time which shows a minimum of 1 minute and a maximum of 1404. Leading to a range of €1 to €1,170 for the C_c parameter. The *Downtime cost* range, similar to the situation in the test case, utilises rail section values. The minimum rail section value is €1,500 per hour and the maximum is €33,000 per hour. With a probability of causing downtime upon failure of 0.067 based on a data set of 150,000 part failures this translates to a range of €72,360 to €1,591,920 per month for C_d. The *Spare part procurement cost* range is gained through the assumption that the most cheap spare parts can be in the category of single euros (nuts and bolts etc.) and the most complicated spare parts can be over €100,000 (like complex rail switch installations (Nissen, 2009)). The range used for C_s then is €1 to €100,000. A *Holding cost* range is calculated through assuming it as a percentage of the procurement cost. Based on Berling (2008) this percentage is set to 22% yielding a range of €0.18 to €2,392 per month for C_h. For the *Minimal maintenance cost* range the

preventive maintenance cost are doubled. This because a minimal maintenance task would require much more labour than a regular replacement would. The range used is €2 to €2,392 for C_m.

For the *Lead time* range the low end will be represented by the lowest value possible, being 1 period. The high end of the range is formed by the term which is usually allotted for producing specialty units like complex switches or structural elements of bridges. The range results in 1 month to 12 months for t_s. The *Inspection intervals* range is based on information from the contractors as given in an FMECA analysis. The observed minimum in these analyses is 3 months and the maximum 72 months, which constitutes the range for T.

The *Number of parts* is considered not to be influential on the workings of the model and is determined by practical considerations regarding test time available as well as computational resources. It is set to 30 parts. The *Simulation horizon* similarly is considered not to be influential and is based on the lifetime of the parts. In order to incorporate a full cycle of a part's lifetime in the simulation the simulation horizon is set to twice the estimated average lifetime, 1690 months.

All tests are performed for three demand settings: low, medium and high demand. This demand is determined by *Probability density functions* for the failure threshold and the step-wise deterioration. The values for these distributions are based on a data set describing demand for spare parts of a track circuit installation over a period of 15 years. Following L. Wang et al. (2009) the PDFs are modelled using a Weibull distribution for the failure threshold and a Gamma distribution for the step-wise deterioration. Because the Gamma and Weibull distribution together determine the average lifetime of a part one of the two can be chosen arbitrarily. For ease of use and to reduce noise in the model the Gamma distribution is chosen to result in an average of 1 unit of deterioration per time-step, i.e. a shape parameter of 1.667 and a scale parameter of 0.6. This means that the Weibull distribution should represent the lifetime in periods (here months). This is done using the Maximum Likelihood Estimators (MLEs). These MLEs resulted in a shape parameter of 23.406 and a scale parameter of 871.98. These values will be used as the medium demand scenario. The low and high demand scenarios will be formed by respectively doubling and halving the lifetimes of the parts used for the MLE estimation of the medium scenario.

The *Number of Monte Carlo iterations* is calculated according to the method as described in section 4:

$$n = \left(\frac{100 z_{\alpha/2} S_x}{E \bar{x}} \right)^2 \tag{5.1}$$

To serve a high level of accuracy for the sensitivity analysis the levels of α and E are set to 0.01 and 7.5 respectively. The sample mean and standard deviation are gathered from a set of 800 individual runs which resulted in $\bar{x} = 52.742$ and $S_x = 34.647$. These runs are done using the median scenario in table 5.3. These cause the number of iterations (rounded up to the nearest integer) to be 510 for the sensitivity analyses.

All experiment setup which was ran for Experiment 1: Algorithm choice and Experiment 3: Sensitivity analysis are indicated in Appendix C.

5.3. Results

The previous section detailed the experimental setup of three different experiments. This chapter describes the results of each of these experiments.

5.3.1. Experiment 1: Algorithm choice

As described in section 5.2.1, the algorithms are scored on their *performance*, *speed* and *stability*. Performance being defined as the magnitude of the objective found by an algorithm in a given computation budget (here 200 runs). Speed is defined by both the average iteration time and the

time an algorithm needs to find an objective below a certain threshold (here 1.5 times the average objective found after 200 iterations). Stability is defined by the variation in the definitive result of an algorithm. The result of the algorithm choice experiment is reported in table 5.4.

Table 5.4: Algorithm choice experiment results

Run	Algorithm	Decision variables [S, s, L_p]	Best Objective value
1	Randomsearch	[6, 2, 670]	39.578
2	Randomsearch	[10, 1, 410]	47.095
3	Randomsearch	[8, 3, 646]	43.334
4	Randomsearch	[13, 1, 409]	54.571
5	Randomsearch	[9, 1, 477]	41.923
6	Randomsearch	[6, 2, 590]	41.839
7	Randomsearch	[15, 0, 596]	46.523
8	Randomsearch	[11, 2, 676]	49.922
9	Randomsearch	[31, 30, 134]	53.517
10	Randomsearch	[9, 4, 527]	52.856

Run	Algorithm	Decision variables [S, s, L_p]	Best Objective value
1	HyperOpt	[7, 1, 596]	36.092
2	HyperOpt	[6, 1, 674]	41.524
3	HyperOpt	[5, 0, 656]	32.963
4	HyperOpt	[12, 0, 614]	41.689
5	HyperOpt	[9, 0, 652]	38.467
6	HyperOpt	[5, 0, 594]	35.847
7	HyperOpt	[4, 0, 632]	38.623
8	HyperOpt	[7, 1, 644]	38.644
9	HyperOpt	[8, 2, 648]	39.730
10	HyperOpt	[4, 1, 617]	38.418

Run	Algorithm	Decision variables [S, s, L_p]	Best Objective value
1	IDONE	[6, 0, 434]	50.161
2	IDONE	[15, 0, 323]	51.634
3	IDONE	[30, 0, 415]	58.633
4	IDONE	[12, 2, 186]	72.487
5	IDONE	[9, 3, 623]	44.123
6	IDONE	[8, 0, 627]	33.746
7	IDONE	[11, 4, 659]	51.348
8	IDONE	[7, 1, 480]	38.680
9	IDONE	[13, 0, 389]	53.921
10	IDONE	[6, 0, 697]	32.962

Run	Algorithm	Decision variables [S, s, L_p]	Best Objective value
1	MVRSM	[6, 0, 624]	32.257
2	MVRSM	[7, 0, 350]	43.395
3	MVRSM	[7, 0, 488]	36.405
4	MVRSM	[8, 0, 492]	37.880
5	MVRSM	[4, 0, 659]	31.261
6	MVRSM	[15, 0, 348]	49.835
7	MVRSM	[7, 1, 602]	34.815
8	MVRSM	[7, 0, 669]	32.666
9	MVRSM	[10, 0, 514]	39.780
10	MVRSM	[4, 0, 690]	35.903

Each row shows the results of a single run for each algorithm for the scenario described in section 5.2.1. The result, estimated best values for all three decision variables, found in each run are shown in the third column and the accompanying objective value, estimated costs per part per period, is shown in the fourth column. The results in table 5.4 show the variety of best found decision variable values between algorithms and between different runs of the same algorithm. The objective values found by Randomsearch for instance vary between 41.839 and 54.571 and the objective values found by MVRSM vary between 31.261 and 49.835. Moreover the results show a clear preference of all algorithms for a very low reorder level *s*. With the exception of the ninth run of Randomsearch. To get a better view of the differences between the algorithms, averages of the results are displayed in table 5.5. The averages over the 10 runs for each algorithm are shown for each decision variable as well as for the objective. These averages show that the HyperOpt and MVRSM algorithms find a more favourable result of the objective than the Randomsearch and IDONE algorithms. The two algorithms that show the highest average level of the objective (Randomsearch and IDONE) have in common that their average reorder level and order up-to level is higher than that of the two algorithms with the lowest average level of the objective (HyperOpt and MVRSM).

Table 5.5: Averages of experiment results

Algorithm	S	s	L_p	Objective
Randomsearch	7.2	4.6	513.50	47.12
HyperOpt	6.1	0.6	632.7	38.20
IDONE	10.7	1.0	483.3	48.77
MVRSM	7.4	0.1	543.6	37.42

This all indicates that the HyperOpt and MVRSM algorithms score better on the *performance* metric than the Randomsearch and IDONE algorithms in these tests.

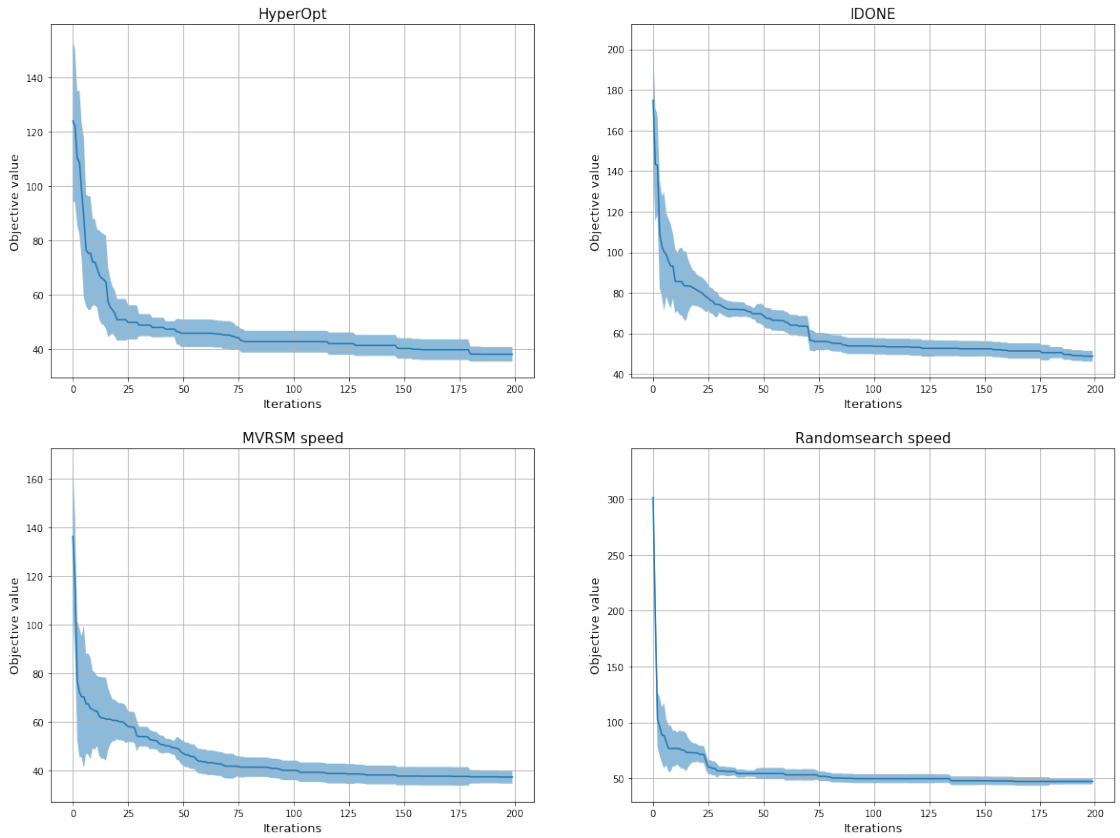

Figure 5.2: Performance runs averaged over the 10 repetitions of the experiment with a shaded area indicating one standard deviation of the results above and below the average.

A visual representation of the progression of the best objective value found by each algorithm over the 200 runs is shown in figure 5.2.

To determine whether these results actually significantly differ between algorithms a pair-wise t-test is performed between all algorithms over their best objective values found in all 10 runs. The significance results are shown in table 5.6.

Table 5.6: Pair-wise t-test significance values

	Randomsearch	HyperOpt	IDONE
HyperOpt	0.0001	-	-
IDONE	0.6709	0.0184	-
MVRSM	0.0034	0.6986	0.0294

Table 5.6 shows that a significant difference ($p < 0.05$) can be found between the pairs of HyperOpt - Randomsearch and HyperOpt - IDONE as well as MVRSM - Randomsearch and MVRSM - IDONE. But no significant difference is found between the algorithm pairs of HyperOpt - MVRSM and Randomsearch - IDONE. This further represents the dichotomy seen in the average objective values in table 5.5.

To score the algorithms on the *Speed* metric, Figure 5.3 shows a visual representation of the iteration time (in seconds) per iteration, averaged over the 10 repetitions of the experiment per algorithm. The average iteration length for each algorithm is noted in the top right corner of each graph. The figure clearly shows that the IDONE and MVRSM algorithms both have a very low iteration time for the first three iterations. This is the result of the algorithms requiring a base set of data to start actually fitting the surrogate. IDONE and MVRSM therefore perform random picks of the decision variables for the first three iterations which results in very low iteration times. After these first iterations the iteration times of IDONE and MVRSM increase to values beyond those of the HyperOpt or Randomsearch algorithms. This results in average iteration times for IDONE and MVRSM of 13.87 s and 13.61 s respectively which are both significantly higher than the 10.72 s and 10.87 s average iteration time for HyperOpt and Randomsearch respectively.

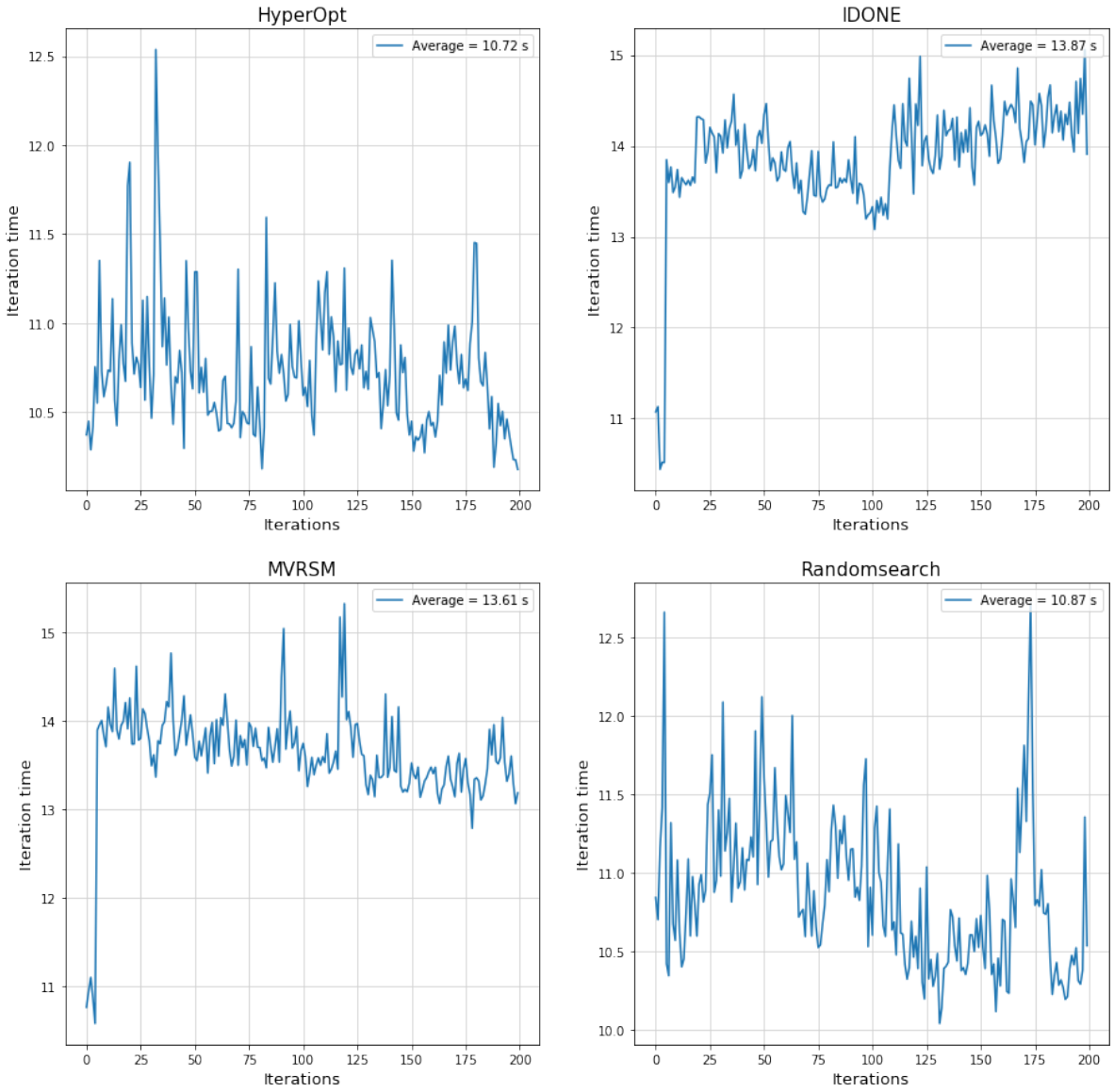

Figure 5.3: Iteration time per iteration averaged for all 10 repetitions of the experiment

The other method of determining *speed* was to determine the time required for each algorithm to find a certain objective. This objective was determined at 1.5 times the average result found after 200 iterations. The results in table 5.5 show that this average result is 42.88 €/part/period. Which

would make the objective to test for speed 1.5 * 42.88 = 64.32 €/part/period.

Table 5.7: Time per experiment repetition to reach the goal objective of 64.32 €/part/period for all algorithms in minutes

Experiment repetition	HyperOpt	IDONE	MVRSM	Randomsearch
1	3.22	15.67	6.25	5.47
2	0.97	17.85	0.85	2.61
3	3.16	1.35	9.89	4.80
4	1.33	-	10.29	5.08
5	4.56	3.10	3.20	7.34
6	2.15	5.25	9.04	4.91
7	5.08	9.75	0.33	4.25
8	2.15	7.50	0.67	1.34
9	1.33	7.49	1.27	12.82
10	2.81	2.15	3.84	11.92
Average	2.68	7.79	4.56	6.05

Table 5.7 details the time each algorithm took to reach the objective of 64.32 in minutes for each of the 10 experiment repetitions as well as the average over these 10 repetitions. The IDONE algorithm was not able to find an objective below 64.32 within the 200 iterations for which it was run. The average for IDONE therefore is taken over the other nine repetitions but should be interpreted as a skewed representation of the "real" average for this algorithm which is likely to be slightly higher. Further, the averages show that in general terms HyperOpt is the quickest algorithm to reach the goal objective with an average of 2.68 minutes.

Stability of an algorithm is defined in this research as being the variation in the definitive result of an algorithm. The nature of the studied problem and the method used for solving it includes the fact that the end result can never be guaranteed to be the actual optimum. Also, the inherent randomness in all algorithms subsequently potentially deliver different results when running the same algorithm twice for the same problem. This difference could be very slight or could be larger depending on the algorithm and the problem under consideration.

Table 5.8: Coefficients of variation of experiment results

Algorithm	**CV**(S)	**CV**(s)	**CV**(L_p)	**Cumulative CV**(*DV's*)		**CV**(*Obj*)
Randomsearch	0.55	1.86	0.31	2.72		0.11
HyperOpt	0.40	1.11	0.04	1.55		0.07
IDONE	0.66	1.41	0.33	2.4		0.23
MVRSM	0.41	3.00	0.22	3.63		0.14

The variations of the found decision variables and the objective value are represented in table 5.8 by their respective coefficients of variation (CV). Table 5.8 also shows the cumulative coefficient of variation over all decision variables. This reveals HyperOpt to have the lowest CV for the found objective value. Moreover the sum of the CV's for the decision variables is lowest for HyperOpt as well (1.55), followed by IDONE (2.40) then Randomsearch (2.72) and the largest CV for the decision variables is produced by MVRSM (3.65). This means that HyperOpt can in this test be regarded as the most stable algorithm of the four.

5.3.2. Experiment 2: Real-life test case

The result of the 200 runs of the HyperOpt algorithm with the parameter values as determined for the RUTA unit yielded the a result which estimated the optimum at a reorder level of 15 units, an order up-to level of 398 units and a preventive maintenance threshold of 140 units of deterioration. The objective value of the expected cost that accompany these decision variable values is €138.84 per part per period. Before finding this final result in the 155th iteration, the HyperOpt algorithm found 5 previous best objective values. This results in a graph of best results found which is rather simple as seen in figure 5.4.

Figure 5.4: Best Objective value found per iteration

To illustrate the search process the values for the objective found in each iteration are presented in figure 5.5. For visibility the 6 values with an objective higher than €1,000 are filtered out in this visualisation (iteration 3 - €108,431.90, iteration 125 - €100,779.70, iteration 192 - €30,276.22, iteration 38 - €11,009.18, iteration 35 - €8,942.75 and iteration 68 - €1,730.19). The algorithm can be seen to explore the full range of values between €526 and €139 in no apparent pattern, with an average of €314.13.

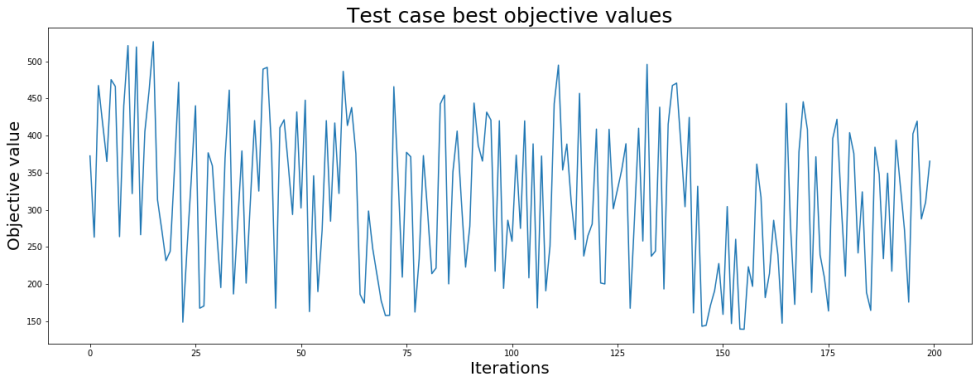

Figure 5.5: Objective value found per iteration

For further analysis a 4D plot is made of all values tested by the R-SIMS model. The colour gradient indicates the objective value. Figure 5.7 shows the different perspectives on the 4D visual.

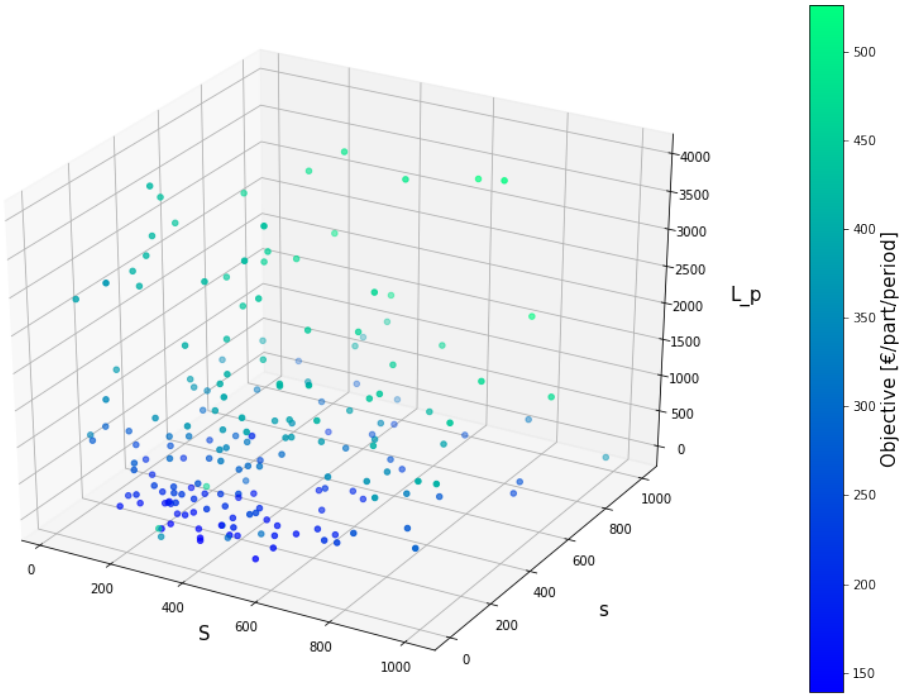

Figure 5.6: 4D plot of objective and decision variables found by R-SIMS model

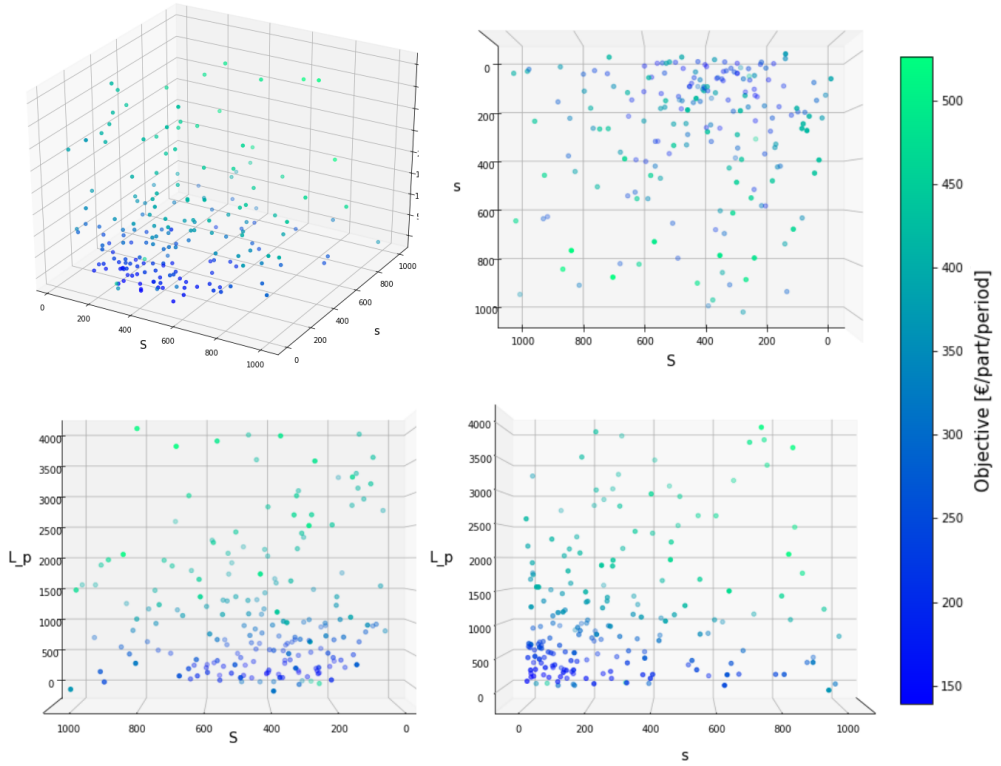

Figure 5.7: 4D plots of objective and decision variables found by R-SIMS model

5.3.3. Experiment 3: Sensitivity analysis

All analyses are performed using a full factorial design. Meaning that for each parameter all possible levels are tested while keeping the other parameters at a fixed level. These fixed levels are those of the median values. Meaning that in the value schema presented in table 5.3 the base scenario would have the following values: {C_i, €55 ; C_p, €598.50 ; C_c, €585.50 ; C_d, €832,140 ; C_s, €50,005 ; C_h, €916.76 ; C_m, €1,196.50 ; t_s, 7 ; T, 37}. The developed levels of parameters yields 90 single-parameter experiments to be performed per demand setting. Figure 5.8 shows the plots for all analyses for the medium demand setting.

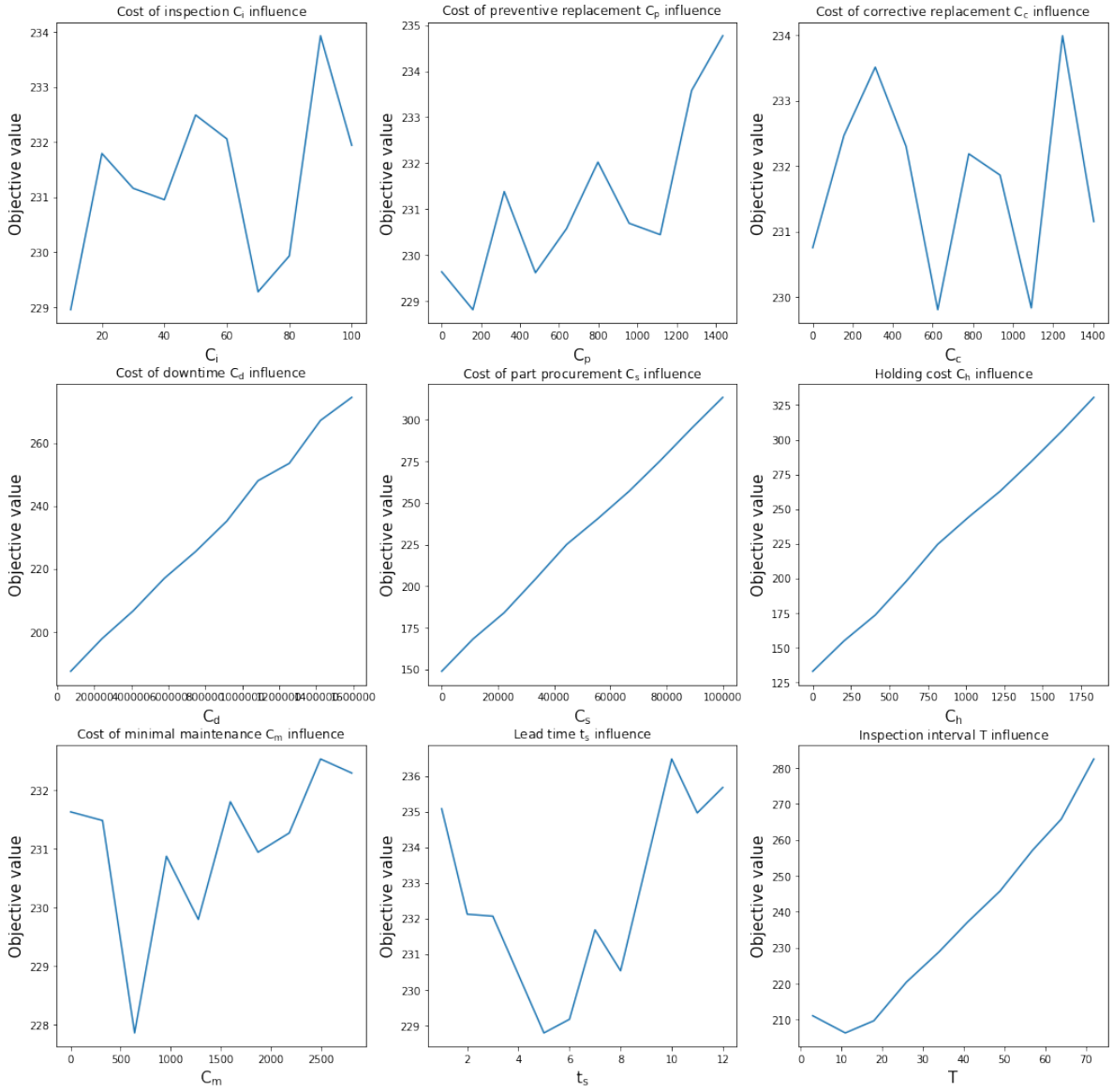

Figure 5.8: Plots of sensitivity analyses for the medium demand setting

The plots in figure 5.8 show a clear connection between the downtime cost, part cost and holding cost parameters and the objective. These show a linearly increasing connection. The inspection interval also resembles an early curve an later linear connection. Other parameters show little to no clear influence on the objective. Figures 5.9 and 5.10 show the same plots for the low and high demand settings. The cost parameters in both settings seem to produce a similar result to that of the medium demand setting. The lead time and inspection interval parameters however differ slightly.

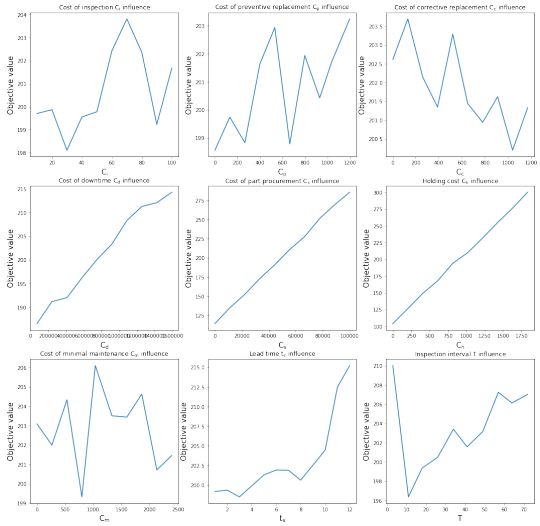

Figure 5.9: Plots of analyses for low demand setting

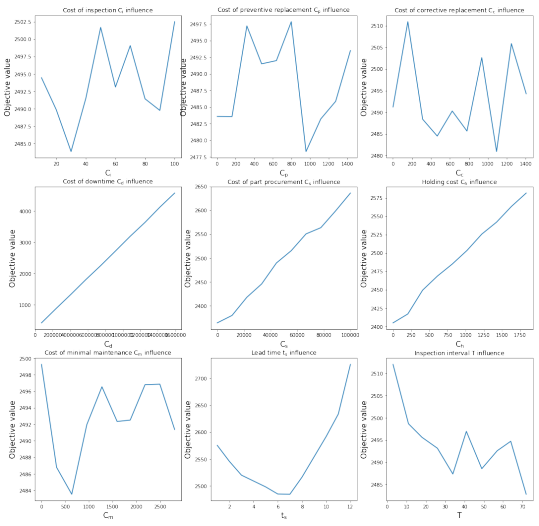

Figure 5.10: Plots of analyses for high demand setting

6

Discussion

This chapter serves as reflection on the research performed. It describes interpretations of the produced results, explaining what these results entail and what new questions may arise from them. The methodology used for the research in general and the experimentation in particular is critically reviewed.

6.1. Results

The produced results are separated into the *algorithm choice experiment* results, the results of the *test case* and the *sensitivity analysis* results.

6.1.1. Algorithm choice experiment

The three metrics that are developed to rate the applicability of the surrogate modelling algorithms in this research are *performance*, *speed* and *stability*.

Performance; the determination of performance in this context happened through examining the results gained from 10 runs of 200 iterations for each of the four algorithms. These results showed a clear distinction between two algorithms performing better than the two others. MVRSM and HyperOpt being the better performing algorithms in this case. This performance can be seen both in the graphical (figure 5.2) and in the numerical representation (table 5.5). The development of the objective over the iterations of the algorithms in figure 5.2 shows that even before the 10th iteration both MVRSM and HyperOpt have a better objective value which stays that way for the entire rest of the run. A further pair-wise t-test analysis of the speed results show that no significant difference is found between the pair of better performing algorithms, nor between the pair of worse performing algorithms. This indicates that from the speed-tests alone, no single algorithm can be advised to be best applicable to the simulation problem studied in this research.

Speed; two different analyses were performed to rank the four algorithms on speed: iteration time and time required for an algorithm to reach a certain goal in terms of an objective value. The iteration time for IDONE and MVRSM showed an odd but expected pattern (see figure 5.3). The first three iterations of both algorithms structurally had a significantly lower run time than the rest of the iterations. This could be explained by both algorithms using three random iterations before actually using the gathered data to estimate a surrogate. Further no clear patterns could be detected in the data. Averages showed that HyperOpt and Randomsearch both scored better than MVRSM and IDONE in terms of iteration time. The second analysis pertained the time required by each algorithm to reach the goal of 64.32 €/part/period for the objective. Again HyperOpt used the least amount of time on average to reach this goal, followed by MVRSM, then Randomsearch and finally IDONE which was even unable to reach the goal within the 200 iterations of the experiments in one

of the experiment repetitions. The two speed analyses indicated HyperOpt and Randomsearch as best algorithms in terms of the iteration time and HyperOpt as the algorithm also performing best in terms of time required to reach a preset objective goal. This would lead to the conclusion that when regarding *speed*, HyperOpt could be considered the best applicable algorithm.

Stability; to further specify the final choice for the best fitting optimisation algorithm, stability is tested. The stability of an algorithm is in this research determined by the variation in the outcome of the same runs as used in the speed tests. This variation is calculated for all decision variable outcomes as well as the resulting objective value. When taking into account all of those variations, in terms of coefficient of variation (σ/μ) the HyperOpt algorithm is clearly the most stable. Another interesting observation is that the MVRSM algorithm, which performed equally as well in the speed tests, is second to worst in terms of its stability over the objective and even the worst performing algorithm in terms of its stability over the decision variables.

The algorithm choice experiments were undertaken in order to motivate a choice for a single, best fitting, algorithm for optimising the studied simulation problem. Notably, the tests were only ran for a single problem seed (a single set of parameter values). This leads to uncertainty about the universality of the outcomes of these test. Yet, for the purposes of the current research, HyperOpt is deemed the best performing, and therefore best suitable optimisation algorithm for the Railway - Spare Inventory & Maintenance Scheduling model.

6.1.2. Test case

The test case results are a good representation of a real-world case. It predicts that the party responsible for maintaining this particular part would, from a financial perspective, be best off to adhere to a maintenance and spare part policy of holding 15 spare RUTA units as reorder level, ordering 398 units each time an order is placed and to preventively replace the units when their relative deterioration is at 140 units.

The use of the model, while functional, is not very well adjusted to the information sources available in this case. ProRail has a limited amount of data available regarding failure behaviours which leads to a lot of assumptions and generalisations. This leads to a case of 'garbage in - garbage out', meaning that the final result is only as reliable as the failure data input into the model. Moreover six different data sources were consulted to gather all the parameter values used for this test case (an overview of data sources used can be found in appendix B). This not only takes a lot of time and resources but could also lead to contradicting data or different parameter values that counteract one another. An advice would therefore be that any user would first spend time gathering the right data sources and integrating them into a single large data set which is reliable and straight-forward in its use for this model.

The performance of the R-SIMS model in this test case can be regarded as highly satisfactory. By exploring the search space in an efficient way it found the final result in 155 iterations. This may not always be the case due to the stochasticity inherent to both the simulation and the optimisation algorithm but still indicates that shorter run times could be used in practice than is done in this research.

6.1.3. Sensitivity analysis

The sensitivity analysis serves to inform users of the developed model about the influence of the nine studied parameters on the outcome of the newly developed Monte Carlo simulation. This can help focus data gathering efforts and thereby increase the added value of the model as a whole for the user.

To further analyse the influence of each of the studied parameters on the objective value, table 6.1 shows the Pearson and Spearman correlation coefficients calculated for each parameter's relationship with the objective found in the sensitivity analysis experiments. Coefficients are

calculated for all three demand settings.

Table 6.1: Pearson and Spearman coefficients of sensitivity analysis experiments

		C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T
Demand	**Coefficient**									
Low	Pearon	0.510	0.636	-0.710	0.992	1.000	1.000	-0.110	0.847	0.383
Low	Spearman	0.358	0.673	-0.745	1.000	1.000	1.000	-0.091	0.879	0.382
Medium	Pearon	0.362	0.787	-0.040	0.999	1.000	1.000	0.436	0.375	0.975
Medium	Spearman	0.394	0.721	-0.079	1.000	1.000	1.000	0.455	0.297	0.964
High	Pearon	0.388	0.004	0.009	1.000	0.997	0.998	0.237	0.571	-0.724
High	Spearman	0.248	0.018	0.018	1.000	1.000	1.000	0.152	0.394	-0.704

While reflecting on the results of the sensitivity analysis one should keep in mind that in these experiments only a single parameter is altered in value. This means that in the case of the results regarding the inspection cost parameter C_i, only that parameter was changed over the different runs, no other changes in for instance, the number of inspections performed or the number of parts have taken place. The only additional variation between experiments within a single demand setting is caused by the inherent stochasticity of the simulations which should, in part, be counteracted by the Monte Carlo repetitions.

Both the graphical (figures 5.8, 5.9 & 5.10) and the numerical (table 6.1) analysis show a clear influence of some parameters on the objective. The cost parameters for downtime, part procurement and holding have a clear linear correlation with the objective value. This holds for all three demand settings as well.

The cost of minimal maintenance on the other hand, seems to have little to no effect on the objective. Both correlation coefficients fall within the (-0.5, 0.5) range for all three demand settings. Similarly the cost of inspection has no clear correlation with the objective.

The costs for preventive- and corrective replacement differ over the demand settings. For the low setting both seem to have some effect. The preventive replacement cost have a slight positive correlation with the objective and the corrective replacement cost a slight negative one. Upon closer inspection of the objective values displayed in the plots in figure 5.9 however, both effects seem insignificant when compared to the effects of the three strongly correlated cost parameters. For comparison, C_c shows a maximum influence on the objective of roughly 5 €/part/period over its total range and the C_p a maximum influence of 3.5 €/part/period while the C_d, C_s and C_h parameters have an influence of roughly 25, 160 and 200 €/part/period over their respective ranges. For the medium demand scenario the cost of corrective replacement indicates no correlation. While the correlation between the preventive replacement and the objective is stronger than in the low demand setting, the same small effect relative to the effect of the three cost functions of downtime, procurement and holding can be seen.
Finally for the high demand setting, neither C_p or C_c show any influence on, or correlation with the objective.

The two time parameters, lead time t_s and inspection interval T show complex results.
The lead time, according to its correlation coefficients does have quite a strong correlation with the objective in the low demand setting. The total effect on the objective over its range (15 €/part/period) is comparable to that of the downtime cost (25 €/part/period). In the other demand settings this correlation is not present. The inspection interval has a very strong correlation in the medium demand setting. It is very comparable to the cost of downtime in both its correlation coefficients (>0.95) and its influence on the objective over its range (>70 €/part/period). This all indicates that the inspection interval has a strong positive correlation with the objective. The high demand setting however, shows a negative correlation with the objective value. Though both the

correlation coefficients and the value of its influence is less extreme than in the medium demand setting. In the low demand setting the inspection interval seems to have little effect at all.

Cost parameters; all of the observations regarding the cost parameter correlations can be linked simply to the magnitude of the parameter values used combined with logical reasoning on their occurrence (see table 5.1 for the values used).

When looking at the four cost parameters with the highest median value, three are also the three with the largest correlations and influence on the objective. The median values of C_d and C_s are the absolute highest of all cost parameter values. While the third, C_h is ranked fourth in terms of the value of its median, the value of its predecessor C_m is less impactful because costs of minimal maintenance are rarely incurred while holding costs are incurred every period. Note that the highest costs, those of downtime, occur in a similar frequency as minimal maintenance, yet the downtime cost are two orders of magnitude higher than those of minimal maintenance which could explain the difference in their effect on the objective.

C_i has the lowest median value by a large margin which explains its lack of observed influence on the objective.

C_p and C_c have similar magnitudes and also similar observed effects. The reason that these cost parameters seem to differ very much in their effect on the objective over different demand settings is likely to stem from their inherent connection to that demand. Because of the full-factorial design of the sensitivity analysis all other parameters were kept stationary while the parameter under study was varied in its value. When the demand then rises and all parameters stay the same, the logical expectation would be that the number of corrective maintenance actions necessary would increase.

Time parameters; the lead time and inspection interval showed more complicated relationships with the objective.
The only significant correlation seen between the lead time and the objective is presented in the low demand setting. One could theorise that this results from the decreased inventory on hand which would be present because of the longer lead time in that setting. If the demand is low a longer lead time is unlikely to cause any stock-outs, especially when compared to a medium or high demand setting. On the other hand it would cause less stock to be on hand and thus will decrease the holding cost. In higher demand settings this effect would thus be counter-acted by possible stock-outs and the resulting minimal maintenance costs and possible downtime costs.

The effects of the inspection interval are even more erratic than that of the lead time. The strong positive correlation in the medium demand setting clearly indicates that when the inspection interval becomes too large, costs increase. The implication here is that, when not enough inspections are performed, some parts may fail before they can be preventively serviced. This could then incur costs for preventive replacement, minimal repair or downtime. On the other hand, in the high demand setting the connection is negative. While the influence of the parameter over its range is small (approximately 28 €/part/period) and the correlation is less strong, it could mean that a turning point has been reached in terms of the demand. When demand gets to be too high, failures might become so abundant that inspection may have little to no effect any longer. Moreover, more failures cause more accompanying costs for downtime and maintenance which subsequently decrease the impact made by inspection costs or even by cost savings as a result of inspection causing the weaker correlation.

Overall results from the sensitivity analysis indicate that users of the developed model should take care in dividing their energy and attention over the different parameters when determining values to use in their rendition of the simulation. If only limited resources are available the advice

would be to first focus on accurately determining the values for the holding costs C_h, the part procurement costs C_s and the downtime costs C_d. Thereafter, depending on the demand setting to be studied, one should take care to accurately determine the costs of both preventive replacement C_p and corrective replacement C_c as well as the time parameters for lead time t_s and inspection interval T. Only if time and resources allow should one invest in determining accurate values for the cost of minimal maintenance C_m or the inspection costs C_i.

6.2. Methodology

The developed simulation is heavily based on previous research by L. Wang et al. (2009). Their model is the basis for the R-SIMS model developed here. The main difference between the models is the addition of the possibility for minimal maintenance. This mechanism allows a new choice in the model for the maintenance provider to not fully replace a part immediately but to repair it only to the "as-bad-as-old" condition. This means that in case of a stock-out of spare parts, downtime will not necessarily occur as a consequence. This mechanism directly affects the choices made for stock levels and replacement decisions as the risk of parts failing is slightly mediated in some cases.

Though, more alterations have been made to the base simulation:

- During initialisation of the model, all parts' deterioration level is distributed randomly beneath their individual failure thresholds. Where the base model assumed all parts on initialisation of the model had a deterioration level of 0 (as-good-as-new). The new method of initialisation better suits a system which is already in place an in which parts have not been installed all at the same time. This makes the developed model better suited to analyse currently installed components instead of components which are still to be installed.

- Each time-step, the R-SIMS model handles parts in a random order. As a result of the modeling techniques and language, each part is given an index to keep track of its characteristics during the modelling process. If then the part with index 1 is always handled before the part with index 2, this first part will have a higher chance of being serviced upon its failure than the second. After all, if in any time-step there is only a single spare part available and it is used to replace the failed instance of part 1, it will no longer be available to replace part 2. Especially in systems where parts are installed hundreds of times over this may have large effects on the distribution of maintenance operations and spare parts. Therefore the developed model picks the order of parts to handle randomly every time-step.

- The R-SIMS model implements the no-more-inspection clause to failed parts as well as parts which have been observed to exceed the CBM threshold. The base model stated only that this mechanic was invoked for parts that exceed the CBM threshold. However, after a part has been seen to have failed it will no longer benefit from further inspection, after all, it has already failed. By not inspecting failed parts some inspection costs can be saved which may influence the outcome of the decision variables.

However, one of the inaccuracies of the original model is also carried through to the R-SIMS model:

- Downtime is recorded in full time-steps. This is a result of the time granularity used in the modelling and is not in line with the real world. Especially when taking a large time-step (e.g. months/years) one would quickly over-estimate the downtime costs. While a part may fail, causing downtime in the beginning of a time-step, the entire time-step worth of time is counted as downtime and accompanying costs. A mediating effort would be to only count half a period worth of downtime, theoretically averaging out the actual moment the downtime occurred over the length of a time-step. Even more advanced would be to model

the start of downtime as a function of the condition of the part. While failure thresholds are stochastic, the model calculates this threshold upon installation of the part and therefore knows this beforehand. By assuming that the step-wise deterioration occurs linearly over the time-step, one would be able to estimate at what point during the time-step the part actually failed, causing the downtime.
This would however cause even higher computational costs at little extra accuracy.

Other practical aspects of the method used in this research could also be regarded as points of interest:

- The method of coding in this model is very sequential, each time period is handled after another and within each time-step, all parts are handled in order. This makes it very hard to increase computation speed especially at scale. The HyperOpt library does support some parallelisation through the use of MongoDB (Bergstra, Yamins, & Cox, 2013c) but this does not include parallelisation of the internal problem (here the simulation). A more critical review of the functional code may therefore be in order to develop a more efficient program thereby decreasing computational resources necessary.

- The method used in this research cuts off the optimisation algorithm at a certain number of iterations. While this is a standard used often in industry, a more sophisticated measure of stopping the algorithm may be applicable to any particular situation. In cases where multiple parts in a system have to be ran through the model, different speeds of finding a solution will occur. No method is right for every situation here. Every application calls for re-evaluation of which cut-off condition is used.

6.2.1. Experimentation

The experiments performed in this research were three-fold: an algorithm choice experiment, a test case and a sensitivity analysis. All these can also be generally regarded as practical use-cases of the model and revealed several shortcomings and difficulties in use:

- The algorithm choice experiments are only based on experiments ran using only one particular seed. This means that the best performing algorithm can only be presented as the best algorithm for that specific part. It could very well be the case that the HyperOpt algorithm, which was selected as the best fitting one, performs significantly worse on other seeds. For instance seeds which describe specific parts which have very high failure rates or specifically low downtime cost.

- Moreover the algorithms are only graded on their performance, speed and stability. In some cases an application may call for another metric or may find one of these more important than another. If one is not at all interested in the stability of a part (maybe from the perspective of the contractor this could be the case) but only interested in the performance metric of an algorithm one could better choose for the MVRSM algorithm than the HyperOpt one. Again, this highlights that any application requires a critical look at the outcomes of this research and their validity for that specific situation.

- The sensitivity analysis required actual real values to be assigned to the parameters in the model. This has proven to be difficult when data is not readily available. The levels used for the experimentation in this case were constructed by estimating a range (min-max) for each parameter based on real data. A shortcoming of this method is the lack of real data available, as demonstrated in the test-case experiment which showed the RUTA unit to have a cost of inspection which was lower than the lower bound of the range determined for that parameter. This range was subsequently divided into ten different levels uniformly. In truth

however, few parameters are actually likely to be distributed uniformly over its entire range in practice. For example, the full range of downtime costs are based on rail section values but not all rail section values occur with the same frequency. The rail sections with a value of 22 are very rare and the average rail section value most likely lies closer to 15 than 11. This does not have very large implications on the validity of the sensitivity analysis but if users would like more detailed knowledge of influence of several parameters in the most relevant regions, extra experiments are advised.

- One of the parameters not included in the sensitivity analysis was the simulation horizon. The reasoning behind excluding this parameter was that the simulation horizon increases accuracy as well as computational demands when it grows in magnitude. The implication is that a user should always adhere a simulation horizon which is as large as their computational resources allows. However, it might still be informative to quantify the relationships between the simulation horizon, the accuracy of the result of the simulation and the computational demands. As multiple decisions already have to be taken to match constraints of the use of the model, this information could facilitate better practical implementation of the simulation.

- Probability density functions have not been extensively studied in the experimentation phase of this research. All experiments use the method of fixing the step-wise deterioration at an average of 1 and subsequently using the MLE method to describe the lifetime in terms of time-steps with the failure threshold PDF. This methodology was a result of a lack of proper data available for the research. Yet in practice it is highly advisable to gather data both on daily/monthly deterioration and link it to a real-world metric (i.e. number of particulates in oil or resistance in electrical components). This would increase reliability of the model and give the user more freedom in choosing the time-step to fit their computational and accuracy needs.

7

Conclusion

In this chapter, the research questions are answered. The connection between these answers and the research goal is discussed. The impact is discussed which these answers and this research as a whole can have on the practice of railway maintenance management as well as the alignment wit theory in this field. Finally future research directions are also presented in line with the shortcomings and opportunities as described in chapter 6.

7.1. Research questions

This research encompassed a single research question and six sub-questions. First all six sub-questions are answered separately after which the central research question is answered.

SQ1: *What strategy of maintenance scheduling best suits the railway domain?*

The strategy of maintenance scheduling used in this research is Condition Based Maintenance with Periodical inspection. As described in section 2.2.1, railway infrastructure is highly dependent on a reliable maintenance scheduling strategy. Out of the strategies reviewed, condition based maintenance was the most advanced one. Within CBM a last choice can be made for continuous versus periodic review of the parts' condition. While the continuous review strategy yields even more reliable results, it is harder to implement in practice especially in large systems like the railways. Therefore, for this research, the periodic review strategy is deemed most suitable.

SQ2: *What strategy of spare parts inventory control best suits the railway domain?*

The governance of spare parts in railway infrastructure (and large systems like infrastructure in general) is outsourced. This entails a third party managing the actual warehousing operations. System and part owners do still hold responsibility over setting operational parameters like procurement timing and quantities but the reporting of stock levels and shipping of parts is outsourced to the third party. This immediately implicates that the dedicated function of this third party also includes highly sophisticated measurements of stock levels. In terms of the spare parts inventory control strategies reviewed in section 2 this translates to the two continuous review policies ((s,S) and (s,nQ)) being most applicable.

SQ3: *How can costs of spare inventory and maintenance scheduling be modelled?*

Costs are modelled using the simulation described in section 3 and 4. It based on L. Wang et al. (2009) but is altered to include possibilities for minimal maintenance to better reflect the real-world

operations going on in railway infrastructure maintenance management. This simulation is a Monte Carlo type simulation which takes into account stochastic step-wise deterioration of parts as well as a stochastic failure threshold for the deterioration of parts. It also accounts for downtime, preventive as well as corrective maintenance, inspection costs, holding costs and minimal maintenance costs.

SQ4: *How could decision variable in the spare inventory control strategy and the maintenance scheduling strategy be determined to minimise operational cost?*

The complexity of the simulation model causes a closed form objective function, the objective being the estimated costs, impossible to formulate. Optimisation therefore has to be derivative-free. Moreover the evaluation of this function is computationally costly. A particular method which is developed for optimisation expensive black-box functions is called Surrogate Modelling. Several surrogate modelling algorithms were reviewed (see chapter 2) and four were tested in chapter 5. The results of the numerical experiments showed that the algorithm used in the HyperOpt library, Tree-structured Parzen Estimators, is most suitable for this specific problem.

SQ5: *How could this type of model add value to railway infrastructure management?*

This model can be used by infrastructure managers and maintenance providers alike. Most directly in railway infrastructure but also in other infrastructure domains like highways or waterways.

Moreover it yields scientific/theoretical insights in the form of a new simulation model and a thorough comparison of surrogate modelling algorithms in a practical application.

Finally for the main research question:

RQ: How can spare inventory control decisions and maintenance scheduling decisions be modelled for the application in railway networks to minimise operational cost?

This can be done using the developed model.

The developed model calculates the estimated costs using a novel Monte Carlo simulation which simulates multiple identical parts installed in the system which share a common pool of spares. These parts are modelled as independent and their deterioration is stochastic. Moreover the simulation includes the option for minimal repair.

The simulation incorporates a condition based maintenance strategy with periodical inspection and a spare inventory control policy which uses continuous monitoring and an order up-to level (the (s,S) policy). This simulation is optimised through a surrogate modelling algorithm from the HyperOpt library and optimises the objective of estimated costs based on the decision variables of reorder level s, order up-to level S and CBM threshold L_p. The result of this model is the optimised values for these three decision variables and the accompanying value for the objective: the maintenance operations cost per part per period (€/part/period). Because of the stochasticity in the simulation as well as the surrogate modelling algorithm these results can not be guaranteed to be the absolute optimum. Yet, the results are deemed useful in practice and can contribute to a more resilient railway network through better maintenance management.
As stated in chapter 1, the goal of this thesis is to develop a model which optimises financial costs and returns the associated values for decision variables which govern maintenance scheduling and control of inventory of spare parts. This model should be specifically suited for application in railway systems.

The developed model satisfies this goal, it optimises estimated costs per part per period and returns values for the three decision variables which govern both spare inventory control (reorder level s and order up-to level S) and maintenance scheduling (CBM threshold L_p). By choosing

proper elements to build the simulation (e.g. the CBM and (s,S) policy and adding minimal maintenance) this model better suits the railway infrastructure context than the models available in literature (see table 2.1

7.2. Impact

Overall this research, as all scientific research, should be interpreted by users for their own specific demands and situations. No research can be unilaterally used in the same way. It should always be handled with care and criticism and never taken for granted as being the ground truth. Nonetheless, practitioners in the railway maintenance management field could use the developed model to inform some of their practices. Moreover this model and the testing thereof have some interesting similarities as well as disparities with existing literature and may even spur some future research endeavors.

7.2.1. Practice

The practical implications of this research are focused on the application in industry.

- Infrastructure managers can use the developed model to make decisions regarding their own spare inventory and maintenance scheduling. This is the most straight forward use of the R-SIMS model. The three decisions which are optimised can be implemented directly in a maintenance provider's operations and thereby hopefully improve their financial performance.
- In the case of ProRail this model can guide negotiations with both current and future contractors. In this case the R-SIMS model would not necessarily be used to directly influence operational decisions, but to gain insight into what decisions may be taken in a situation which the user (ProRail) has no direct access to. This would make the model function more like a predictive model than an optimisation.
- The simulation separately can be used as a starting point for evaluating ones own entire supply chain, by getting an estimate of the costs of spare inventory and repair/replacement actions one can support decision making throughout the procurement and operational process of an infrastructure system.

7.2.2. Theory

- The literature (see chapter 2) showed no previous combined model which describes maintenance scheduling as well as spare inventory control that also include minimal maintenance. This contribution is significant only in specific situations (moreover specified by the fact that periodic monitoring is assumed for the inspection strategy) but for those specific situations this model can yield a large added value when compared to models that do not include minimal maintenance.
- This research includes a relatively large comparison of several surrogate modeling algorithms. Since the field of surrogate modelling is rather young, all extra practical applications as well as comparison upon a new problem can increase overall understanding of these algorithms.

7.3. Future research directions

This research, while conclusive in its own right, revealed some more questions which have not been answered. These form opportunities for future research.

- More research could be done to gain insights into best practices when it comes to data selection. The data required to utilise operational models is often very detailed and

widespread throughout an organisation or over stakeholders. The aggregation of data sources and control of access is an entire field of data science which would be applicable to this research.

- The sensitivity analysis presented in chapter 5 only analyses the effects of parameter changes on the outcome of the Monte Carlo simulation. This analysis was performed only on this part of the model because it is the newly developed part. A full sensitivity analysis of the entire modelling framework including the optimisation algorithm would be highly advised as this would yield valuable insights into the actual effects of the parameter deviations on the final product of the model rather than only on the simulation.

- This research modelled only a single spare inventory control and maintenance scheduling strategy. Further development of the R-SIMS model presented here could include other strategies as well. Parts could then be included which have continuous monitoring capabilities. Because large systems like the railway infrastructure often involve different sets of parts. Some of these parts in practice have their own dedicated continuous monitoring systems which allow the owners to implement another maintenance scheduling strategy. Making the model more modular in such a way that the user could run these parts through as well would be a great increase in value for the model. Moreover, including parts which have an *(s,nQ)* inventory control strategy would be beneficial. Some part sets have a different inventory control strategy from the one used in the R-SIMS model. By again making the model more modular to accommodate for these strategies it would be more widely applicable in practice.

- Better methods for estimating downtime costs should be developed. In the developed model, because of the granularity of time it is very difficult to balance computational needs with an accurate estimate of downtime cost. Especially in infrastructure systems like the railways where downtime is extremely costly this factor is important. Further work on the implementation of the downtime costs would therefore be a great contribution to the R-SIMS model in particular.

- Possible expansion of the application outside the railway management domain. While one of the biggest contributions of this research is the development of a combined spare inventory & maintenance scheduling model specifically suited for railway infrastructure, the literature review showed that in general these models rarely include functionalities for minimal maintenance. This functionality is particularly valuable in the railway setting but could also be applied in specified other applications which for instance are hard to reach with spare parts. Research in this wider application would therefore increase the benefit industry would be able to gain through the use of the R-SIMS model.

References

Al-Turki, U., Duffuaa, S., & Bendaya, M. (2019). Trends in turnaround maintenance planning: literature review. *Journal of quality in maintenance engineering*, *25*(2), 253–271.

Baptista, R., & Poloczek, M. (2018). Bayesian optimization of combinatorial structures. In *International conference on machine learning* (pp. 462–471).

Basten, R., & van Houtum, G. (2014). System-oriented inventory models for spare parts. *Surveys in Operations Research and Management Science*, *19*(1), 34-55.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *25th annual conference on neural information processing systems (nips 2011)* (Vol. 24).

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2), 281–305.

Bergstra, J., Yamins, D., & Cox, D. (2013b). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning* (pp. 115–123).

Bergstra, J., Yamins, D., & Cox, D. D. (2013a). Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th python in science conference* (Vol. 13, p. 20).

Bergstra, J., Yamins, D., & Cox, D. D. (2013c). *Mongodb - hyperopt documentation.* http://hyperopt.github.io/hyperopt/scaleout/mongodb/.

Berling, P. (2008). Holding cost determination: An activity-based cost approach. *International Journal of Production Economics*, *112*(2), 829–840.

Bhosekar, A., & Ierapetritou, M. (2018). Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Computers & Chemical Engineering*, *108*, 250–267.

Bliek, L., Guijt, A., Karlsson, R., Verwer, S., & de Weerdt, M. (2021). EXPObench: Benchmarking surrogate-based optimisation algorithms on expensive black-box functions. *arXiv preprint arXiv:2106.04618*.

Bliek, L., Verstraete, H. R., Verhaegen, M., & Wahls, S. (2016). Online optimization with costly and noisy measurements using random fourier expansions. *IEEE transactions on neural networks and learning systems*, *29*(1), 167–182.

Bliek, L., Verwer, S., & de Weerdt, M. (2020a). Black-box combinatorial optimization using models with integer-valued minima. *Annals of Mathematics and Artificial Intelligence*, *89*, 1–15.

Bliek, L., Verwer, S., & de Weerdt, M. (2020b). Black-box mixed-variable optimisation using a surrogate model that satisfies integer constraints. *arXiv preprint arXiv:2006.04508*.

Deshwal, A., Belakaria, S., & Doppa, J. R. (2020). Scalable combinatorial bayesian optimization with tractable statistical models. *arXiv preprint arXiv:2008.08177*.

Donselaar, K. H., & Broekmeulen, R. A. (2017, March). *Lecture notes and toolbox for the course stochastic operations management 1cv20.* (BETA working paper 447)

Driels, M. R., & Shin, Y. S. (2004). *Determining the number of iterations for monte carlo simulations of weapon effectiveness* (Tech. Rep.). Monterey, CA: NAVAL POSTGRADUATE SCHOOL DEPT OF MECHANICAL AND ASTRONAUTICAL ENGINEERING.

Ebeling, C. E. (2004). *An introduction to reliability and maintainability engineering.* Tata McGraw-Hill Education.

Enthought. (2013, July). *Hyperopt: A python library for optimizing machine learning algorithms; scipy 2013.* https://www.youtube.com/watch?v=Mp1xnPfE4PY.

Hooke, R., & Jeeves, T. A. (1961). "direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, *8*(2), 212–229.

Hu, Q., Boylan, J., Chen, H., & Labib, A. (2018). Or in spare parts management: A review. *European Journal of Operational Research*, *266*(2), 395-414.

Huang, L., Chen, Y., Chen, S., & Jiang, H. (2012). Application of rcm analysis based predictive maintenance in nuclear power plants. In *2012 international conference on quality, reliability, risk, maintenance, and safety engineering* (pp. 1015–1021).

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2010). Sequential model-based optimization for general algorithm configuration (extended version). *Technical Report TR-2010–10, University of British Columbia, Computer Science, Tech. Rep.*.

Hwang, J., & Samat, H. (2019). A review on joint optimization of maintenance with production planning and spare part inventory management. In *Iop conference series: Materials science and engineering* (Vol. 530, pp. 12–48).

Jamshidi, A., Hajizadeh, S., Su, Z., Naeimi, M., Núnez, A., Dollevoet, R., ... Li, Z. (2018). A decision support approach for condition-based maintenance of rails based on big data analysis. *Transportation Research Part C: Emerging Technologies*, *95*, 185–206.

Jiménez, J., & Ginebra, J. (2017). pygpgo: Bayesian optimization for python. *Journal of Open Source Software*, *2*(19), 431.

Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, *21*(4), 345–383.

Keizer, M. C. O., Teunter, R. H., & Veldman, J. (2017). Joint condition-based maintenance and inventory optimization for systems with multiple components. *European Journal of Operational Research*, *257*(1), 209–222.

Lidén, T. (2015). Railway infrastructure maintenance-a survey of planning problems and conducted research. *Transportation Research Procedia*, *10*, 574–583.

Madisetti, V. (1997). *The digital signal processing handbook*. CRC press.

Mueller, J. (n.d.). *Optimizing expensive blackbox simulations*. Retrieved from https://cs.lbl.gov/assets/CSSSP-Slides/20180719-Mueller.pdf (Center for Computational Science and Engineering)

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The computer journal*, *7*(4), 308–313.

Nissen, A. (2009). Lcc analysis for switches and crossings: a case study from the swedish railway network. *International Journal of COMADEM*, *12*(2), 10–19.

Rausch, M., & Liao, H. (2010). Joint production and spare part inventory control strategy driven by condition based maintenance. *IEEE Transactions on Reliability*, *59*(3), 507–516.

Ru, B., Alvi, A., Nguyen, V., Osborne, M. A., & Roberts, S. (2020). Bayesian optimisation over multiple continuous and categorical inputs. In *International conference on machine learning* (pp. 8276–8285).

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., ... Tarantola, S. (2008). *Global sensitivity analysis: the primer*. John Wiley & Sons.

Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling* (Vol. 3). Wiley New York.

Ueno, T., Rhone, T. D., Hou, Z., Mizoguchi, T., & Tsuda, K. (2016). Combo: an efficient bayesian optimization library for materials science. *Materials discovery*, *4*, 18–21.

Van Horenbeek, A., Buré, J., Cattrysse, D., Pintelon, L., & Vansteenwegen, P. (2013). Joint maintenance and inventory optimization systems: A review. *International Journal of Production Economics*, *143*(2), 499–508.

Van Horenbeek, A., & Pintelon, L. (2015). A joint predictive maintenance and inventory policy. In *Engineering asset management-systems, professional practices and certification* (pp. 387–399).

Springer.

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European journal of operational research*, *139*(3), 469–489.

Wang, J., & Zhu, X. (2021). Joint optimization of condition-based maintenance and inventory control for a k-out-of-n: F system of multi-state degrading components. *European Journal of Operational Research*, *290*(2), 514–529.

Wang, L., Chu, J., & Mao, W. (2008). An optimum condition-based replacement and spare provisioning policy based on markov chains. *Journal of Quality in Maintenance Engineering*, *14*(4), 387–401.

Wang, L., Chu, J., & Mao, W. (2009). A condition-based replacement and spare provisioning policy for deteriorating systems with uncertain deterioration to failure. *European Journal of Operational Research*, *194*(1), 184–205.

Wang, W. (2012). A stochastic model for joint spare parts inventory and planned maintenance optimisation. *European Journal of Operational Research*, *216*(1), 127–139.

Wright, S., & Nocedal, J. (1999). Numerical optimization. *Springer Science*, *35*(67-68), 7.

Xie, J., & Wang, H. (2008). Joint optimization of condition-based preventive maintenance and spare ordering policy. In *2008 4th international conference on wireless communications, networking and mobile computing* (pp. 1–5).

Zhang, X., Liao, H., Zeng, J., Shi, G., & Zhao, B. (2021). Optimal condition-based opportunistic maintenance and spare parts provisioning for a two-unit system using a state space partitioning approach. *Reliability Engineering and System Safety*, *209*, 1–24.

A

Scientific article

Railway - Spare Inventory & Maintenance Scheduling model development using Monte-Carlo simulation and Surrogate Modelling

D. Sommers BSc.
Faculty of Civil Engineering and Geosciences
Delft University of Technology
Delft, The Netherlands

Dr. N. Bešinović
Faculty of Civil Engineering and Geosciences
Delft University of Technology
Delft, The Netherlands

Dr. S. Fazi
Faculty of Technology, Policy and Management
Delft University of Technology
Delft, The Netherlands

Drs. M. Grashoff
Supply Chain Management, Procurement
ProRail B.V.
Utrecht, The Netherlands

Prof. dr. R.M.P. Goverde
Faculty of Civil Engineering and Geosciences
Delft University of Technology
Delft, The Netherlands

Abstract*—To better govern inventory control of spare parts in tandem with scheduling of maintenance operations in a railway setting, a Railway - Spare Inventory & Maintenance Scheduling (R-SIMS) model is developed. To suit the railway industry setting the model includes an inventory control strategy which envokes continuous monitoring and a reorder level (s) as well as an order up-to level (S). Moreover a condition based maintenance (CBM) strategy is used with periodical inspections, again to fit the railway domain. These strategies are combined in a simulation model which assumes stochastic step-wise deterioration of parts as well as stochastic lifetime lengths to predict the expected cost per part per period. The three maintenance scheduling and spare inventory decisions to be optimised then are 1) when to replace parts based on their condition, 2) when to buy new spare parts and 3) how many spare parts to procure at each order. These decisions are represented by values of the three decision variables: the CBM threshold L_p, the reorder level s and the order up-to level S. The optimisation of the simulation happens through surrogate modelling. A Python library called HyperOpt is used with a Tree-structured Parzen Estimator algorithm to optimise the three decision variables and minimise the estimated cost. This research contributes to the existing theory by creating a modelling framework for the joint optimisation of spare inventory and maintenance scheduling decisions which is specifically tailored to the railway context and includes possibilities for minimal maintenance. In practice this model can be used by maintenance providers to increase the financial success of their maintenance operations and by infrastructure managers to increase insight into their maintenance providers' operations.

***Index Terms*—Spare inventory control, Maintenance scheduling, Surrogate modelling, Simulation, Operations research, Railway, Optimisation**

1. Introduction

Planning of maintenance operations is hard. Especially in large systems this becomes highly complex quickly.

To aid in governing the maintenance process, mathematical models have proven very effective. Statistical methodologies have been developed and applied to analyse failure of components in technical systems but also to forecast other stochastic processes like lead times of spare parts or the success of a repair action. These models to aid in decision making for maintenance operations have mainly been focused on production operations or single machines. For large and interconnected systems like the railway network operational decisions in maintenance work differently. If one regards a railway network as a production facility they might overestimate the level component interactions, after all the components in a railway network are far more dispersed. Or if the same maintenance rules are applied to railways as there are to a car engine the possibilities for improvised or minimal repairs may get overlooked.

This research addresses how spare inventory and maintenance scheduling decisions can be optimised from a financial point of view for the application in railway networks.

The specific relevant decisions in maintenance management of railway networks being optimised are: when to replace parts based on their condition, when to buy spare parts and how many spare parts to buy. These decisions often have far-reaching influence on the proper functioning of a system and a model to support these decisions which in this research is specifically designed for the railway domain could prove very beneficial to the industry. An example of a use case of this model would be that of a specific type of railway switch. If one would consider the Dutch railway network in which this type of switch would be installed 200 times, spread over the system. A user of the developed model would then be able to

enter data regarding that type of switch; cost of procurement of the part, cost of replacement operations but also data on its failure frequencies and probability of downtime when the part fails unexpectedly. The model would then process this data and suggest the estimated financially best values for the three decisions: when to replace these types of switches based on their condition, when to buy spare switches and how many spare switches to buy each time.

The contribution of this research is the development of a Monte Carlo simulation which simulates the expected costs of maintenance operations of parts in a railway infrastructure. This simulation includes a Condition Based Maintenance strategy with periodic inspections and an (s, S) inventory control strategy. Moreover this simulation includes options for minimal maintenance and extended options to prevent unnecessary inspections in the system.

Further a modelling framework is developed to optimise financial costs as predicted by the simulation and return the associated values for decision variables which govern maintenance scheduling and control of inventory of spare parts. This optimisation is performed using surrogate modelling.

The model calculates these optimal values of the decision variables for multiple identical parts installed in the system which share a common pool of spares. These parts are regarded as independent, meaning that they do not influence each others' failure behaviours and their failure should is based on stochastic deterioration rather than time. The model takes into account costs for maintenance operations, spare part warehousing and procurement.

This paper details first the literature research to find a base model for the simulation development and to find the potential optimisation algorithms which may be applicable. Thereafter the problem definition and methodology describe the development of the simulation. To determine the most suitable optimisation algorithm numerical experiments are performed. Also the a real world test-case is analysed to demonstrate the full modelling framework in action. Finally conclusions are drawn regarding the main outcomes and recommendations for further research are described.

2. LITERATURE

To model both spare inventory control and maintenance scheduling, overarching strategies are chosen to fit the problem situation. For the sake of maintenance scheduling, a highly advanced level of scheduling should be adhered to. The railway network after all has a high demand of reliability. The most advanced methods of maintenance scheduling strategies in the typology by [1] are the *condition based maintenance* (CBM) strategies. In which parts are preventively replaced whenever possible. This involves checking the parts' condition and replacing that part when its deterioration exceeds a threshold L_p which is based on knowledge about its failure processes (preventive in-place repair is not considered as an option as a result of the reality of the railway network operations). The inspections of the condition of a part are in this research considered to happen periodically, while some parts in the railway system do have sensor technology allowing for continuous data on their condition, these are very rare [2] and have less need for the developed model.

Spare inventory control can be steered through strategies that specify how the stock levels are reviewed as well as how to buy new parts. A comprehensive categorisation is developed by [3]. The review regime can be either periodical, checking the stock levels every R periods, or continuous, updating the stock levels every time a part is added or used. The procurement of spare parts can then take the form of an *(s,nQ)* system where n units of quantity Q are ordered each time the stock level falls below s such that the stock level is restored to above s. Alternatively the *(s,S)* system can be used, where every time the stock level drops below s enough spare parts are ordered to replenish the stock level back to S. Warehousing of spare parts in large systems like a railway network is often outsourced to specialised third-party operators. These operators use advanced dedicated systems to keep track of their stock levels. Therefore, continuous review of stock levels is the most applicable method of governing spare inventory for the developed model.

To fulfill the research goal, the developed model should thus optimise L_p, s & S to suit the minimisation of the operational cost. For the operational modelling a base model is sought in literature. When considering the joint spare inventory and maintenance scheduling modelling literature a few requirements are kept in mind for a base model to be applicable to the studied situation: as stated above, the base model should contain a condition based maintenance strategy with periodic inspections and a spare inventory control regime described by [3] as *(s,S)* or *(s,nQ)*. Moreover the a potential base model should be applicable to *Multi-unit systems*, meaning that it considers more than one single installed part or machine. Finally the model should consider components to not be inter-dependent. While components influence each other heavily in smaller systems or machines, the railway network is of such scale that this inter-dependence is negligible and the model should represent that.

[6] developed such a model. They modelled deterioration of parts using Markov-Chains, allowing for modelling different states of deterioration of parts which also creates opportunities for the user to assign characteristics to each state. This however requires quite a lot of data to use. [7] use a very interesting approach to their CBM strategy. Their model includes the replacement of parts based on their condition but also has a preventive maintenance threshold based on operational time. While this strategy may be less prone to unexpected failure, it is too complex for the railway domain. [8] use a continuous deterioration process which is stochastic. This method closely resembles the reality while requiring little data from the user. Their (s,S) inventory strategy allows only for a maximum of a single order to be outstanding at all times. It also only assumes perfect instantaneous repairs or replacements and no minimal repairs. [5] also used stochastic step-wise deterioration but

TABLE I
LITERATURE REVIEW TABLE *: [4] USED A CONTINUOUS-REVIEW CBM POLICY, THEREFORE EACH FAILURE IS NOTICED IMMEDIATELY WITHOUT HAVING TO BE ANNOUNCED BETWEEN INSPECTION PERIODS. **: [5] USE A SPECIAL FORM OF THE (S,S) POLICY WHERE A VARIABLE ORDER UP-TO LEVEL IS USED WHICH DEPENDS ON THE SYSTEM CONDITION.

Authors	Staged deterioration	Self-announced failure	(s,S) or (s,nQ) Inventory policy	Markov Decision Process	Component inter-dependence	Minimal repairs
L. Wang, Chu and Mao (2008)	x		x			
Xie and Wang (2008)			x			
J. Wang and Zhu (2021)	x		**	x	x	
Keizer et al. (2017)	x	*	x	x		
Van Horenbeek and Pintelon (2015)			x		x	
L. Wang, Chu, and Mao (2009)		x	x			
W. Wang (2012)		x				
This paper		x	x			x

discretised parts' deterioration levels to, similarly to [6], allow for operational characteristics to be added to a parts condition. While they do not directly model component inter-dependence, they do consider a *k-out-of-n:F system*. Which considers a system to contain a total of n components and when k out of those are in a failed state simultaneously at some point, the system will fail. While the step-wise deterioration of each individual part is not linked to other parts, the subsequent maintenance decisions in this case obviously are linked by the condition of multiple parts. [4] again discretise the deterioration levels of parts. The step-wise deterioration for each part is modelled using a Poisson distribution and parts have no influence on other parts' deterioration processes, in other words, there is no part interdependence. Notable is that [4] define the cost of operation and replacement as a function of a part's deterioration. Thereby representing that a more deteriorated part functions less efficiently and is harder to replace than a less deteriorated one. [9] model multiple different parts instead of multiple parts which are the same. This is very useful and common in the machine or production setting but less applicable in the railway infrastructure domain. Their optimisation technique is also interesting as they perform their optimisation sequentially, first finding the best value of the repair/replacement time and thereafter determining the best maintenance actions to perform at those times to produce the least amount of costs. Finally, [10] uses an (R,s,nQ) inventory control model. While this disqualifies it as a potential base model as described earlier in this chapter, the model itself is still valuable to review in terms of inspiration for possible extensions on another possible base-models. One of the interesting peculiarities of this model is that the failure of parts is modelled by using a Homogeneous Poisson Process to approximate the arrival of defective parts. Rather than modelling parts separately, this model sees the entire set of identical installed parts as a queue from which parts stochastically progress to the failed state.

This review of joint spare inventory and maintenance scheduling models has combined the work of [11] and a review of the more recent research published in this field. An overview of the most important aspects of each of the reviewed models as well as the model developed in this research are displayed in table I.

Because of the characteristics of the model to be developed and the close resemblance to the model by [8], this model is chosen as a base model to develop further into the new simulation in this research.

Optimisation

All these models use different methods of optimisation, ranging from analytical solutions to Markov Decision Processes (MDP) [5]. For the developed model, an MDP was considered too cumbersome for the user and analytical options or other options containing derivative information were not applicable as the developed model sees the determination of the objective value as a black-box as a result of its stochastisity. Therefore derivative free global surrogate modelling is considered to be the best suitable method of optimisation here.

Surrogate modelling is a technique used in optimisation which uses experimental data of decision variable values and accompanying objective values to estimate a relationship between both. This estimated relationship is called the *surrogate* [12]. The way in which these surrogates are used in optimisation is schematically displayed in figure 1.

The experiment in figure 1 in this research takes the form of the simulation part of the railway spare inventory and maintenance scheduling model. This model will take data (inspection times, costs etc.) as an input and produces an expected cost (the objective) as the output. Several different surrogate modelling algorithms have been developed and each has its own peculiarities to account for.

Randomsearch is one of the most basic algorithms available [13]. It randomly selects values for the decision parameters and tests the core problem (the experiment) to find the value of the objective. It skips fitting a surrogate and its DoE can be described as taking a random pick for every decision variable from a uniform distribution over each variable's respective range. Surprisingly this method has proven quite useful in certain situations, therefore it is always wise to

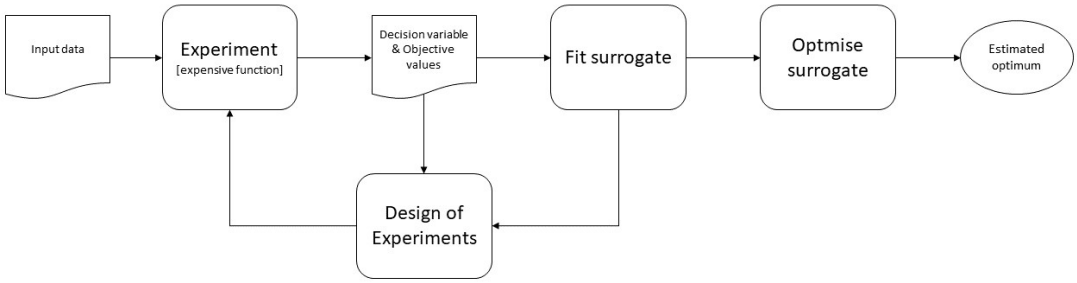

Fig. 1. Basic structure of surrogates used in optimisation

include this algorithm as a base-line for the performance on one's surrogate modelling approach. *SMAC* uses random forests [14], an approach which is deemed very well suited for categorical or discrete variables. *HyperOpt* [13] uses the Tree Parzen Estimators (TPE) algorithm. In practice *SMAC* and *HyperOpt* produce very similar results. SMAC however is less usable in this instance in terms of software integration. *COMBO* or COMmon Bayesian Optimization uses a continuous surrogate [15]. Similarly, *Bayesianopt* and *pyGPGO* [16] also use a continuous surrogate as does *DONE* [17], which has specifically shown a distortion effect at the edges of the search space, which is advisable to keep in mind when selecting an algorithm to fit ones specific problem situation. *CoCaBo* is tailored to combine categorical and continuous inputs [18]. *IDONE* being an extension on the *DONE* algorithm does use a continuous surrogate as well, but is adjusted in such a way that the final optimal solution is always an integer valued one [19] thereby subverting the distortion around the edges of the search space. One of the drawbacks of *IDONE* is that it has a limited exploration parameter. This means that it is more local-leaning. While none of the algorithms described can fully assure that a global optimum is reached, one might consider expanding the exploration parameter somewhat to increase the search area of the *IDONE* algorithm. This is exactly what is effectively done in the *MVRSM* algorithm [20], apart from also accommodating for continuous variables rather than only integer ones. Finally *Submodular Relaxation BOCS* [21], being an extension of the better known classical *BOCS* [22], might also be suitable for this problem although practical concerns regarding the computational effort of this method could be considered in the choice to include it in the roster of algorithms to be tested.

The final modelling framework thus consists of two parts: the developed Monte Carlo simulation which predicts an estimated cost based on part data and values for the three decision variables and an optimisation algorithm which optimises the three decision variable values through surrogate modelling. Together these form the functional Railway - Spare Inventory & Maintenance Scheduling (R-SIMS) model.

3. Problem definition

The simulation was developed using a base model from literature [8]. Other than general alterations of the base simulation which increased accuracy and applicability in railways, the new simulation was equipped with possibilities for minimal maintenance in a case of stock-out after an unexpected failure of a part. The possible failure dynamics and subsequent maintenance actions are visually represented in figure 2.

To model these processes, several categories of assumptions can be defined:

Parts

The studied system is modelled for *n* identical parts. Regarding these parts, the following is assumed:

Each part, upon installation, is as-good-as-new, meaning that the deterioration of the part, described by $x(t)$ is 0. After installation, the part immediately starts to deteriorate incrementally over time at a rate of $\frac{\Delta x}{\Delta t}$. Where Δx is randomly picked from a pre-determined probability distribution for every part for every time-step. This indicates that $x(t) = x(t-1) + \Delta x$. Since Δx is always positive, the deterioration will only increase through time. Further, each part gets assigned a *failure threshold* X_f for their lifetime. Once the deterioration of a part exceeds this threshold, the part is considered *failed*. This threshold is picked from a pre-determined probability distribution similarly to the step-wise deterioration. Every time a new part is installed this threshold is re-picked and this threshold is always positive. Because all parts in the considered system are identical, all failure behaviour is described by the same distributions throughout the system. This means that local differences in environmental conditions or different levels of use of a part compared to another is not taken into account in this model. Part failures are also assumed to be self-announced. This means that in a period which does not include an inspection, failed parts are still noticed and replacement decisions can be made as a result outside the inspection cycle.

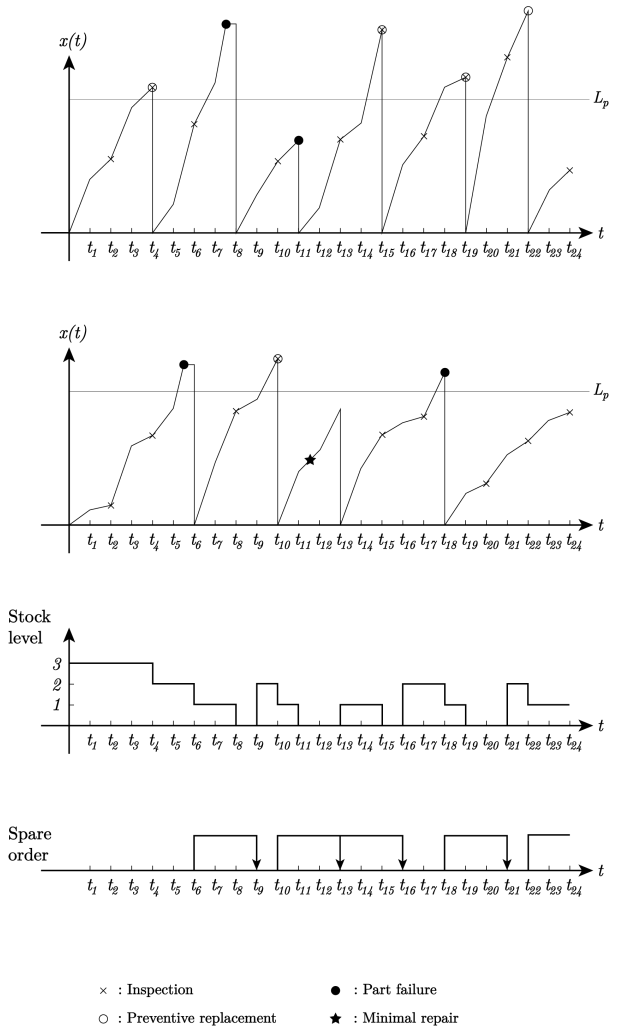

Fig. 2. Behaviour of part failure and maintenance decisions on stock levels and system functioning (based on [8])

Maintenance operations

Three different maintenance operations are considered: inspection, replacement and minimal repair. Replacement can be done either in a preventive or corrective fashion. All of these operations have a certain cost. Where performing inspections is governed by pre-determined time intervals (T), both the minimal repair as well as the replacement decisions are based on the condition of the part (Condition Based Maintenance). The specific condition can either be the level of deterioration in comparison with a CBM threshold or whether part has failed or not. The following assumptions apply:

Inspection is performed on a part every T time steps. Meaning that if the inspection interval T for instance equals 4, inspections are performed in time step 4, 8, 12, 16, ... until the part is replaced or failed. After replacement of a part, the cycle restarts, this means that if a part is replaced at $t = 17$, the new inspection cycle will look like 21, 25, 29, ... Each of these inspections is assumed instantaneous and non-invasive, meaning that it does not influence the deterioration level of the inspected part. Upon inspection the part's condition is perfectly determined. Replacement as well as minimal repair are also considered to be instantaneous.

If a part is in a failed state, the system will try to correctively replace said part as soon as possible. Yet, a part can only be replaced if spare part stock is available. A part that can not be replaced correctively due to stock-out of spare parts will be immediately subjected to a minimal repair, meaning that the part will exit the failed state and will be restored to the condition directly before failure. The deterioration level will be reset to the level before failure and a new failure threshold X_f will be picked such that it is higher than the current deterioration level. The part will subsequently be scheduled to be replaced correctively as soon as spare stock allows.

If a part is inspected, and the deterioration is shown to exceed the CBM threshold L_p, a part will be replaced preventively as soon as possible. A part that can not be replaced preventively will be scheduled to be replaced preventively as soon as spare stock allows. In both cases where a part has failed or exceeded the CBM threshold but can not be replaced due to stock-out the 'no-more-inspection' clause will come in to effect. Meaning that the part will not be inspected any more until it is replaced.

Spare inventory control

Inventory of spare parts is controlled using an (s,S) policy in which $s < S$. This policy entails continuous monitoring of the spare IOH. The stock is assumed to always be available for operations and all stock keeping actions are instantaneous. Further assumptions are:

If spare stock equals reorder level s, new spares will be ordered if no order is outstanding at that time. Only a single order can be outstanding at any one time. If the order marker $OC = 1$ no orders can be placed. Only once those ordered parts are delivered a lead time (t_s) later can another order be placed. Spare orders are always of the quantity $S - s$ in order to bring the Inventory Position back up to the order up-to level S. Moreover the spare stock level (the physical amount of spares available) SL can never exceed S or become lower than 0. Spares that remain in stock incur a holding cost each period that they are kept in stock. Parts can be kept in stock indefinitely and will not deteriorate while kept in stock. They have an infinite shelf-life. Finally, every replacement action decreases SL by exactly 1.

4. METHODOLOGY

All of these assumptions lead to a Monte Carlo simulation which is further represented in Supplement A. The decision tree which results from all the assumptions made is depicted in figure 3.

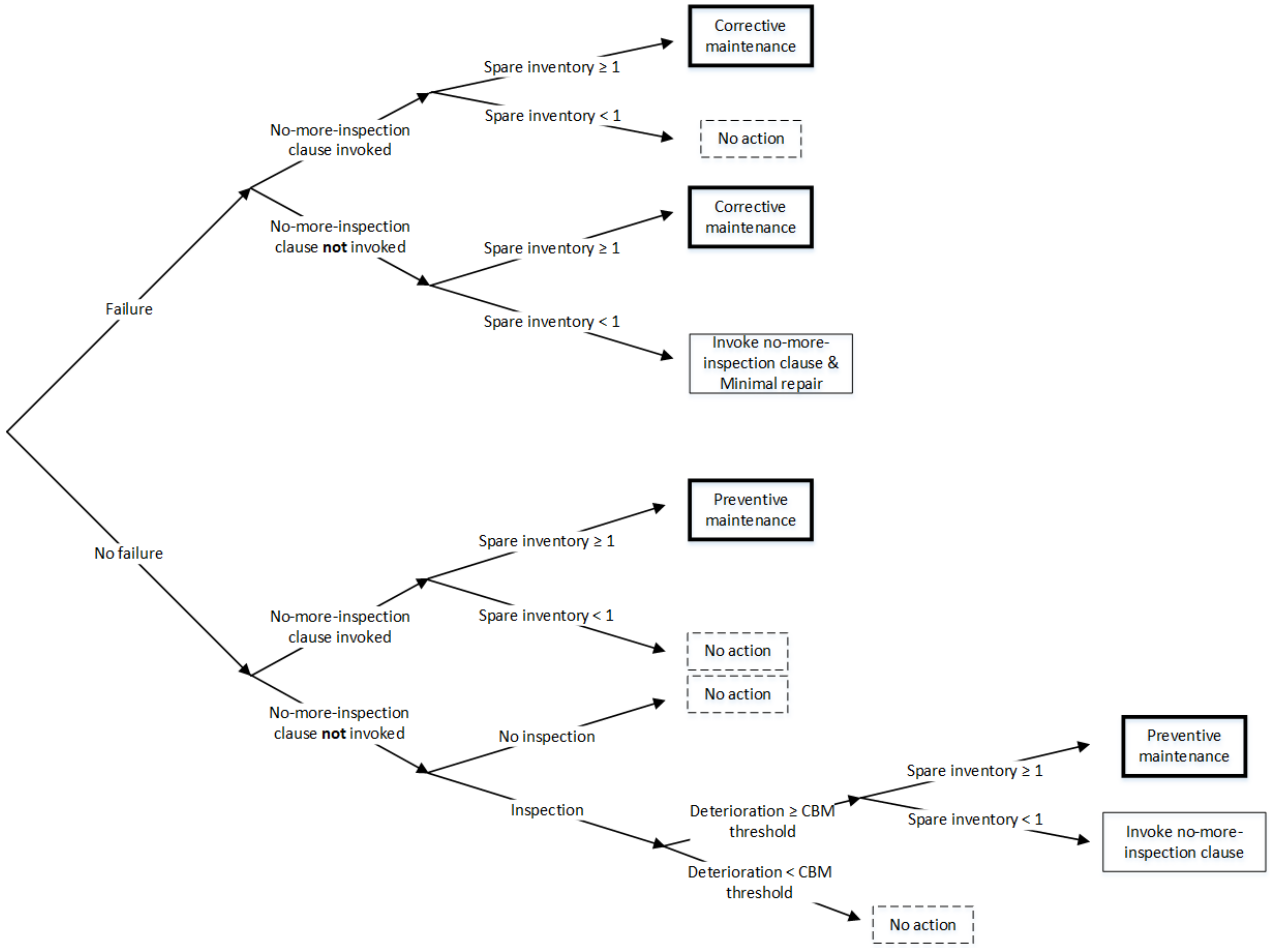

Fig. 3. Representation of the actions/choices made for each part in each time step in the proposed model

All actions and decisions in the described model result in incurred monetary cost. Cost factors are included for inspection and maintenance actions but also for holding costs, costs of downtime and cost of part procurement. All of these costs combined form the basis for the simulation's output which is the expected cost per part per time period EC [€/part/period]. This output is calculated by adding all costs together and dividing them by the time horizon multiplied by the number of parts included. This cost function can be represented using:

$$EC = \frac{C_i N_i + C_p N_p + C_c N_c + C_h N_h + C_s N_s + C_d N_d + C_m N_m}{t_m \cdot n}$$

The full simulation algorithm is described in Supplement A in pseudo-code. The separate steps of that algorithm are:

Data entry {Require}
This step entails the entry of all the necessary parameter values by the user. Note that the three decision variables are also required as the optimisation step lies a level beyond the simulation in the surrogate modelling algorithm. The output of the simulation is also defined as the estimated cost.
Initialization {L4 - L11}
All variables that do not pertain to a single part are initiated. As well as all values which are part-specific. Most notably every part gets assigned a failure threshold for the first lifetime and a deterioration level which is randomly assigned for a value between 0 and the failure threshold.
Next the first time-step begins, each time-step has the same sequence of actions and these are all processed in ascending order until the entered time horizon is reached.
Spare order handling {L13 - L19}
At the start of each period newly delivered orders are accepted. If an order was outstanding and the order has been placed a lead-time ago, this order will arrive and be added to the current inventory position. Because of the (s,S) inventory control strategy, orders always amount to $S - s$ parts. If an order is accepted, the order marker can also be reset to 0.
Next each part is handled separately. The order in which the parts are handled is randomised.
Deterioration and failure {L24 - L29}
The part's deterioration level is updated by Δx. This value is picked from its appropriate probability density function (PDF). This new deterioration level can then be checked against the

failure threshold to determine whether the part has failed or is still operational.

No-inspection clause {L30 - L35}
Instrumental in this model, is the option that parts are no longer inspected after they may have failed or exceeded the CBM threshold in earlier time-steps. If this is the case ($IN_j = 0$) the part will have to be replaced if the spare stock allows. The part will then be correctively replaced if it has failed and preventively be replaced if it is still operational.

Corrective replacement or minimal repair {L37 - L48}
If a part has failed and stock levels allow, it should correctively be replaced. If it has failed and stock levels are not sufficient, the part will be set to no longer be inspected, after all, the failure is now known and the part should be replaced as soon as possible. If it indeed has failed and no spare parts are available to replace, the part can be minimally repaired, but only if it has not been before. If it is minimally repaired, the marker is adjusted accordingly and its failure status is reset to operational as well. Finally the part's failure threshold has to be reset to a value, again picked from its appropriate PDF but in this case higher than the current deterioration level.

Inspection and preventive replacement {L50 - L63}
If the part has not failed at all, it will be considered to possibly be inspected. This inspection takes place periodically (with T as inspection interval) so if the current period in the part's lifetime is a multiple of T it shall be inspected. If upon inspection the part is found to have a deterioration level greater than or equal to the CBM threshold and enough spare stock is available, the part will be preventively replaced. If no stock is available, the part is set not to be inspected any longer and therefore to be replaced as soon as the spare stock allows.

Installation of new part {L64 - L74}
If a part is replaced, at this point this is recorded. This means that a new part is installed and some parameters have to be reset. The deterioration is set back to 0, indicating an as-good-as-new part. The start of the lifetime of the part is set to the current time-step. The failure threshold is renewed (again from its PDF). The minimal maintenance marker is reset as well as the marker for the no-inspection clause. Importantly, the spare stock is decreased by 1.

Spare ordering decision {L75 - L78}
Once the IOH of spares is updated, it is compared to the reorder level, if it has dropped to the reorder level and no order is outstanding, a new order will be placed, thereby resetting the last order time to the current time-step.
After all parts have been cycled through, each time-step ends with administrative steps.

Counting, resetting and advancing time {L80 - L90}
Every action or cost factor is marked and counted by its respective marker or memory variable. These values are all added to the values of the counters from the previous time-step. After all these actions are counted, the necessary markers are reset before the start of the next time-step. Finally time is advanced and a new time-step can be started.

Cost calculation {L91 - L95}
After the time-horizon is reached, the final total costs can be calculated through multiplying all cost counters by their cost factors. The Expected Cost over the simulation horizon is calculated by dividing the total costs by the time horizon multiplied by the number of parts. Finally the definitive estimate of the cost is calculated through dividing the cumulative Expected Cost by the number of Monte-Carlo iterations.

Optimisation

To optimise the decision variables, a suitable optimisation technique is found in surrogate modelling. Four surrogate modelling algorithms are identified as potential fits for the problem at hand. These algorithms are focused on problems with a single objective and integer decision variables. Moreover they are able to deal with problems which would have a solution close to the edges of a search space. The four candidate algorithms are: *Randomsearch* [23], *HyperOpt* [24], *IDONE* [19] and *MVRSM* [20]. These four algorithms are tested on the developed simulation to reveal which functions best on the given problem.

Randomsearch is the most basic of the possible methods [23]. Randomsearch technically does not make use of an entire surrogate modelling algorithm. Every iteration, the methodology pics a random value for each of the decision variables from their respective ranges. It thereby assumes a uniform distribution over the entire search space for each of the decision variables. Rather than storing the values of the decision variables and the objective for each iteration, the method only stores these values for the best iteration so far.

HyperOpt itself is a python based library created by [24] which in itself gives options for utilising multiple different algorithms for optimisation of expensive functions. The algorithm most prominently featured by the creators as a well-performing one is the Tree-structured Parzen Estimator approach (TPE) [25]. This algorithm is compared to Randomsearch by Bergstra as: "The TPE algorithm is like taking Randomsearch and then slowly refining it to not choose values for hyper-parameters that are strongly correlated with terrible performance" [26]. While the algorithm in this case is applied directly to parameters of our simulation the same holds true. HyperOpt in this research therefore indicates the use of the HyperOpt library [13] to optimise with the application of the TPE algorithm.

The TPE algorithm itself functions not through estimating a surrogate model to describe y, the objective value, as a result of x, the decision variable values; resulting in a probability of an objective value given a certain set of decision variable values $p(y|x)$. But by estimating the reverse (which decision variable values would be probable to fit a certain value of the objective): $p(x|y)$. It does so by taking the history of observed decision variable values and accompanying objective values and splitting it over an arbitrarily chosen desired performance y^* and estimating for each decision variable a probability

density function for y. This results in:

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \tag{1}$$

The algorithm is then trained to try new parameters which are more likely to have a higher density in $g(x)$ and a lower density in $l(x)$. This is done through selecting new experiment parameters (the DOE) based on optimising the Expected Improvement (EI). The EI is defined as the expected difference between the desired performance and the actual performance given a certain value of the decision variable:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*}(y^* - y)p(y|x)dy = \int_{-\infty}^{y^*}(y^* - y)\frac{p(x|y)p(y)}{p(x)}dy \tag{2}$$

The way in which y^* is determined is through the user choosing the ratio of the observations they would like to fall above or below the threshold. This ratio can be defined as $\gamma = p(y < y^*)$. In other words, say $\gamma = 0.8$ then 80% of observations would be determined to be less optimal than the desired performance. By implementing the expressions for γ, $l(x)$ and $g(x)$ into Eq.(2) one gets the final expression for EI:

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy}{\gamma l(x) + (1-\gamma)g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1} \tag{3}$$

Eq.(3) shows that in order to maximise EI, the algorithm will have to propose values of the decision variables x^* which have a higher likelihood in $l(x)$ than in $g(x)$. After each iteration the experiment is ran again with the new values of x^*, the results are taken into the history and used together with the previously ran experiments to develop new PDFs and new estimated optima.

IDONE stands for Integer Data-based Online Nonlinear Extremum-seeker [19], which is an extension of the *DONE* algorithm [17]. IDONE, different from DONE, uses a linear combination of Rectified Linear Units (ReLU) as its surrogate. The surrogate model itself is indicated by $g(x)$ and consists of D basis functions.

$$\begin{aligned} g(\mathbf{x}) &= \sum_{k=1}^{D} c_k \max\{0, z_k(\mathbf{x})\} \\ z_k(\mathbf{x}) &= w_k^T \mathbf{x} + b_k \end{aligned} \tag{4}$$

Where $\mathbf{x} \in \mathbb{R}^d$ are the d decision variables. By fixing the parameters w_k and b_k, the model becomes linear in its parameters c_k. The specific choice of the basis functions makes this model have local minima exactly on its corner points [19]. Each iteration, this model is fit to the available data. The available data consists of pairs of decision variable values and their corresponding objective value ((x_i, y_i) being the (decision variables, objective value) pairs for iteration i). The model fit is performed by solving the regularized linear least squares problem:

$$\min_{c_N} \sum_{n=1}^{N} (y_n - g(\mathbf{x}_n, c_N))^2 + \lambda \left\| c_N - c_0 \right\|_2^2 \tag{5}$$

To solve this problem the recursive least squares algorithm is used [27]. This algorithm uses the optimum of the fitted surrogate as its DOE for the next iteration. This means that at each iteration the surrogate described in Eq.(4) is fit to the data using Eq.(5) and subsequently optimised using the BFGS method described by [28]. This optimisation yields a single set of decision variable values $\mathbf{x}^*$ which is then used as the new experiment to run in order to produce new data. However, to avoid getting stuck in a local minimum, first a small alteration is added to the values: $\mathbf{x}_{N+1} = \mathbf{x}^* + \delta$ where $\delta \in \{-1, 0, 1\}$. This cycle is repeated for as many iterations as the user desires after which the last optimum calculated will form the suggested optimum of the original problem.

MVRSM, the Mixed-Variable ReLUbased Surrogate Modelling algorithm, is an adaptation of IDONE. Where IDONE was designed specifically to deal with integer valued decision variables, MVRSM was designed to deal with a mixture of integer and continuous variables [20]. The adapted surrogate used in MVRSM distinguishes continuous ($\mathbf{x}_c$) and discrete ($\mathbf{x}_d$) decision variables:

$$\begin{aligned} g(\mathbf{x}_c, \mathbf{x}_d) &= \sum_{k=1}^{D} c_k \max\{0, z_k(\mathbf{x}_c, \mathbf{x}_d)\} \\ z_k(\mathbf{x}_c, \mathbf{x}_d) &= \begin{bmatrix} v_k^T & w_k^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_d \end{bmatrix} + b_k \end{aligned} \tag{6}$$

By then choosing v_k, w_k and b_k at the start and fixing them the model becomes linear in its parameters c_k again which lends it to be optimised using linear regression. The method of choosing these parameters influences the ability of the algorithm to deal with both continuous and integer valued decision variables by locating any local minimum on intersections of decision variables where the integer constraints are respected for those variables which require them and these constraints are relaxed for the continuous variables. The fitting of the surrogate happens exactly in the same manner as in IDONE, using the recursive least squares algorithm. Also similar to IDONE, MVRSM uses the optimum of the fitted surrogate for its DOE for the experiments in the next iteration. However, the Limited-memory BFGS (L-BFGS) algorithm is used to optimise the fitted surrogate as it uses less memory than BFGS [28]. The optimum of the surrogate found by the L-BFGS algorithm $(\mathbf{x}_c^*, \mathbf{x}_d^*)$ is then used to construct the seed for the experiment in the next iteration. This requires the final step of adding a deviation in order to avoid getting stuck in a local minimum through $(\mathbf{x}_c^{N+1}, \mathbf{x}_d^{N+1}) = (\mathbf{x}_c^*, \mathbf{x}_d^*) + (\delta_c, \delta_d)$ where $\delta_c \in \mathbb{R}^{d_c}$ and $\delta_d \in \mathbb{Z}^{d_d}$. Note here that the perturbations made to the optimal values are real values for the continuous variables and integer values for the discrete variables. Moreover this method leaves room for the user to choose a mix of possible values to use as perturbation which exceed the confines of those used in the IDONE algorithm. This allows MVRSM to use larger exploration parameters.

TABLE II
ALGORITHM CHOICE EXPERIMENTS PARAMETER LEVELS (SEE SUPPLEMENT A FOR NOMENCLATURE)

C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	$f^F(x)$	$f^D(x)$	t_m	n
50	224	140	832140	5500	100.83	448	6	12	*Weibull(871.984, 23.406)*	*Gamma(1.667, 0.6)*	1690	30

Now that the developed simulation is described and the potential surrogate modelling algorithms are detailed, experimentation can further inform about the use and fit of these parts in the final modelling framework.

5. NUMERICAL EXPERIMENTS

Two experiments are performed, the first to choose the best fitting algorithm and a second to gain demonstrate the full model. To perform these experiments, the simulation algorithm is written in functional code in Python. That simulation is subsequently implemented in the EXPOBench benchmark library [29]. This library, also written in Python, provides options for optimisation of the entered problem using different surrogate modelling algorithms, including the four candidate algorithms in this research. All experiments are ran on a Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz with 8 GB of RAM system or comparable consumer-oriented systems. More extensive results and analyses are presented in [30]

Algorithm choice

First, to finish off the model development itself, an *algorithm choice* experiment is performed. To test the fit of the algorithms, three metrics are designed: *Performance*, here defined as the objective value produced by an algorithm in a given amount of time, *Speed*, measured using the time per iteration as well as with the time required by an algorithm to reach a certain threshold in terms of the objective value and *Stability*, defined by the variation in the definitive result of an algorithm. The experiments encompass a single problem seed being ran through all four candidate algorithms for a total of

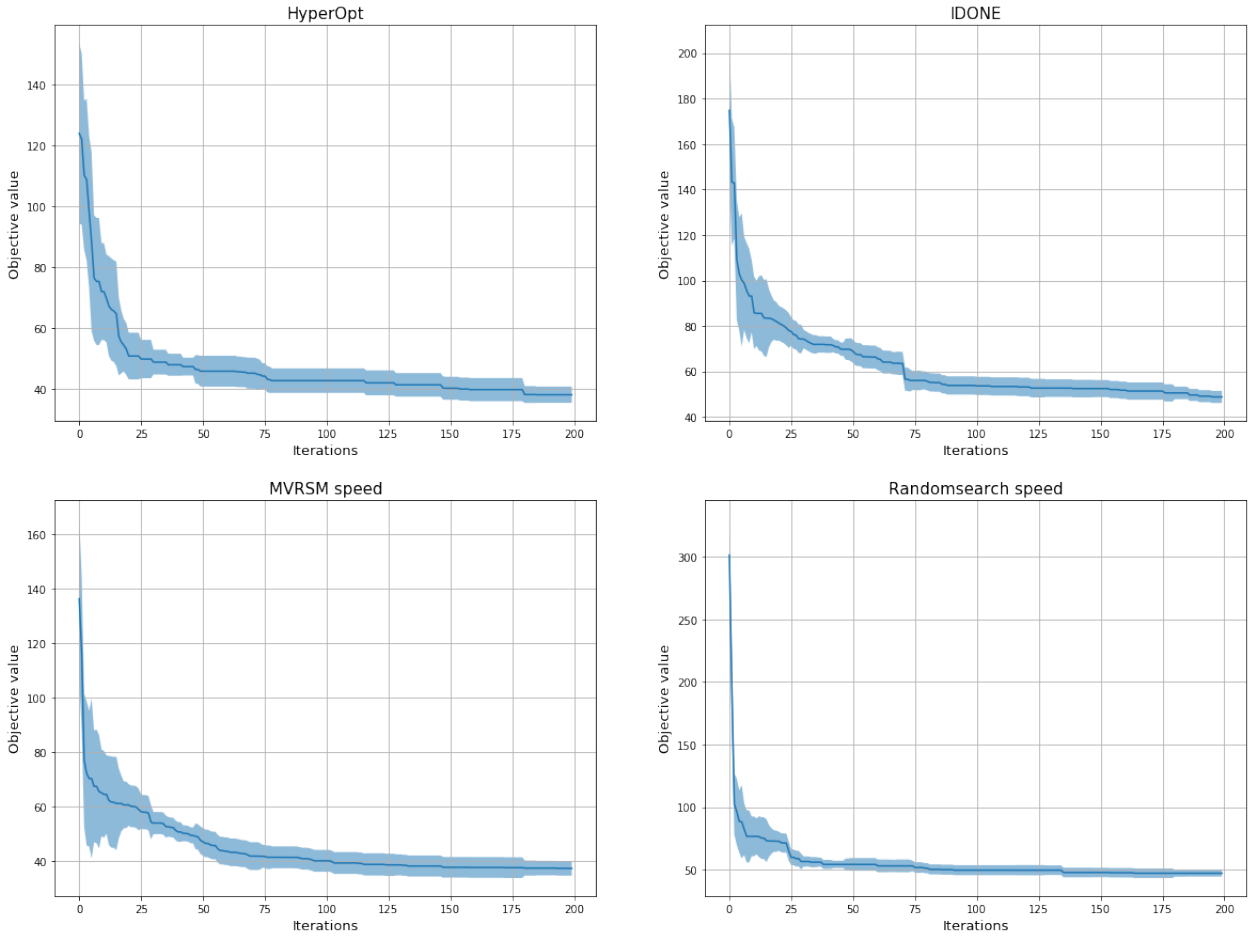

Fig. 4. Performance runs averaged over the 10 repetitions of the experiment with a shaded area indicating one standard deviation of the results above and below the average.

200 iterations to produce an estimated optimal result, this is repeated 10 times. The parameter values in the seed are shown in table II.

The number of Monte Carlo iterations for these experiments is set to 25 to facilitate the production of proper quantities of data. The objective value threshold for the speed experiment is set to 150% of the average objective value found by all algorithms in 200 iterations, this value amounts to 64.32 €/part/period. Figure 4 shows the progression of the best objective found by each algorithm over the 200 iterations.

Results show that for *performance*, the MVRSM and HyperOpt algorithm perform equally well with an average objective value found of 37.42 and 38.20 €/part/period respectively versus an average of 47.12 and 48.77 €/part/period for Randomsearch and IDONE respectively. A pair-wise t-test showed no significant difference between HyperOpt and MVRSM ($p = 0.6986$) nor between Randomsearch and IDONE ($p = 0.6709$). Other pairings of the algorithms did show significant differences ($p < 0.025$) leading to HyperOpt and MVRSM scoring equally well on *performance*. To test speed however HyperOpt out-performs MVRSM for the objective value found (coefficient of variation of the objective of 0.07 for HyperOpt and 0.14 for MVRSM) as well as for the found values of the decision variables (combined coefficient of variation of 1.55 for HyperOpt versus 3.63 for MVRSM). This leads to HyperOpt being the most favourable algorithm for optimisation of the studied problem.

The propagation of the iteration time per iteration over the 200 iterations of each algorithm is shown in figure 5. Interestingly, IDONE and MVRSM algorithms both have a very low iteration time for the first three iterations. This is the result of the algorithms requiring a base set of data to start actually fitting the surrogate. IDONE and MVRSM therefore perform random picks of the decision variables for the first three iterations which results in very low iteration times [19] [20]. After these first iterations the iteration times of IDONE and MVRSM increase to values beyond those of the HyperOpt or Randomsearch algorithms. This results in average iteration times for IDONE and MVRSM of 13.87 s and 13.61 s respectively which are both significantly higher than the

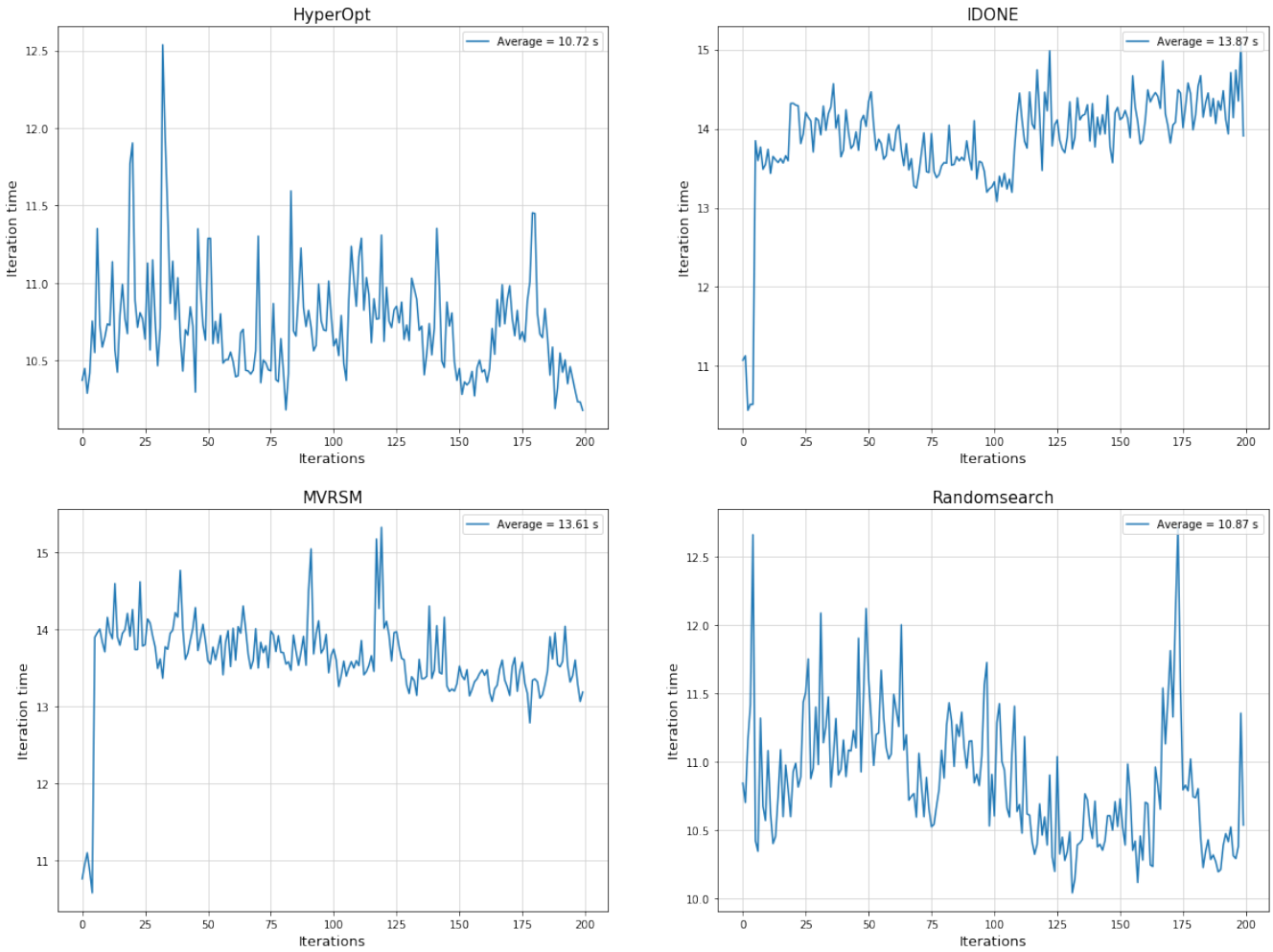

Fig. 5. Iteration time per iteration averaged for all 10 repetitions of the experiment

10.72 s and 10.87 s average iteration time for HyperOpt and Randomsearch respectively.

For the time required by each algorithm to reach the goal of 64.32 €/part/period for the objective results are shown in table IV. Again HyperOpt used the least amount of time on average to reach this goal (2.68 minutes), followed by MVRSM (4.56 minutes), then Randomsearch (6.05 minutes) and finally IDONE (7.79 minutes) which was even unable to reach the goal within the 200 iterations of the experiments in one of the experiment repetitions.

Both of these analyses showed HyperOpt to be (one of) the best performing algorithms in terms of *Speed*. Which would make it the most suitable for the developed Monte Carlo simulation in that regard.

TABLE IV
TIME PER EXPERIMENT REPETITION TO REACH THE GOAL OBJECTIVE OF 64.32 €/PART/PERIOD FOR ALL ALGORITHMS IN MINUTES

Experiment repetition	HyperOpt	IDONE	MVRSM	Randomsearch
1	3.22	15.67	6.25	5.47
2	0.97	17.85	0.85	2.61
3	3.16	1.35	9.89	4.80
4	1.33	-	10.29	5.08
5	4.56	3.10	3.20	7.34
6	2.15	5.25	9.04	4.91
7	5.08	9.75	0.33	4.25
8	2.15	7.50	0.67	1.34
9	1.33	7.49	1.27	12.82
10	2.81	2.15	3.84	11.92
Average	2.68	7.79	4.56	6.05

To score the algorithms on their *Stability*, the variation of their results is calculated for all decision variable outcomes as well as the resulting objective value (see table V). When taking into account all of those variations, in terms of coefficient of variation (σ/μ) the HyperOpt algorithm is clearly the most stable. which supports the choice for HyperOpt as the algorithm from a *Stability* perspective.

TABLE V
COEFFICIENTS OF VARIATION OF EXPERIMENT RESULTS

Algorithm	**CV**(S)	**CV**(s)	**CV**(L_p)	**Cumulative CV**(*DV's*)		**CV**(*Obj*)
Randomsearch	0.55	1.86	0.31	2.72		0.11
HyperOpt	0.40	1.11	0.04	1.55		0.07
IDONE	0.66	1.41	0.33	2.4		0.23
MVRSM	0.41	3.00	0.22	3.63		0.14

Test-case

With the best fitting algorithm identified, as a means of practical testing, the model is used to optimise the decision variable values for a specific railway infrastructure part. This part is a sub-assembly of a track circuit sensor called a RUTA.

To run the developed model for this particular part, all parameter values are to be gathered as accurately as possible. All parameter values for this RUTA are gained from data provided by ProRail and its contractors. ProRail is the largest Dutch railway infrastructure manager. Information about the cost of inspection, replacement and maintenance actions are gathered by inquiring maintenance engineers about their work. The lead time and cost of the part were stored in supplier information available, the holding costs are calculated as a percentage of the part costs. the downtime costs were calculated through averaging rail section values then the average downtime per failure is determined from failure data sets. Finally the inspection interval is provided by the contractor's FMECA analysis. The final test data can be found in table III.

The results of this test-case show an estimated optimum at a reorder level (s) of 15 units, an order up-to level (S) of 398 units and a preventive maintenance threshold (L_p) of 140 units of deterioration. The objective value of the expected cost that accompany these decision variable values is €138.84 per part per period. Before finding this final result in the 155th iteration, the HyperOpt algorithm found 5 previous best objective values. This results in a graph of best results found which is rather simple as seen in figure 6.

All objective values found during the 200 iterations of searching are visualised in figure 7.

The use of the model, while functional, is not very well adjusted to the information sources available in this case. Limited amounts of data were available regarding failure behaviours which leads to a lot of assumptions and generalisations. This leads to a case of 'garbage in - garbage out', meaning that the final result is only as reliable as the failure data input into the model. Moreover six different data sources were consulted to gather all the parameter values used for this test case. This not only takes a lot of time and resources but could also lead to contradicting data or different parameter values that counteract one another. An advice would therefore be that any user would first spend time gathering the right data sources and integrating them into a single large data set which is reliable and straightforward in its use for this model.

TABLE III
RUTA TEST CASE DATA

C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	$f^F(x)$	$f^D(x)$	t_m	n
7.50	12.50	12.50	389,367	7,026.62	122.82	146.25	10	24	Weibull(1306.416, 1.764)	Gamma(1.667, 0.6)	1774	1009

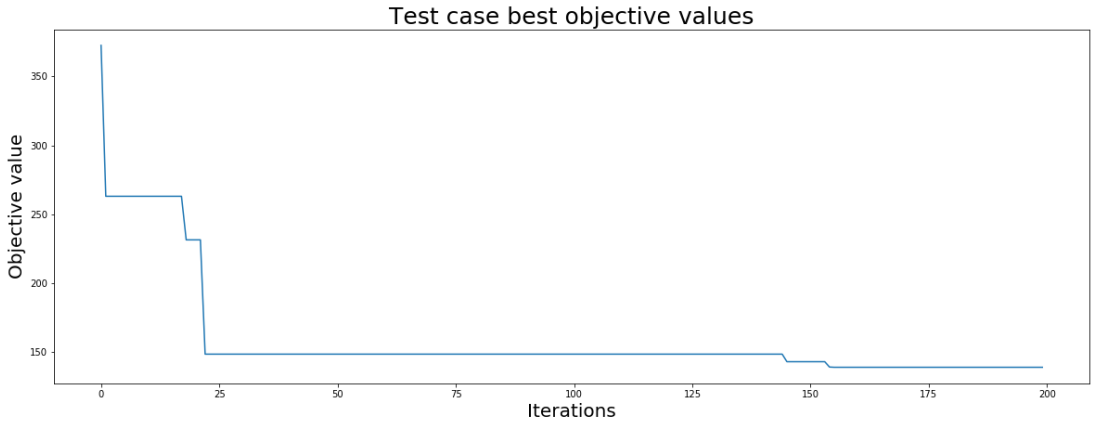

Fig. 6. Best Objective value found per iteration

Regardless, the performance of the R-SIMS model in this test case can be regarded as highly satisfactory. By exploring the search space in an efficient way it found the final result in 155 iterations. This may not always be the case due to the stochasticity inherent to both the simulation and the optimisation algorithm but still indicates that shorter run times could be used in practice than is done in this research.

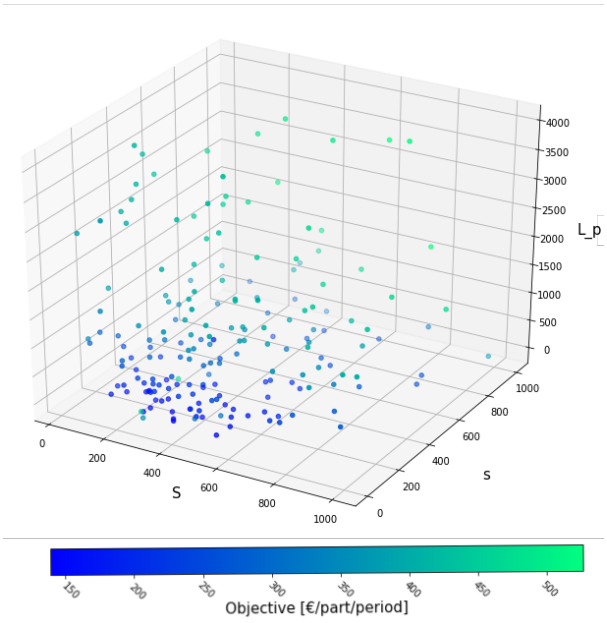

Fig. 7. All points tested in the search space during test-case run (for visibility the 6 values with an objective higher than €1,000 are cencored)

6. Conclusion

The developed model satisfies the research goal, it optimises estimated costs per part per period and returns values for the three decision variables which govern both spare inventory control (reorder level s and order up-to level S) and maintenance scheduling (CBM threshold L_p). By choosing proper elements to build the simulation (e.g. the CBM and (s,S) policy and adding minimal maintenance) this model better suits the railway infrastructure context than the models available in literature.

Future research on this topic could focus on:

More research could be done to gain insights into best practices when it comes to data selection. The data required to utilise operational models is often very detailed and widespread throughout an organisation or over stakeholders. The aggregation of data sources and control of access is an entire field of data science which would be applicable to this research. Secondly, alterations to allow the model to include parts which have continuous monitoring capabilities would be helpful. Because large systems like the railway infrastructure often involve different sets of parts. Some of these parts in practice have their own dedicated continuous monitoring systems which allow the owners to implement another maintenance scheduling strategy. Making the model more modular in such a way that the user could run these parts through as well would be a great increase in value for the model. Thirdly, including parts which have an *(s,nQ)* inventory control strategy. Similarly to the previous point, some part sets have a different inventory control strategy from the one used in the R-SIMS model. By again making the model more modular to accommodate for these strategies it would be more widely applicable in practice. Fourth, better methods for estimating downtime costs should be developed. In the developed model, because of the granularity of time it is very difficult to balance computational needs with an accurate estimate of downtime cost. Especially in infrastructure systems like the railways where downtime is extremely costly this factor is important. Further work on the implementation of the downtime costs would therefore be a great contribution to the R-SIMS model in particular. Finally, possible expansion

of the application outside the railway management domain. While one of the biggest contributions of this research is the development of a combined spare inventory & maintenance scheduling model specifically suited for railway infrastructure, the literature review showed that in general these models rarely include functionalities for minimal maintenance. This functionality is particularly valuable in the railway setting but could also be applied in specified other applications which for instance are hard to reach with spare parts. Research in this wider application would therefore increase the benefit industry would be able to gain through the use of the R-SIMS model.

REFERENCES

[1] L. Huang, Y. Chen, S. Chen, and H. Jiang, "Application of rcm analysis based predictive maintenance in nuclear power plants," in *2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*. IEEE, 2012, pp. 1015–1021.

[2] A. Jamshidi, S. Hajizadeh, Z. Su, M. Naeimi, A. Núnez, R. Dollevoet, B. De Schutter, and Z. Li, "A decision support approach for condition-based maintenance of rails based on big data analysis," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 185–206, 2018.

[3] K. H. Donselaar and R. A. Broekmeulen, "Lecture notes and toolbox for the course stochastic operations management 1cv20," March 2017, bETA working paper 447.

[4] M. C. O. Keizer, R. H. Teunter, and J. Veldman, "Joint condition-based maintenance and inventory optimization for systems with multiple components," *European Journal of Operational Research*, vol. 257, no. 1, pp. 209–222, 2017.

[5] J. Wang and X. Zhu, "Joint optimization of condition-based maintenance and inventory control for a k-out-of-n: F system of multi-state degrading components," *European Journal of Operational Research*, vol. 290, no. 2, pp. 514–529, 2021.

[6] L. Wang, J. Chu, and W. Mao, "An optimum condition-based replacement and spare provisioning policy based on markov chains," *Journal of Quality in Maintenance Engineering*, vol. 14, no. 4, pp. 387–401, 2008.

[7] J. Xie and H. Wang, "Joint optimization of condition-based preventive maintenance and spare ordering policy," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2008, pp. 1–5.

[8] L. Wang, J. Chu, and W. Mao, "A condition-based replacement and spare provisioning policy for deteriorating systems with uncertain deterioration to failure," *European Journal of Operational Research*, vol. 194, no. 1, pp. 184–205, 2009.

[9] A. Van Horenbeek and L. Pintelon, "A joint predictive maintenance and inventory policy," in *Engineering asset management-systems, professional practices and certification*. Springer, 2015, pp. 387–399.

[10] W. Wang, "A stochastic model for joint spare parts inventory and planned maintenance optimisation," *European Journal of Operational Research*, vol. 216, no. 1, pp. 127–139, 2012.

[11] A. Van Horenbeek, J. Buré, D. Cattrysse, L. Pintelon, and P. Vansteenwegen, "Joint maintenance and inventory optimization systems: A review," *International Journal of Production Economics*, vol. 143, no. 2, pp. 499–508, 2013.

[12] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of global optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[13] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *International conference on machine learning*. PMLR, 2013, pp. 115–123.

[14] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration (extended version)," *Technical Report TR-2010–10, University of British Columbia, Computer Science, Tech. Rep.*, 2010.

[15] T. Ueno, T. D. Rhone, Z. Hou, T. Mizoguchi, and K. Tsuda, "Combo: an efficient bayesian optimization library for materials science," *Materials discovery*, vol. 4, pp. 18–21, 2016.

[16] J. Jiménez and J. Ginebra, "pygpgo: Bayesian optimization for python," *Journal of Open Source Software*, vol. 2, no. 19, p. 431, 2017.

[17] L. Bliek, H. R. Verstraete, M. Verhaegen, and S. Wahls, "Online optimization with costly and noisy measurements using random fourier expansions," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 1, pp. 167–182, 2016.

[18] B. Ru, A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts, "Bayesian optimisation over multiple continuous and categorical inputs," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8276–8285.

[19] L. Bliek, S. Verwer, and M. de Weerdt, "Black-box combinatorial optimization using models with integer-valued minima," *Annals of Mathematics and Artificial Intelligence*, vol. 89, pp. 1–15, 2020.

[20] ——, "Black-box mixed-variable optimisation using a surrogate model that satisfies integer constraints," *arXiv preprint arXiv:2006.04508*, 2020.

[21] A. Deshwal, S. Belakaria, and J. R. Doppa, "Scalable combinatorial bayesian optimization with tractable statistical models," *arXiv preprint arXiv:2008.08177*, 2020.

[22] R. Baptista and M. Poloczek, "Bayesian optimization of combinatorial structures," in *International Conference on Machine Learning*. PMLR, 2018, pp. 462–471.

[23] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. 2, pp. 281–305, 2012.

[24] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in science conference*, vol. 13. Citeseer, 2013, p. 20.

[25] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *25th annual conference on neural information processing systems (NIPS 2011)*, vol. 24. Neural Information Processing Systems Foundation, 2011.

[26] Enthought, "Hyperopt: A python library for optimizing machine learning algorithms; scipy 2013," https://www.youtube.com/watch?v=Mp1xnPfE4PY, July 2013.

[27] V. Madisetti, *The digital signal processing handbook*. CRC press, 1997.

[28] S. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, no. 67-68, p. 7, 1999.

[29] L. Bliek, A. Guijt, R. Karlsson, S. Verwer, and M. de Weerdt, "EXPObench: Benchmarking surrogate-based optimisation algorithms on expensive black-box functions," *arXiv preprint arXiv:2106.04618*, 2021.

[30] D. Sommers, "Railway - spare inventory & maintenance scheduling model development," Master's thesis, Delft University of Technology, July 2021.

A - SIMULATION MODEL

Nomenclature

$x_j(t)$	Deterioration level of part j at time-step t
$X_{f,j}$	Failure threshold for deterioration of part j
$f^F(x)$	Probability Density Function for the Failure threshold
Δx	Incemental deterioration
$f^D(x)$	Probability Density Function for the Incremental deterioration
L_p	Condition Based Maintenance threshold for deterioration
S	Order up-to level
s	Reorder level
EC	Expected Cost per part per time period over the simulation horizon
EC_{MC}	Cumulative Expected Cost per part per time period
n	Number of parts installed
T	Inspection interval
t_s	Lead time
I_j	Marker for whether part j should be inspected in the current time-step
F_j	Marker for whether part j is in the failed state in the current time-step
IPR	Marker for whether part j is replaced preventively in the current time-step
ICR	Marker for whether part j is replaced correctively in the current time-step
IR	Marker for whether part j is replaced in the current time-step
IM	Marker for whether part j is minimally repaired in the current time-step
IN	Marker for whether part j requires inspection in the future
OC	Marker for whether an order for spare parts is outstanding or not
IMC	Counter of number of minimal maintenance actions per period
$LastOT$	The last time a new order was placed for spare parts
NA	The number of spare parts that arrive in the current time-step
SL	The Inventory On Hand for spare parts
ST_j	The time-step in which part j was installed (start of new lifetime)
C_i	Cost of inspection
C_p	Cost of preventive replacement
C_c	Cost of corrective replacement
C_d	Cost of downtime (per part per period)
C_s	Cost of procurement of spare part
C_h	Holding cost of spare stock (per part per period)
C_m	Cost of minimal maintenance
N_i	Number of inspections performed
N_p	Number of preventive replacements performed
N_c	Number of corrective replacements performed
N_d	Number of periods of downtime (cumulative)
N_s	Number of spares procured
N_h	Number of spares in warehouse (cumulative)
N_m	Number of minimal maintenance tasks performed
j	Index, j = 1,2,...,n
t	Current time-step
t_m	Simulation horizon
MC	Number of Monte Carlo iterations

TABLE VI
NOMENCLATURE FOR SIMULATION MODEL

Algorithm 1 Simulation Model for finding the objective value (estimated cost)

Require: C_i, C_p, C_c, C_h, C_s, C_d, C_m, t_s, T, n, t_m, L_p, S, s, MC ▷ ***Data entry***
Ensure: EC
Count := 0
Cost := 0
while Count $\leq MC$ **do**
  $LastOT := 0$, $OC := 0$, $NA := 0$, $SL := S$, $t := 0$ ▷ ***Initialization***
  $N_i := 0$, $N_p := 0$, $N_c := 0$, $N_m := 0$, $N_s := 0$, $N_d := 0$, $N_h := 0$
  for j = 1 to n **do**
    $F_j := 0$, $I_j := 0$, $ICR_j := 0$, $IPR_j := 0$, $IR_j := 0$, $IM_J := 0$, $ST_j := 0$, $IN_j := 1$
    Failure threshold := $X_{f,j}$ ▷ $P[X_{f,j} = x] = f^F(x)$
    Deterioration level := $x_j(t)$
    $x_j(t) = X_{f,j} \cdot U(0, 1)$
  end for
  while $t \leq t_m$ **do**
    if $t - LastOT = t_s$ and $OC = 1$ **then** ▷ ***Spare order handling***
      $NA = S - s$
      $OC = 0$
    else
      $NA = 0$
    end if
    $SL := SL + NA$
    $q := \{1, 2, ..., n-1, n\}$
    while $q \neq \{\ \}$ **do**
      $j \in q$
      $q := q - j$ ▷ ***Deterioration and failure***
      $x_j(t) := x_j(t-1) + \Delta x$ ▷ $P[\Delta x = x] = f^D(x)$
      if $x_j(t) \geq X_{f,j}$ **then**
        $F_j := 1$
      else
        $F_j := 0$
      end if
      if $IN_j = 0$ **then** ▷ ***No-more-inspection clause***
        if $F_j = 1$ and $SL > 0$ **then**
          $ICR_j := 1$
        else if $F_j = 0$ and $SL > 0$ **then**
          $IPR_j := 1$
        end if
      else
        if $F_j = 1$ and $SL > 0$ **then** ▷ ***Corrective replacement or minimal repair***
          $ICR_j := 1$
        else if $F_j = 1$ and $SL \leq 0$ **then**
          $IN_j := 0$
          if $IM_j = 0$ **then**
            $IM_j := 1$
            $IMC := IMC + 1$
            $F_j := 0$
            while $x_j(t) > X_{f,j}$ **do**
              $X_{f,j} := \{X \mid P[X = x] = f^F(x)\}$
            end while
          end if
        else
          if $ST_j \neq t$ and $(t - ST_j) \mod T \equiv 0$ **then** ▷ ***Inspection and preventive replacement***
            $I_j := 1$
            if $x_j(t) \geq L_p$ **then**
              if $SL > 0$ **then**
                $IPR_j := 1$
              else
                $IN_j := 0$
              end if
            end if
          else
            $I_j := 0$

Algorithm 2 Simulation Model for finding the objective value (estimated cost) (continued)

61: **end if**
62: **end if**
63: **end if**
64: **if** $IPR_j = 1$ or $ICR_j = 1$ **then** ▷ ***Installation of new part***
65: IR_j := 1
66: $x_j(t)$:= 0
67: $ST_j := t$
68: $X_{f,j} := \{X \mid P[X = x] = f^F(x)\}$
69: IM_j := 0
70: IN_j := 1
71: $SL := SL - 1$
72: **else**
73: IR_j := 0
74: **end if**
75: **if** $SL \leq s$ and $OC = 0$ **then** ▷ ***Spare ordering decision***
76: OC := 1
77: $LastOT := t$
78: **end if**
79: **end while**
80: $N_i := N_i + \sum_{j=0}^{n} I_j$ ▷ ***Counting, resetting and advancing time***
81: $N_p := N_p + \sum_{j=0}^{n} IPR_j$
82: $N_c := N_c + \sum_{j=0}^{n} ICR_j$
83: $N_s := N_i + NA$
84: $N_d := N_d + \sum_{j=0}^{n} F_j$
85: $N_h := N_h + SL$
86: $N_m := N_m + IMC$
87: **for** j = 1 to n **do**
88: F_j := 0, I_j := 0, ICR_j := 0, IPR_j := 0, IMC := 0, IR_j := 0
89: **end for**
90: $t := t + 1$
91: **end while**
92: Cost = Cost $+C_iN_i + C_pN_p + C_cN_c + C_hN_h + C_sN_s + C_dN_d + C_mN_m$ ▷ ***Cost calculation***
93: Count = Count + 1
94: $EC_{MC} := EC_{MC} + \frac{\text{Cost}}{t_m \cdot n}$
95: **end while**
96: $EC = EC_{MC}/MC$;

B

Data sources

A multitude of data sources has been used during this research, most pertaining to failure data or historic figures on demand. While some of this data is confidential, all data sources are listed with a contact of the owner for the sake of reproducibility of this research.

Table B.1: Data sources

	Data used	**Data source**	**Contact**
1	Failure data; Repair times & traffic hinderance probability	Storingen Rapportage	Mark Grashoff (mark.grashoff@prorail.nl)
2	Demand data; Monthly demand, part costs & lead times	PSSSL componenten	Mark Grashoff (mark.grashoff@prorail.nl)
3	Inspection intervals	FMECA analyse	Boy van den Ende (boy.vandenende@prorail.nl
4	Rail section values	Baanvakwaardes	Mark Grashoff (mark.grashoff@prorail.nl
5	Failure repair times	Storingen Rapportage	Mark Grashoff (mark.grashoff@prorail.nl)

C

Experiment data

Sensitivity analysis

Table C.1: Experiments for sensitivity analysis

Test	C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	L_p	S	s	n	Weibull (scale)	Weibull (shape)	Gamma (shape)	Gamma (scale)	t_m	Average costrate (Objective)
C_i	10	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.70
C_i	20	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.87
C_i	30	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	198.11
C_i	40	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.55
C_i	50	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.78
C_i	60	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	202.43
C_i	70	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.81
C_i	80	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	202.36
C_i	90	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.23
C_i	100	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.67
C_p	55	1	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	198.56
C_p	55	133	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.74
C_p	55	267	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	198.83
C_p	55	399	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.63
C_p	55	532	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	202.93
C_p	55	665	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	198.79
C_p	55	798	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.94
C_p	55	930	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	200.42
C_p	55	1036	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.71
C_p	55	1196	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.24
C_c	55	598.5	1	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	202.62
C_c	55	598.5	131	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.70
C_c	55	598.5	261	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	202.14
C_c	55	598.5	389	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.35
C_c	55	598.5	521	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.29
C_c	55	598.5	650	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.44
C_c	55	598.5	780	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	200.94
C_c	55	598.5	910	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.62
C_c	55	598.5	1040	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	200.19
C_c	55	598.5	1170	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.33
C_d	55	598.5	585.5	72360	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	186.56
C_d	55	598.5	585.5	241200	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	191.19
C_d	55	598.5	585.5	410040	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	192.01
C_d	55	598.5	585.5	578880	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	196.21
C_d	55	598.5	585.5	747720	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	200.06
C_d	55	598.5	585.5	916560	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.34
C_d	55	598.5	585.5	1085400	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	208.30
C_d	55	598.5	585.5	1254240	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	211.28
C_d	55	598.5	585.5	1423080	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	212.08
C_d	55	598.5	585.5	1591920	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	214.28
C_s	55	598.5	585.5	832140	10	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	114.39
C_s	55	598.5	585.5	832140	11120	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	135.04
C_s	55	598.5	585.5	832140	22230	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	152.52
C_s	55	598.5	585.5	832140	33340	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	173.00
C_s	55	598.5	585.5	832140	44450	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	191.06
C_s	55	598.5	585.5	832140	55560	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	210.91
C_s	55	598.5	585.5	832140	66670	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	227.84
C_s	55	598.5	585.5	832140	77780	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	251.36
C_s	55	598.5	585.5	832140	88890	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	269.22
C_s	55	598.5	585.5	832140	100000	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	285.41

Table C.2: Experiments for sensitivity analysis (continued)

Test	C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	L_p	S	s	n	Weibull (scale)	Weibull (shape)	Gamma (shape)	Gamma (scale)	t_m	Average costrate (Objective)
C_h	55	598.5	585.5	832140	50005	0.18	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	104.14
C_h	55	598.5	585.5	832140	50005	203.87	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	126.42
C_h	55	598.5	585.5	832140	50005	407.55	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	149.21
C_h	55	598.5	585.5	832140	50005	611.23	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	168.12
C_h	55	598.5	585.5	832140	50005	814.92	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	194.12
C_h	55	598.5	585.5	832140	50005	1018.6	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	210.06
C_h	55	598.5	585.5	832140	50005	1222.28	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	232.04
C_h	55	598.5	585.5	832140	50005	1425.98	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	255.10
C_h	55	598.5	585.5	832140	50005	1629.65	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	276.73
C_h	55	598.5	585.5	832140	50005	1833.33	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	300.32
C_m	55	598.5	585.5	832140	50005	916.76	2	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.09
C_m	55	598.5	585.5	832140	50005	916.76	267	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.99
C_m	55	598.5	585.5	832140	50005	916.76	533	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	204.33
C_m	55	598.5	585.5	832140	50005	916.76	798	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.31
C_m	55	598.5	585.5	832140	50005	916.76	1036	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	206.10
C_m	55	598.5	585.5	832140	50005	916.76	1330	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.51
C_m	55	598.5	585.5	832140	50005	916.76	1595	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.44
C_m	55	598.5	585.5	832140	50005	916.76	1860	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	204.63
C_m	55	598.5	585.5	832140	50005	916.76	2127	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	200.71
C_m	55	598.5	585.5	832140	50005	916.76	2392	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.45
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	1	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.19
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	2	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.36
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	3	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	198.50
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	5	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.30
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	6	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.93
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.90
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	8	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	200.64
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	10	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	204.46
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	11	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	212.50
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	12	37	693	6	0	30	1743.979	23.403	0.6	1.667	3395	215.22
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	3	693	6	0	30	1743.979	23.403	0.6	1.667	3395	210.01
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	11	693	6	0	30	1743.979	23.403	0.6	1.667	3395	196.36
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	18	693	6	0	30	1743.979	23.403	0.6	1.667	3395	199.36
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	26	693	6	0	30	1743.979	23.403	0.6	1.667	3395	200.49
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	34	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.41
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	41	693	6	0	30	1743.979	23.403	0.6	1.667	3395	201.57
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	49	693	6	0	30	1743.979	23.403	0.6	1.667	3395	203.14
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	57	693	6	0	30	1743.979	23.403	0.6	1.667	3395	207.23
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	64	693	6	0	30	1743.979	23.403	0.6	1.667	3395	206.12
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	72	693	6	0	30	1743.979	23.403	0.6	1.667	3395	207.02
C_i	10	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	228.95
C_i	20	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.79
C_i	30	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.16
C_i	40	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.95
C_i	50	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.49
C_i	60	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.06
C_i	70	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.28
C_i	80	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.93
C_i	90	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	233.93
C_i	100	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.94
C_p	55	1	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.64
C_p	55	133	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	228.82
C_p	55	267	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.39
C_p	55	399	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.62
C_p	55	532	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.58
C_p	55	665	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.02
C_p	55	798	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.70
C_p	55	930	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.45
C_p	55	1036	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	233.58
C_p	55	1196	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	234.77
C_c	55	598.5	1	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.75
C_c	55	598.5	131	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.47
C_c	55	598.5	261	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	233.51
C_c	55	598.5	389	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.30
C_c	55	598.5	521	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.81
C_c	55	598.5	650	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.19
C_c	55	598.5	780	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.86
C_c	55	598.5	910	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.84
C_c	55	598.5	1040	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	233.99
C_c	55	598.5	1170	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.15
C_d	55	598.5	585.5	72360	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	187.46
C_d	55	598.5	585.5	241200	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	197.80
C_d	55	598.5	585.5	410040	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	206.66
C_d	55	598.5	585.5	578880	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	217.00
C_d	55	598.5	585.5	747720	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	225.50
C_d	55	598.5	585.5	916560	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	235.18
C_d	55	598.5	585.5	1085400	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	248.01
C_d	55	598.5	585.5	1254240	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	253.52
C_d	55	598.5	585.5	1423080	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	267.15
C_d	55	598.5	585.5	1591920	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	274.47

Table C.3: Experiments for sensitivity analysis (continued)

Test	C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	L_p	S	s	n	Weibull (scale)	Weibull (shape)	Gamma (shape)	Gamma (scale)	t_m	Average costrate (Objective)
C_s	55	598.5	585.5	832140	10	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	148.85
C_s	55	598.5	585.5	832140	11120	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	168.20
C_s	55	598.5	585.5	832140	22230	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	184.07
C_s	55	598.5	585.5	832140	33340	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	204.23
C_s	55	598.5	585.5	832140	44450	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	225.04
C_s	55	598.5	585.5	832140	55560	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	240.56
C_s	55	598.5	585.5	832140	66670	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	257.00
C_s	55	598.5	585.5	832140	77780	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	275.43
C_s	55	598.5	585.5	832140	88890	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	294.76
C_s	55	598.5	585.5	832140	100000	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	313.39
C_h	55	598.5	585.5	832140	50005	0.18	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	132.92
C_h	55	598.5	585.5	832140	50005	203.87	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	154.85
C_h	55	598.5	585.5	832140	50005	407.55	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	173.49
C_h	55	598.5	585.5	832140	50005	611.23	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	198.01
C_h	55	598.5	585.5	832140	50005	814.92	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	224.53
C_h	55	598.5	585.5	832140	50005	1018.6	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	244.36
C_h	55	598.5	585.5	832140	50005	1222.28	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	262.87
C_h	55	598.5	585.5	832140	50005	1425.98	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	284.32
C_h	55	598.5	585.5	832140	50005	1629.65	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	306.78
C_h	55	598.5	585.5	832140	50005	1833.33	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	330.58
C_m	55	598.5	585.5	832140	50005	916.76	2	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.63
C_m	55	598.5	585.5	832140	50005	916.76	267	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.48
C_m	55	598.5	585.5	832140	50005	916.76	533	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	227.86
C_m	55	598.5	585.5	832140	50005	916.76	798	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.87
C_m	55	598.5	585.5	832140	50005	916.76	1036	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.80
C_m	55	598.5	585.5	832140	50005	916.76	1330	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.80
C_m	55	598.5	585.5	832140	50005	916.76	1595	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.94
C_m	55	598.5	585.5	832140	50005	916.76	1860	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.27
C_m	55	598.5	585.5	832140	50005	916.76	2127	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.53
C_m	55	598.5	585.5	832140	50005	916.76	2392	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.29
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	1	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	235.08
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	2	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.13
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	3	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	232.07
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	5	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	228.79
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	6	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	229.18
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	231.69
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	8	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	230.54
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	10	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	236.48
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	11	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	234.97
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	12	37	693	6	0	30	871.984	23.406	0.6	1.667	1690	235.68
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	3	693	6	0	30	871.984	23.406	0.6	1.667	1690	211.09
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	11	693	6	0	30	871.984	23.406	0.6	1.667	1690	206.31
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	18	693	6	0	30	871.984	23.406	0.6	1.667	1690	209.70
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	26	693	6	0	30	871.984	23.406	0.6	1.667	1690	220.45
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	34	693	6	0	30	871.984	23.406	0.6	1.667	1690	228.86
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	41	693	6	0	30	871.984	23.406	0.6	1.667	1690	237.13
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	49	693	6	0	30	871.984	23.406	0.6	1.667	1690	245.81
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	57	693	6	0	30	871.984	23.406	0.6	1.667	1690	257.18
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	64	693	6	0	30	871.984	23.406	0.6	1.667	1690	265.79
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	72	693	6	0	30	871.984	23.406	0.6	1.667	1690	282.53
C_i	10	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2494.46
C_i	20	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2489.85
C_i	30	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2483.84
C_i	40	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2491.49
C_i	50	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2501.69
C_i	60	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2493.12
C_i	70	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2499.07
C_i	80	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2491.42
C_i	90	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2489.75
C_i	100	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2502.51
C_p	55	1	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2483.59
C_p	55	133	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2483.57
C_p	55	267	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2497.22
C_p	55	399	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2491.51
C_p	55	532	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2492.00
C_p	55	665	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2497.86
C_p	55	798	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2478.30
C_p	55	930	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2483.21
C_p	55	1036	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2485.86
C_p	55	1196	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2493.51
C_c	55	598.5	1	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2491.21
C_c	55	598.5	131	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2510.91
C_c	55	598.5	261	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2488.39
C_c	55	598.5	389	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2484.50
C_c	55	598.5	521	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2490.28
C_c	55	598.5	650	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2485.66
C_c	55	598.5	780	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2502.61
C_c	55	598.5	910	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2480.93
C_c	55	598.5	1040	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2505.85
C_c	55	598.5	1170	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2494.27

Table C.4: Experiments for sensitivity analysis (continued)

Test	C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	L_p	S	s	n	Weibull (scale)	Weibull (shape)	Gamma (shape)	Gamma (scale)	t_m	Average costrate (Objective)
C_d	55	598.5	585.5	72360	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	426.89
C_d	55	598.5	585.5	241200	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	889.75
C_d	55	598.5	585.5	410040	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	1341.45
C_d	55	598.5	585.5	578880	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	1808.24
C_d	55	598.5	585.5	747720	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2252.80
C_d	55	598.5	585.5	916560	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2718.87
C_d	55	598.5	585.5	1085400	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	3190.74
C_d	55	598.5	585.5	1254240	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	3633.97
C_d	55	598.5	585.5	1423080	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	4117.53
C_d	55	598.5	585.5	1591920	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	4568.09
C_s	55	598.5	585.5	832140	10	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2364.45
C_s	55	598.5	585.5	832140	11120	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2380.01
C_s	55	598.5	585.5	832140	22230	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2418.00
C_s	55	598.5	585.5	832140	33340	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2445.86
C_s	55	598.5	585.5	832140	44450	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2489.79
C_s	55	598.5	585.5	832140	55560	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2515.46
C_s	55	598.5	585.5	832140	66670	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2550.56
C_s	55	598.5	585.5	832140	77780	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2563.57
C_s	55	598.5	585.5	832140	88890	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2598.74
C_s	55	598.5	585.5	832140	100000	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2635.99
C_h	55	598.5	585.5	832140	50005	0.18	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2404.92
C_h	55	598.5	585.5	832140	50005	203.87	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2417.05
C_h	55	598.5	585.5	832140	50005	407.55	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2449.42
C_h	55	598.5	585.5	832140	50005	611.23	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2468.61
C_h	55	598.5	585.5	832140	50005	814.92	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2484.89
C_h	55	598.5	585.5	832140	50005	1018.6	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2503.46
C_h	55	598.5	585.5	832140	50005	1222.28	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2525.69
C_h	55	598.5	585.5	832140	50005	1425.98	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2542.01
C_h	55	598.5	585.5	832140	50005	1629.65	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2563.28
C_h	55	598.5	585.5	832140	50005	1833.33	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2581.53
C_m	55	598.5	585.5	832140	50005	916.76	2	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2499.26
C_m	55	598.5	585.5	832140	50005	916.76	267	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2486.80
C_m	55	598.5	585.5	832140	50005	916.76	533	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2483.53
C_m	55	598.5	585.5	832140	50005	916.76	798	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2491.97
C_m	55	598.5	585.5	832140	50005	916.76	1036	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2496.55
C_m	55	598.5	585.5	832140	50005	916.76	1330	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2492.35
C_m	55	598.5	585.5	832140	50005	916.76	1595	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2492.52
C_m	55	598.5	585.5	832140	50005	916.76	1860	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2496.80
C_m	55	598.5	585.5	832140	50005	916.76	2127	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2496.87
C_m	55	598.5	585.5	832140	50005	916.76	2392	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2491.38
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	1	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2575.45
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	2	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2545.70
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	3	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2520.01
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	5	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2498.18
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	6	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2484.95
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	7	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2484.52
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	8	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2516.26
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	10	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2591.98
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	11	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2633.68
t_s	55	598.5	585.5	832140	50005	916.76	1196.5	12	37	693	6	0	30	435.996	23.409	0.6	1.667	849	2725.93
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	3	693	6	0	30	435.996	23.409	0.6	1.667	849	2511.99
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	11	693	6	0	30	435.996	23.409	0.6	1.667	849	2498.66
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	18	693	6	0	30	435.996	23.409	0.6	1.667	849	2495.62
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	26	693	6	0	30	435.996	23.409	0.6	1.667	849	2493.16
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	34	693	6	0	30	435.996	23.409	0.6	1.667	849	2487.37
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	41	693	6	0	30	435.996	23.409	0.6	1.667	849	2496.95
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	49	693	6	0	30	435.996	23.409	0.6	1.667	849	2488.56
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	57	693	6	0	30	435.996	23.409	0.6	1.667	849	2492.60
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	64	693	6	0	30	435.996	23.409	0.6	1.667	849	2494.72
T	55	598.5	585.5	832140	50005	916.76	1196.5	7	72	693	6	0	30	435.996	23.409	0.6	1.667	849	2482.82

Algorithm choice

Table C.5: Algorithm choice experiments

Algorithm	C_i	C_p	C_c	C_d	C_s	C_h	C_m	t_s	T	**Weibull**		**Gamma**		t_m	n	**Decision variables**			**Objective**
										(scale)	**(shape)**	**(scale)**	**(shape)**			S	s	L_p	
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	4	2	670	39.578
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	9	1	410	47.095
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	5	3	646	43.334
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	12	1	409	54.571
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	8	1	477	41.923
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	4	2	590	41.839
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	15	0	596	46.523
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	9	2	676	49.922
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	1	30	134	53.517
Randomsearch	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	5	4	527	52.856
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	1	596	36.092
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	5	1	674	41.524
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	5	0	656	32.963
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	12	0	614	41.689
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	9	0	652	38.467
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	5	0	594	35.847
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	4	0	632	38.623
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	1	644	38.644
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	2	648	39.730
HyperOpt	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	3	1	617	38.418
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	0	434	50.161
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	15	0	323	51.634
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	30	0	415	58.633
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	10	2	186	72.487
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	3	623	44.123
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	8	0	627	33.746
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	7	4	659	51.348
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	1	480	38.680
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	13	0	389	53.921
IDONE	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	0	697	32.962
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	0	624	32.257
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	7	0	350	43.395
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	7	0	488	36.405
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	8	0	492	37.880
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	4	0	659	31.261
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	15	0	348	49.835
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	6	1	602	34.815
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	7	0	669	32.666
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	10	0	514	39.780
MVRSM	50	224	140	832140	5500	100.83	448	6	12	871.984	23.406	1.667	0.6	1690	30	4	0	690	35.903

D

Derivation of Expected Improvement optimisation formula

This appendix details the derivation of the Expected Improvement formulas which is used in the Tree-structured Parzen Estimator algorithm described by Bergstra, Bardenet, Bengio, and Kégl (2011).

As stated in chapter 4, by splitting the data available to the TPE algorithm into two sections by choosing a desired performance for the objective value y^*, the probability of the decision variables having a certain value given a value of the objective can be formulated (see Eq. (4.2)):

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \tag{D.1}$$

This desired performance y^* can be chosen by determining the ratio of data which the user would like to fall above or below this desired performance. In other words, by choosing γ where

$$\gamma = p(y < y^*) \tag{D.2}$$

By construction, using Eq.(D.1) the probabilities for the decision variables x can be described by

$$p(x) = \int p(x|y)p(y)dy = \gamma l(x) + (1-\gamma)g(x) \tag{D.3}$$

Then Eq.(4.3) can be re-written using Eq.(D.3) as follows:

$$\begin{aligned} EI_{y^*}(x) &= \int_{-\infty}^{y^*} (y^* - y)\frac{p(x|y)p(y)}{p(x)}dy \\ &= \frac{1}{p(x)}\int_{-\infty}^{y^*} (y^* - y)p(x|y)p(y)dy \\ &= \frac{1}{\gamma l(x) + (1-\gamma)g(x)}\int_{-\infty}^{y^*} (y^* - y)p(x|y)p(y)dy \end{aligned} \tag{D.4}$$

Then because the limits of the integral in Eq.(D.4) do not reach beyond y^* the expression $p(x|y)$ within the integral only takes the value of the case where $y < y^*$ (see Eq.(D.1) which leads to:

$$
\begin{aligned}
EI_{y^*}(x) &= \frac{1}{\gamma l(x) + (1-\gamma) g(x)} \int_{-\infty}^{y^*} (y^* - y) l(x) p(y) dy \\
&= \frac{l(x)}{\gamma l(x) + (1-\gamma) g(x)} \int_{-\infty}^{y^*} (y^* - y) p(y) dy \\
&= \frac{l(x)}{\gamma l(x) + (1-\gamma) g(x)} \left(\int_{-\infty}^{y^*} y^* p(y) dy - \int_{-\infty}^{y^*} y p(y) dy \right) \\
&= \frac{l(x)}{\gamma l(x) + (1-\gamma) g(x)} \left(y^* \int_{-\infty}^{y^*} p(y) dy - \int_{-\infty}^{y^*} y p(y) dy \right)
\end{aligned}
\tag{D.5}
$$

Since the definition in Eq.(D.2) $\gamma = p(y < y^*) = \int_{-\infty}^{y^*} p(y) dy$:

$$
\begin{aligned}
EI_{y^*}(x) &= \frac{l(x)}{\gamma l(x) + (1-\gamma) g(x)} \left(\gamma y^* - \int_{-\infty}^{y^*} y p(y) dy \right) \\
&= \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} y p(y) dy}{\gamma l(x) + (1-\gamma) g(x)} \\
&= \frac{\gamma y^* - \int_{-\infty}^{y^*} y p(y) dy}{\gamma + (1-\gamma) \frac{g(x)}{l(x)}} \\
&= \left(\gamma y^* - \int_{-\infty}^{y^*} y p(y) dy \right) \left(\gamma + (1-\gamma) \frac{g(x)}{l(x)} \right)^{-1}
\end{aligned}
\tag{D.6}
$$

Because the values of y^* and thereby γ are chosen by the user and will not change during an iteration of the algorithm, the algorithm can only influence this expression by changing the values it chooses for the decision variables which in turn influence $g(x)$ and $l(x)$. This means that:

$$
EI_{y^*}(x) \propto \left(\gamma + (1-\gamma) \frac{g(x)}{l(x)} \right)^{-1} \tag{D.7}
$$

And in order to optimise (max) $EI_{y^*}(x)$ the algorithm will aim to choose decision variable values which have a higher likelihood to fall within $l(x)$ than $g(x)$.