Minimizing utilization error of reserve crew by optimizing scheduling at NS

Summary of confidential thesis

Student:	Luuk Nijs
First supervisor:	dr. S.M. Meisel
Second supervisor:	dr. I. Seyran Topan
Company supervisor:	M. Jacobs
Date:	30-06-2024

Table of contents

Table of figures 2

Table of tables 3

1 Introduction 4

2 Problem context and identification 5

3 Rail crew scheduling concepts 7

4 Design of the Metamodel 8

5 Validation and experiment design 11

5.1 Validation 11

5.2 Simulated annealing parameter selection 11

5.3 Experiment design 14

6 Experiments 15

7 Conclusion 16

8 Bibliography 17

Colophon 20

Table of figures

Figure 1 Reserve types and their schedule 5
Figure 3 Formulation of metamodel 8
Figure 4 Starting temperature determination 12
Figure 5 Cooling Scheme (case 1) 13
Figure 6 Cooling scheme (case 2) 13
Figure 7 Best solution graph (case 2) 13
Figure 8 Current solution graph (case 3) 14

Table of tables

Table 1 Variables machine learning model 9
Table 2 Validation on test dataset 11
Table 3 Results of experiments 15

1 Introduction

NS, short for "Nederlandse Spoorwegen" traces its roots back to 1837 with the founding of 'de Hollandsche Ijzeren Spoorweg Maatschappij (HIJSM)'. It underwent several evolutions, merging in 1937 to become the primary rail service provider in the Netherlands (NS, NS Geschiedenis, 2023).

In 2022, NS facilitated 960,000 passenger journeys daily, totalling 437 million passengers for the year, using 761 trainsets—383 'sprinters' and 378 'inter-city' trains. NS had a turnover of €3.055 million in 2022 (NS, NS Jaarverslag, 2022).

This assignment is carried out within the Data, Innovation, and Analysis (DIA) subdepartment, specifically the Driving Personnel team, which supports Operations with data insights and driver/conductor scheduling. NS employs around 18,800 individuals: 12,900 in Operations, 700 in Commercial and Development, 2,300 in Staff and Central Services, and 2,900 in Stations.

NS wants to provide the best train service possible by balancing their resources with demand. To support stability of this balance, the goal is to minimize disruptions and maximize the number of trains driving according to the timetable. NS has several systems for material and personnel to support this, we focus on the personnel.

The department of Transport Control (TB) manages the operations on the day before and the day of execution. So-called 'TB reserve personnel' is present on certain locations throughout the network. Together with rescheduling regular tasks within shifts, TB minimizes the effects of unexpected staff shortages. When for example a conductor is sick, TB cannot always find a substitute conductor to fulfil all its tasks. Not filling these tasks leads to cancelling train services from the timetable. This has large effects on the whole network and leads to losses on financial aspect and passenger experience.

For this problem, NS has a performance measure: the utilization of a TB reserve shift. The target of this performance measure is what NS agrees to be a good utilization. The error is the absolute difference between the target and actual utilization. Together with history data, this target value is used in creating next year's schedule. NS aims at minimizing TB utilization error. Therefore, the problem statement is formulated as:

Actual utilization and target utilization of the TB reserve differ by X% for drivers and X% for conductors, NS wants to reduce these discrepancies.

To do so the schedule needs to be optimized. A schedule consists of reserve shift quantities per day, per shift type (early, late, night), per function, and per location. Currently this is a standard week schedule for the whole year (e.g. every Monday night in 2023 there was 1 TB reserve conductor in Amsterdam). However, since the timetable is updated 6 times per year and the TB reserve schedule is not, this research also aims at investigating the impact of reducing this scheduling horizon updated timetables.

Therefore, the objective of this research is:

Minimizing TB reserve utilization error by changing the TB reserve crew schedule and by reducing the scheduling horizon.

2 Problem context and identification

NS's scheduling process is divided into four key components: timetable, material, node, and personnel, which operate in a hierarchical and iterative manner.

- **Timetable:**

  Planning of train services and times.
- **Material:**

  Scheduling rolling stock to meet the timetable.
- **Node:**

  Managing the parking and turnaround of rolling stock.
- **Personnel:**

  Assigning staff to shifts and specific tasks.

The scheduling is performed in multiple phases, ranging from long-term strategic planning (beyond 5 years) to daily updates (Specific Day phase). The entire process culminates in the Donderdagse Week (DW) schedule, which is shared with personnel the Thursday before the execution week. The department of Network Design and Operations (NO) handles long-term timetabling and shifts. The Preparation and Adjust (V&B) department adapts and executes the timetable. Service and Operations (S&O) oversee the entire scheduling process up to the day before execution, when control is handed over to Transport Control (TB).

NS employs three types of personnel reserves to manage disruptions:

- **S&O Reserve:** Used for long-term issues like illnesses and regulatory time off, planned annually within the employability budget.
- **V&B Reserve:** Scheduled for events and construction works, with extra shifts requested as needed.
- **TB Reserve:** Used for unexpected disruptions, managed by TB for issues occurring from the night before to the day of execution.

Each type of reserve has a specific timeline and scheduling phase where they are "locked" for use by the relevant department. Figure 1 depicts the reserves and their timeline.

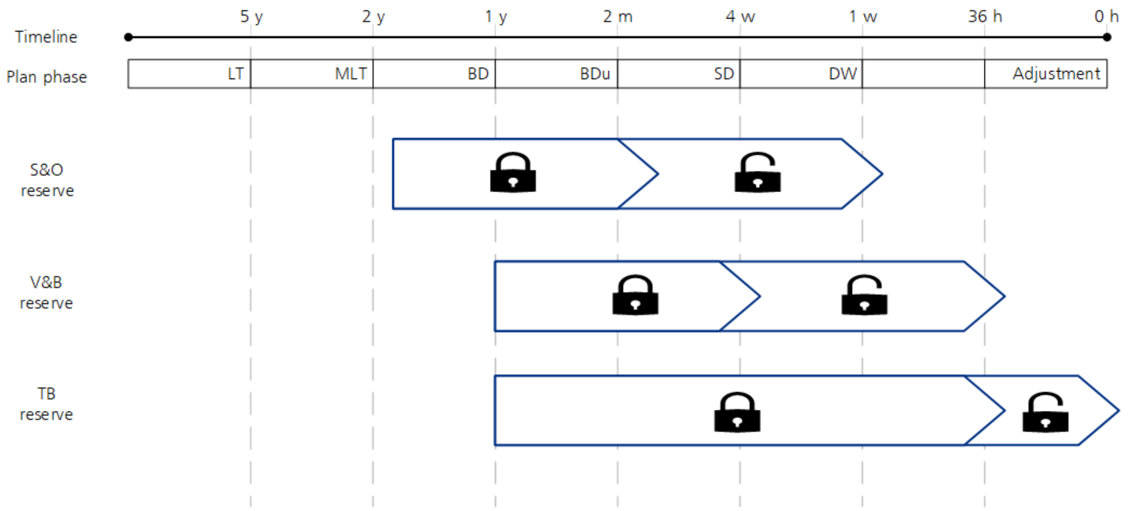

Figure 1 Reserve types and their schedule

With scheduling personnel reserve, several problems exist. They have been discovered by talking to company experts on this matter from the corresponding departments. The specifics of these challenges are linked to each reserve type, affecting their overall effectiveness. Due to confidentiality, we cannot mention these.

To narrow down the research scope, the focus was directed specifically at TB reserve issues for the following reasons:

- S&O and V&B Reserves:

  Addressing challenges in these reserves was deemed too extensive for a master's thesis due to their organizational interconnections and the resources required for comprehensive analysis.

- TB Reserve:

  The challenges associated with TB reserve were more confined and manageable, making them suitable for in-depth analysis within the constraints of this research.

The focus of this research is on addressing these issues to improve the utilization rates of TB reserves. The current TB reserve scheduling is handled annually by the business owner using a model developed by Capgemini. The process includes summarizing total shift hours, calculating utilization rates, and setting target utilization based on historical data.

The primary performance measure for TB reserves is the utilization rate, defined as the percentage of time reserve shifts are engaged in 'useful' tasks such as driving and shunting. This measure is crucial because it reflects how effectively TB reserves are being used to handle disruptions. Despite considering other performance measures like train cancellations due to personnel shortages, utilization remains the most practical metric given current data limitations.

3 Rail crew scheduling concepts

Extensive research exists on railway optimization focusing on timetabling and rolling stock. However, railway crew optimization, particularly scheduling and rescheduling, has been less explored. Heil, Hoffman, & Buscher (2020). distinguish the planning of crew scheduling into three phases: strategic and tactical crew management, operational crew scheduling, and real-time crew rescheduling. Understanding this progression is crucial to position the TB reserve scheduling within the broader context. Heil, Hoffman, & Buscher delineate between crew scheduling (short-term tactical planning) and crew rostering (long-term strategic planning). The integration of these aspects can yield better results but is complex. The review by Cacchiani, et al. (2014) on real-time rescheduling provides insight into managing unforeseen incidents but does not specifically address reserve crew quantities.

The literature on reserve crew scheduling is scarce, particularly for railways. Some insights can be drawn from the airline industry, where models estimate necessary reserve crew based on probabilistic methods or machine learning techniques. However, these models rely on data and variables that NS currently lacks, making direct application challenging.

Our aim is to adapt the TB reserve crew schedule in a manner that ensures utilization aligns as closely as possible to the target, thereby minimizing utilization error. Various scheduling approaches have been explored, but none fully aligned with our research objective.

We are therefore combining two concepts to reach our goal of estimating the needed reserve shifts. This approach involves using a model within a model, also known as a metamodel. By definition, a metamodel mimics a model while being inexpensive to evaluate, in contrast to the original model. This is used when the original model is too complex or too computationally challenging to solve within an acceptable amount of time (Sudret, 2012). Like models are abstractions of reality, metamodels are abstractions of models (Jeusfeld, 2009).

To achieve the desired utilization targets, the number of scheduled reserve shifts needs to be adjusted. We do that in a mathematical optimization model. To evaluate the quality of the solution of the mathematical optimization model, we use machine learning inside the objective function to predict a variable. By integrating the predictive model in a mathematical optimization model, we can dynamically adjust the schedule to target at the least possible error. This results in a metamodel that conceptually can solve our problem case. Figure 2 depicts how the metamodel works.

The input of this mathematical optimization model encompasses all potential TB reserve schedules. However, the complexity of the solution space necessitates the use of metaheuristics to approximate optimal solutions efficiently. Metaheuristics are advanced methods for solving complex optimization problems. We compared several metaheuristic approaches: Greedy Randomized Adaptive Search Procedure (GRASP), Genetic Algorithms, Simulated Annealing, and Tabu Search. Each method has its strengths and weaknesses.

Ultimately, Simulated Annealing is chosen due to its balance of simplicity, global optimization capability, and ease of implementation. Genetic algorithms, while also effective, are more challenging to implement and parameterize, making Simulated Annealing the preferred choice for integrating with the existing machine learning model.

4 Design of the Metamodel

This chapter elaborates on how we will use the metamodel. Figure 2 visually presents the use of the metamodel and how the different parts work together.

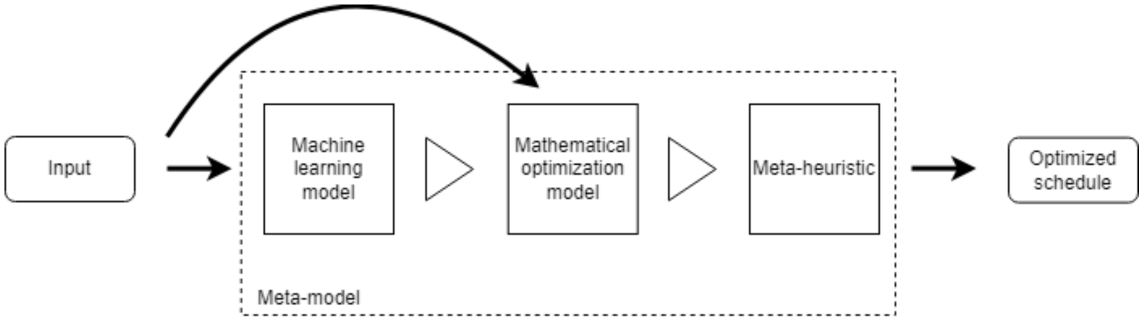

Figure 2 Formulation of metamodel

Below we elaborate of the use of each part of Figure 2:

- *Input*

  Consists of history data and targets. History data is used as training data for the machine learning model and the schedule we want to optimize in the metaheuristics.

- *Machine learning model*

  The machine learning model is targeted at predicting the utilization based on an input schedule. *Input:*

  Schedule (timetable) data with regular shift minutes, tb-reserve shift minutes, and TB utilization for 2022, 2023, and 2024.

  Output:
  Machine learning model able to predict TB utilization based on schedule with regular shift minutes.

- *Mathematical optimization model*

  The mathematical optimization model is targeted at translating the problem of scheduling shifts into a mathematical formulation. This is necessary for implementation in the metaheuristic.

- *Metaheuristic*

  The metaheuristic is targeted at improving a schedule. The metaheuristic uses the machine learning model’s prediction to calculate the performance measure. By changing parameters in the schedule of the machine learning model and followingly evaluating the performance measure, the metaheuristic can find improvements in schedules.

  Input:
  Machine learning model and timetable with regular shifts minutes that needs to be optimized. *Output:*
  Optimized schedule

- *Optimized schedule*

  The optimized schedule, which started as the schedule we wanted to optimize, has gone through the metamodel and underwent changes to improve its performance according to the performance measure.

To forecast the utilization of the TB reserve schedule, we train a machine learning model. It is tasked with predicting TB reserve utilization based on the input variable explained in Table 1. These parameters are based on the available data and parameters we want to consider. The first seven are the dependent variables and the last is the independent variable.

Table 1 Variables machine learning model

Variable	Explanation
Function	Type of function (e.g. driver or conductor)
Day	Day of the week
Plan phase BD	The planning phase, which BD is active
Location	TB reserve location
Shift-type	Part of the day
Planned regular shift minutes	The total sum of planned regular shift minutes.
Planned TB reserve shift minutes	The total sum of planned TB reserve shift minutes.
Utilization TB reserve shift	**The average utilization of a TB reserve shift.**

Training this model, results in a function that can predict the utilization based on weekday, function, location, shift-type, sum of regular planned shift minutes, and sum of TB reserve minutes.

Followingly, we translate the scheduling problem into a mathematical formulation, using variables and constraints to find optimal solutions.

- Index Variables:

  Function, Location, Weekday, Shift-type.

- Decision Variable:

  Quantity of personnel in the TB reserve schedule.

- Input Variable:

  Sum of scheduled regular minutes, target utilization, length of a shift.

- Objective Function:

  Minimize the difference between predicted utilization and target utilization for TB reserve shifts.

The size solution space for this model depends on the levels in the start solution and the maximum number of shifts per level. The number of levels is determined by the possible combinations of weekday, shift, function, and location. Meanwhile, the maximum number of shifts states the maximum availability of personnel in the certain level. These two factors determine, for instance, how many conductors can be scheduled for a late shift on a Thursday in Amsterdam.

For example, if 1094 unique rows (levels) represent the total combinations of parameters and each row has 5 shifts available, this results 5^{1094} possible solutions. However, this results in an exceedingly large number of potential solutions, making it impractical to evaluate all of them within a reasonable timeframe. Consequently, it is computationally unfeasible to solve all possible solution within acceptable time making alternative method necessary to approach optimality.

We selected simulated annealing as the most appropriate metaheuristic for our problem. The algorithm starts with an initial solution and temperature (T_{start}). In each iteration, a random change creates a new neighbouring solution. If this solution is better, it is accepted. If it is worse, it can still be accepted based on an acceptance probability, which depends on the current temperature (T) and the energy difference (ΔE) between solutions, calculated as $e^{\frac{\Delta E}{T}}$. The temperature gradually decreases according to a cooling factor (α), balancing global and local search. The process continues until the temperature reaches a low fixed point $\left(T_{stop}\right)$.

The initial neighbourhood operator is an increment-decrement operator that randomly adds or subtracts one shift from the current solution. However, to prevent the model from optimizing towards scheduling no shifts (yielding a trivial 'optimal' utilization error of zero), the operator is modified to disallow adjustment of shifts to zero. Despite preventing trivial optimizations, this adjustment made the model highly dependent on the starting solution, limiting the exploration of alternative neighbouring solutions. While other operators, like a swap operator, could potentially address these limitations, time constraints restricted the ability to validate and implement them.

Data is gathered from Microsoft Azure Data Lake and processed using Snowflake SQL. The training data includes planning data and execution data since December 2021.

Apache Spark (PySpark) is used for computational efficiency, enabling large-scale data processing. The technical specifications include:

1-3 workers (16-48 GB Memory, 4-12 Cores)

1 Driver (16 GB Memory, 4 Cores)

Runtime (14.3x-scala2.12)

5 Validation and experiment design

5.1 Validation

While running experiments for validation, excessive runtimes where experienced due to use of a machine learning prediction model in an iterative algorithm. 5000 iterations take around 3-20 hours, depending on available memory and Sparkperformance.

For the validation of the XGBoost machine learning model we chose to evaluate three performance measures.

- The Root Mean Squared Error (RMSE)
- R-Squared (R^2)
- Mean Absolute Error (MAE)

We use two methods to train and validate the Machine Learning model:

- **Random split**

  This is done by randomly putting 30% of the data in the testing dataset and 70% in the training dataset.

- **Fixed split**

  Dividing the training and testing data based on timetable year. In our case on timetable year 2022 and 2023.

Random split has 2208 training instances and 1015 test instances. Fixed split has 1054 training instances and 1075 testing instances. Table 2 shows the performance measures of validation on both the testing dataset and the training dataset.

Table 2 Validation on test dataset and training dataset

	Testing dataset		Training dataset	
Performance measure	**Random split**	**Fixed split**	**Random split**	**Fixed split**
RMSE	0,0413	0,0394	0,0312	0,0183
R^2	0,9627	0,9673	0,9793	0,9938
MAE	0,0232	0,0233	0,0183	0,0108

Both methods indicated potential overfitting, with the random split showing less difference between training and testing data. Given the overall strong predictive capability (minimum R-squared value of 0.96) and less sign of overfitting compared to fixed split, the random split was chosen for its ability to incorporate more recent, uncontaminated (Covid) data.

5.2 Simulated annealing parameter selection

The initial temperature should result in a quasi-equilibrium state where all solutions are equally acceptable (Atiqullah, 2004). We do this by visual inspection of the solution. Because of the excessive running times, this was time intensive. Figure 3 depicts the result of the experiment with the most graphs hovering near the quasi-equilibrium. We see that an initial temperature of 0,075 results in a solution graph that moves around a near horizontal line. Therefore, we chose 0,075 as our initial temperature.

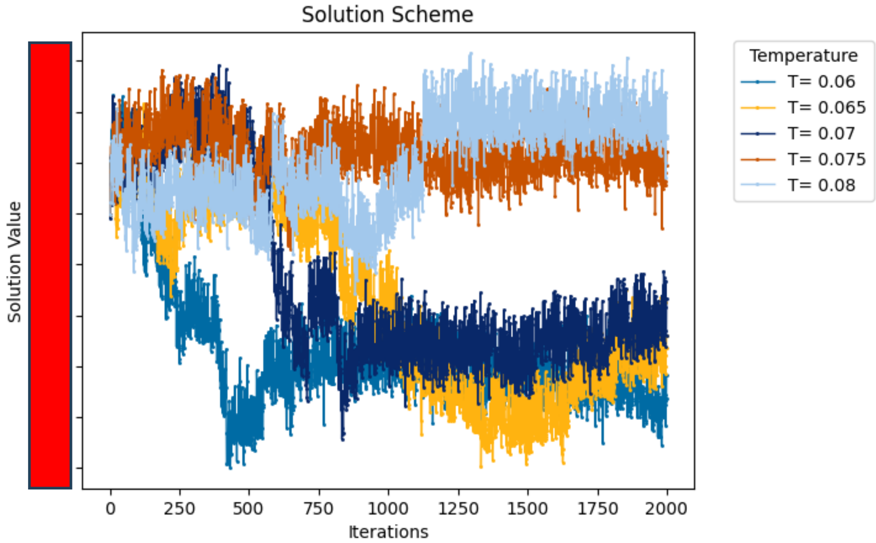

Figure 3 Starting temperature determination

Already in these graphs we can see strange behaviour of the model. The steep descents and ascents in the graphs of 0,06, 0,07, and 0,08 show us that the solution space maybe behaves differently than expected.

For the cooling schedule we need to determine a cooling strategy, cooling rate (α) and stopping temperature (T_{stop}). Again, experiments with visual inspection are needed to determine the right strategy and parameters. By trial and error, we discovered different cooling rates with a constant stopping temperature of $T_{stop} = 0{,}01$. We first tried the most promising cooling approach, only cooling the temperature down when we find an improvement. This results in the cooling rates depicted in Figure 4. The first 4 experiments were all done within 1 hour each. The experiment with $\alpha = 0{,}94$ was cancelled after running for 12 hours.

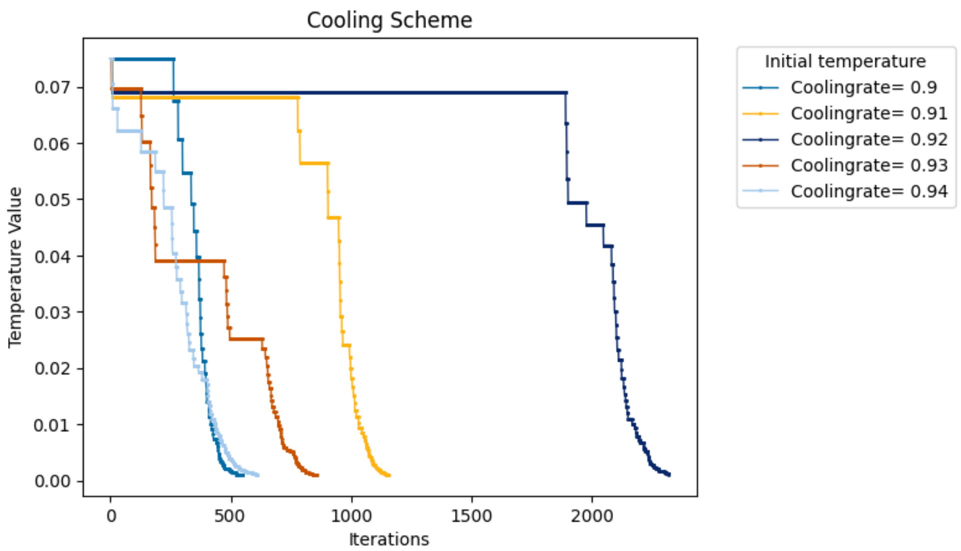

Figure 4 Cooling Scheme (case 1)

One can see that the algorithm gets stuck quite often and for a long number of iterations. This is an indication of a fragmented solution space. Because of this behaviour, we are not able to continue with this cooling strategy since the algorithm sometimes gets stuck and takes over 12 hours to run with higher cooling rates and lower cooling rates converge too fast. This experiment has been repeated, but with a temperature of 0,94 the algorithm again gets stuck.

Therefore, the cooling strategy is adapted to cooling down every iteration. The stopping temperature is decreased accordingly to $T_{stop} = 0{,}0001$. Figure 5 immediately shows a good graph one wants to see. The temperature gradually decreases and allows for local search at the end. However, if one examines Figure 6, it quickly reaches the best solution value found and then gets stuck. Most of the running time was not useful.

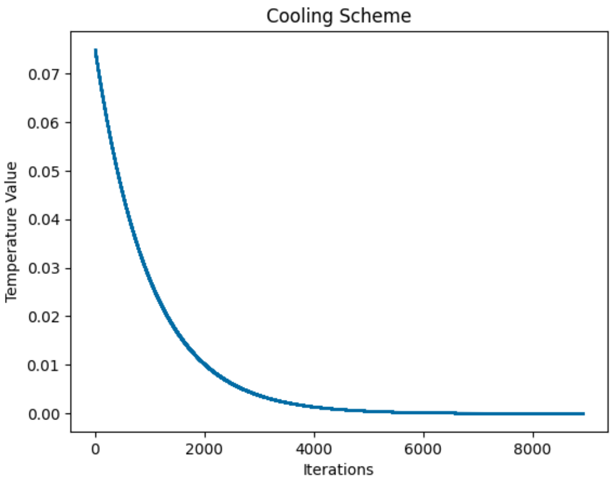

Figure 5 Cooling scheme (case 2)

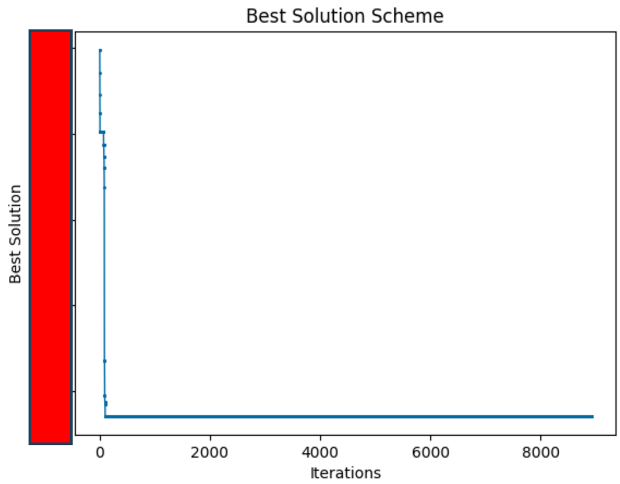

Figure 6 Best solution graph (case 2)

To dive deeper into this matter, the behaviour of the metamodel with different cooling strategies is compared to an iterative algorithm with only accepting improving solutions. This also strengthens the suspicion of a fragmented solution space. Figure 7 depicts the graph of the current solution value during the algorithm. As one can see, after around 1700 iterations, the current solution value jumps to another neighbourhood and does not decrease any further. After 13 hours this experiment is cancelled.

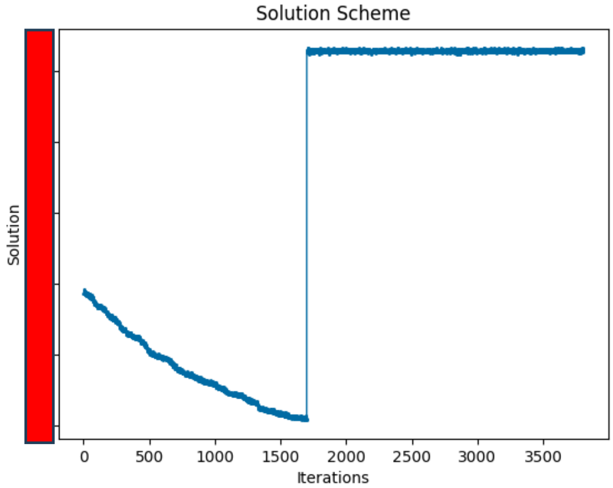

Figure 7 Current solution graph (case 3)

5.3 Experiment design

Due to unexpected behaviour observed during parameter selection, four experiments were designed to further explore and improve the model performance:

- Hill climbing

  The hill climbing algorithm is a local search method to explore improvements within neighbouring solutions. It iteratively searches for improvements by selecting a random neighbouring solution. If no improvement can be found, the algorithm stops.

- Iterative improvement algorithm

  The iterative improvement algorithm works like the hill climbing algorithm, but it does not stop when a worse solution is found. The algorithm runs for a predetermined number of iterations, only storing improving solutions.

- Simulated annealing

  The simulated annealing algorithm works as explained in the previous chapter.

- Adapted simulated annealing

  Because we experienced unexpected behaviour of the model when a worse solution is accepted, we adapt the model in a way that it enables getting out of worse solutions. If a worse solution is accepted and the algorithm finds no improving solution within 100 iterations, the best solution is restored.

6 Experiments

Table 3 presents the results of the 3 best performing experiments we have conducted. The hill climbing stopped after 2 iterations and was not considered for further analysis. The standard simulated annealing approach did not reach promising solutions. The iterative algorithm and adapted simulated annealing however did reach more promising solutions. Both improved the starting solution with around 10%, but this is not the significant improvements we have hoped for. We furthermore discovered interesting locations, and patterns in shift-type and weekday.

The adapted simulated annealing reached the best solution, which was only 1,16% better than the iterative algorithm. Notably, the iterative algorithm needed over 6000 more iterations to reach this point. This suggests that our implementation of adapted simulated annealing balanced global and local search in a more efficient way. However, when running the iterative algorithm again with the exact number of iterations the adapted simulated annealing needed to reach its best solution, the iterative algorithm reached a solution better than the adapted simulated annealing. This again shows the strange behaviour of our solution space.

To further investigate the results, we compared the iterative algorithm and adapted simulated annealing on location, shift-type, and weekday level. For location and shift-type significant overlapping patterns emerged. On weekday level we see a minimal pattern of increased added shifts around Thursday, Friday, and Saturday. Despite this observation, the pattern is too minimal to draw strict conclusions from that.

The main conclusion of the experiments is that the randomness of the neighbourhood operator together with the limited number of iterations due to long computational times, led to insufficient results to draw strict conclusions. The fragmented nature of the solution space and the unexpected behaviour of the model are key issues that will be discussed further in the next chapter.

Table 3 Results of experiments

	Iterative algorithm	Simulated annealing	Adapted simulated annealing
Number of iterations	5000	8920	2228
Number of improvements	208	10	202
Running time	05:47:39	03:24:43	04:38:18
Difference error sum	-10,91%	-0,75%	-11,94%
Conductor shifts | FTE	89 | 24,7	8 | 2,22	40 | 11,11
Driver minutes | FTE	83 | 23,1	21 | 5,83	59 | 16,39

7 Conclusion

Despite the research's goal to minimize TB reserve utilization error by optimizing the TB reserve crew schedule, significant improvements were not achieved. The key reasons are summarized below along with points of discussion and limitations.

- Performance Measure:

  The utilization metric used to measure TB reserve performance showed inadequacies. Numerous assumptions and adaptations were necessary to arrive at a functional metamodel. Utilization may not be a sufficient metric for forecasting new schedules and should be reconsidered.

- Fragmented Solution Space:

  The experiments indicated a possible fragmented solution space, evidenced by the metamodel frequently jumping to local maxima and the hill climbing algorithm failing beyond two iterations. These observations suggest underlying complexities, although complete validation was hindered by computational constraints.

- Computational Difficulties:

  Combining machine learning, mathematical optimization, and metaheuristics resulted in high computational demands, leading to long runtimes and limited iterations. Consequently, the results may have been influenced by randomness, complicating conclusive findings.

- Fit/Overfit:

  The machine learning model exhibited signs of overfitting, necessitating further model validation and robustness checks.

Within the research we found points to further look into by NS or another researcher. Unfortunately, due to confidentiality reasons we cannot share them here.

Below we summarize the contribution to theory and practice:

This research showcases the potential of integrating machine learning with mathematical optimization to form metamodels. While the chosen performance measure posed limitations, the metamodel concept for forecasting and optimization combined holds promise. Future research should focus on validating such models on smaller instances with fewer assumptions to improve efficacy and computational efficiency.

Discussions with NS stakeholders revealed that the problem remains too complex for direct optimization. Essential insights into TB reserve crew performance, the importance of target values, and the implications of such optimization processes were gleaned. NS should further clarify their performance objectives and the intended outcomes for optimized scheduling.

8 Bibliography

A Gentle Introduction to Apache Spark on Databricks. (2024). Opgehaald van DataBricks: https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/346304/2168141618055043/484361/latest.html

Aarts, E., & Korst, J. (1990). Simulated annealing and Boltzman machines - a stochastic approach to combinatorial optimziation and neuarl computing. *Wiley-Interscience seris in discrete mathematics and optimization*.

Abbink, E., Albino, L., Dollevoet, T., Huisman, D., Roussado, J., & Saldanha, R. (2011). Solving large scale crew scheduling problems in practice. *Public Transport*.

Atiqullah, M. M. (2004). An Efficient Simple Cooling Schedule For Simulated Annealing. *Computational Science and Its Applications*, pp. 390-404.

Baştanlar, Y., & Özuysal, M. (2014). *Introduction to Machine Learning.*

Bayliss, C., De Maere, G., & Paelinck, M. (2020). Scheduling Airline Reserve Crew using a Probabilistic Crew Absence and Recovery Model. *Journal of the Operational Research Society*.

Bayliss, C., De Maere, G., Atkin, J., & Paelinck, M. (1998). Probabilistic Airline Reserve Crew Scheduling Model. *ACM Subject Classification G.1.6 Optimization, G.3 Probability and Statistics*.

Bayliss, C., De Maere, G., Atkin, J., & Paelinck, M. (2014). A Simulation Scenario Based Mixed Integer Programming Approach to Airline Reserve Crew Scheduling Under Uncertainty. *10th International Conference of the Practice and Theory of Automated Timetabling*.

Bengtsson, L., Galia, R., Gustafsson, T., Hjorring, C., & Kohl, N. (2007). Railway Crew Pairing Optimization. *LCNS*.

Besluit hoofdrailnet. (2015). Opgehaald van Wetten Overheid: https://wetten.overheid.nl/BWBR0017795/2015-01-01/0

Breugem, T., Van Rossum, B., Dollevoet, T., & Huisman, D. (2022). A column generation approach for the integrated crew re-planning problem. *Omega*.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., & Wagenaar, J. (2014). An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, pp. 15-37.

Caprara, A., Kroon, L., Monaci, M., Peeters, M., & Toth, P. (2007). Chapter 3 Passenger Railway Optimization. *Handbooks in Operations Research and Management Science*, pp. 129-187.

Cerny, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*.

Chen, T., & He, T. (2024). *xgboost: eXtreme Gradient Boosting.* Opgehaald van https://cran.ms.unimelb.edu.au/web/packages/xgboost/vignettes/xgboost.pdf.

De Jong, K. A. (1975). *Analysis of the behavior of a class of a genetic adaptive system.*

Derigs, U., Malcherek, D., & Schäfer, S. (2010). Supporting strategic crew management at passenger railways—model, method and system. *Public Transport*.

Elizondo, R., Parada, V., Pradenas, L., & Artigues, C. (2010). An evolutionary and constructive approach to a crew scheduling problem in underground passenger transport. *Journal of Heuristics*.

Ernst, A., Jiang, H., Krishnamoorthy, M., Nott, H., & Sier, D. (2001). Rail Crew Scheduling and Rostering: Optimisation Algorithms. *Crew planning optimisation for freight trains.*

Feo, T. A., & Resende, M. G. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, pp. 109-133.

Feo, T. A., & Resende, M. G. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*.

Froger, A., Guyon, O., & Pinson, E. (2015). A set packing approach for scheduling passenger train drivers: the French experience. *RailTokyo2015*.

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of Metaheuristics.*

Glover, F. (1989). Tabu Search - Part I. *OSRA Journal on Computing*.

Glover, F. (1990). Tabu Search - Part II. *OSRA Journal on Computing*.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison-Wesley*.

Heil, J., Hoffman, K., & Buscher, U. (2020). Railway crew scheduling: Models, methods and applications. *European Journal of Operational Research*.

Hertz, A., & Widmer, M. (2003). Guidelines for the use of meta-heuristics in combinatorial optimization. *Eueropean Journal of Operational Research* .

Homaie-Shandizi, A.-H., Nia, V. P., Gamache, M., & Agard, B. (2016). Flight deck crew reserve: From data to forecasting. *Engineering Applications of Artificial Intelligence*.

ISO. (2017). *Online Browsing Platform*. Opgehaald van https://www.iso.org/obp/ui/#iso:std:iso-iec:38505:-1:ed-1:v1:en

Jeusfeld, M. A. (2009). Metamodel. *Encyclopedia of Database Systems*.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*.

Kuznetsov, N., Minashina, I., Ryabykh, N., Zakharova, E., & Pashchenko, F. (2016). Design and comparison of freight scheduling algorithms for intelligent control systems. *Procedia Computer Science*, pp. 56-63.

Laarhoven, P. J., & Aarts, E. H. (1987). *Simulated Annealing: Theory and Applications.*

Lawson, C., Martí, J., Radivojeci, T., Jonnalagadda, S., Gentz, R., Hilson, N., . . . Martin, H. (2021). Machine learning for metabolic engineering: A review. *Metabolic Engineering*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*.

Liepins, G., & Hilliard, M. (1989). Genetic Algorithms: Foundations and Applications. *Annals of Operations Research*, pp. 31-58.

Lin, D.-Y., & Tsai, M.-R. (2019). Integrated Crew Scheduling and Roster Problem for Trainmasters of Passenger Railway Transportation. *IEEE Access*.

Lu, K., Han, B., & Zhou, X. (2018). Smart Urban Transit Systems: From Integrated Framework to Interdisciplinary Perspective. *Urban Rail Transit*.

Mansveld, W. J. (2014). *Concessie voor het hoofdrailnet 2015-2025.* Ministerie van Infrastructuur en Milieu.

Meng, L., Corman, F., Zhou, X., & Tang, T. (2018). Special issue on Integrated optimization models and algorithms in rail planning and control. *Transportation Research Part C: Emerging Technologies*.

Mitchell, M. (1998). An Introduction to Genetic Algorithms. *MIT Press*.

Naeem, M., Rizvi, S. T., & Coronato, A. (2020). A Gentle Introduction to Reinforcement Learning and Its Application in Different Fields. *Institute of Electrical and Electronic Engineers*.

NS. (2022). *NS CAO 2022-2023*. Opgehaald van Werken bij NS: https://www.werkenbijns.nl/static/uploads/37219-ca-o-n-s-2022-2023.pdf

NS. (2022). *NS Jaarverslag*. Opgehaald van https://www.nsjaarverslag.nl/

NS. (2023). *NS Geschiedenis*. Opgehaald van NS Geschiedenis: https://www.ns.nl/over-ns/wie-zijn-wij/geschiedenis/

Osman, I., & Kelly, J. (1996). Meta-Heuristics: An Overview. *Osman*.

Pandala, S. R. (2022). *Lazy Predict*. Opgehaald van https://lazypredict.readthedocs.io/en/latest/

Peng, K., & Shen, Y. (2016). An evolutionary algorithm based on grey relational analysis. *The Jornal of Grey System*.

Perrault-Lafleur, C., Carvalho, M., & Desaulniers, G. (2023). A stochastic integer programming approach to reserve staff scheduling with preferences. *International Transactions In Operations Research*.

Reeves, C. R. (1995). *Modern Heuristic Techniques for Combinatinatorial Problems.* McGraw-Hill.

Resende, M. G., & Ribeiro, C. (2003). Greedy Randomized Adaptive Search Procedures. *Handbook of Metaheuristics*.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*.

Schrotenboer, A. H., Wenneker, R., Usavas, E., & Zhu, S. X. (2023). Reliable reserve-crew scheduling for airlines. *Transportation Research Part E*.

Sidey-Gibbons, J. A., & Sidey-Gibbons, C. J. (2019). Machine learning in medicine: a practical introduction. *BMC Medical Research Methodology*.

Stern, H. S. (1996). Neural Networks in Applied Statistics. *Technometrics*.

Sudret, B. (2012). Meta-models for structural reliability and uncertainty quantification. *arXiv: Methodology*.

Tosh, C., Krishnamurthy, A., & Hsu, D. J. (2020). Constrative learning, multi-view redundancy, and linear models. *ArXiv*.

Zhu, X., & Goldberg, A. (2009). *Introduction to Semi-Supervised Learning.*

Colophon

Title	Minimizing utilization error of reserve crew by optimizing scheduling
Subtitle	
University	University of Twente
Company	NS
Auteur(s)	L.C. Nijs
Supervisors UT	dr. S.M. Meisel and dr. I. Seyran Topan
Supervisor NS	M. Jacobs
Date	08 July 2024
Version	3
Status	Summary of confidential thesis
Document	G:\NSR-NSR Transportbesturing\TB-NSR-Kenniscentrum\07 Analisten\Personeel\Afstudeerstage Luuk\Oude versies verslag\Samenvatting.docx