INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research

International Journal of Manufacturing Research

ISSN online: 1750-0605 - ISSN print: 1750-0591
https://www.inderscience.com/ijmr

Reproducible decision support for industrial decision making using a knowledge extraction platform on multi-objective optimisation data

Simon Lidberg, Amos H.C. Ng

DOI: 10.1504/IJMR.2024.10057049

Article History:

Received:	13 November 2022
Last revised:	28 March 2023
Accepted:	29 March 2023
Published online:	20 December 2023

Reproducible decision support for industrial decision making using a knowledge extraction platform on multi-objective optimisation data

Simon Lidberg*

School of Engineering Science,
University of Skövde,
Högskolevägen, Box 408,
541-28, Skövde, Sweden
Email: simon.lidberg@his.se
and
Manufacturing Engineering Development,
Volvo Group Trucks Operations,
John G. Grönvalls Plats 10, 541-37,
Skövde, Sweden
*Corresponding author

Amos H.C. Ng

School of Engineering Science,
University of Skövde,
Högskolevägen, Box 408,
541-28, Skövde, Sweden

Abstract: Simulation-based optimisation enables companies to take decisions based on data, and allows prescriptive analysis of current and future production scenarios, creating a competitive edge. However, effectively visualising and extracting knowledge from the vast amounts of data generated by many-objective optimisation algorithms can be challenging. We present an open-source, web-based application in the R language to extract knowledge from data generated from simulation-based optimisation. For the tool to be useful for real-world industrial decision-making support, several decision makers gave their requirements for such a tool. This information was used to augment the tool to provide the desired features for decision support in the industry. The open-source tool is then used to extract knowledge from two industrial use cases. Furthermore, we discuss future work, including planned additions to the open-source tool and the exploration of automatic model generation.

[Submitted 13 November 2022; Accepted 29 March 2023]

Keywords: knowledge-extraction; reproducible science; simulation-based optimisation; industrial use-case; decision-support; knowledge-driven optimisation.

Reference to this paper should be made as follows: Lidberg, S. and Ng, A.H.C. (2023) 'Reproducible decision support for industrial decision making using a knowledge extraction platform on multi-objective optimisation data', *Int. J. Manufacturing Research*, Vol. 18, No. 4, pp.454–480.

Biographical notes: Simon Lidberg is an Industrial PhD student at the Volvo Group Trucks Operations and the University of Skövde. In 2018, he started his industrial PhD studies focusing on the creation of decision-support by optimising on the factory level using aggregated data and models derived from detailed DES models.

Amos H.C. Ng is a Professor of Automation Engineering at the University of Skövde, Sweden. He holds a PhD in Computing Sciences and Engineering. His main research interest lies in applying multi-objective optimisation and data mining techniques for production systems design, analysis, and improvement.

This paper is a revised and expanded version of a paper entitled 'A knowledge extraction platform for reproducible decision-support from multi-objective optimization data' presented at 10th Swedish Production Symposium, Skövde, Sweden, 26–29 April 2022.

1 Introduction

Research and industrial interests in the concept of Industry 4.0 are steadily increasing (Liao et al., 2017). For industry, adopting the Industry 4.0 paradigm early means getting a competitive edge over other businesses. For academia, there exist funding opportunities in the research area due to national and international interests. Industry 4.0 embraces cyber-physical systems, and more importantly, cyber-physical production systems, resulting in an increase in the use of simulation (Monostori et al., 2016).

Data generated in an Industry 4.0 context can be used for descriptive, diagnostic, predictive, or prescriptive analytics (Mahmoodi et al., 2022). For a production line, descriptive analysis can provide answers about the current state of the process, while diagnostic analytics can identify what went wrong in the process. If predictive analytics are attained, predictions about when the process will fail can be used. The most potent form of analysis is prescriptive analytics, whereas a recipe for improving the process can be attained. Analysis of the large datasets from simulation-based multi-objective optimisation (SBO) have been described as simulation-based innovisation (SBI) or knowledge-driven optimisation (KDO) (Ng et al., 2009; Bandaru and Deb, 2016). The knowledge generated from the optimisation data increases the utility of the optimisation. In the manufacturing context, knowledge can be used to prescribe improvement efforts with which the biggest gain for the least amount of cost can be found, and in which order (Jain et al., 2015).

As manufacturing industries in general, and the automotive industry in particular, are moving towards the electrification of the transport sector, a large transformation is required. This affects the automotive industry wherein new production is required to tackle new products introduced, while in parallel dealing with unstable supply chains. To be proactive about this change, alternatives need to be analysed and the consequences on existing production and supply chain will need to be investigated. As argued in this paper, this kind of analysis can be effectively supported by SBO.

Currently, one of the biggest problems in academic research is reproducibility. One example of the lack of reproducible science was examined in the field of Psychology. Decades' worth of academic output in the form of peer-reviewed literature reported results were found not to be reproducible (Aarts et al., 2015). To counter this development, academic journals embrace open science and open data, encouraging researchers to publish supplementary material, e.g., raw data, processed data, and code to reproduce and corroborate findings. This development aims to increase the reproducibility and transparency of methods and results.

When the researcher relies on graphical user interfaces or particular visualisation software to perform their analysis, the steps taken are difficult to document and reproduce exactly. Instead, each step should be expressed and be reproducible in code. This is important both for publishing, and for the workflow of each researcher. The use and creation of an open-source knowledge-extraction tool allowing reproducible analysis are therefore important, both from the perspective of industry and academia. For industry, it would allow for faster decision support delivered to strengthen the competitiveness of industry, and record steps are taken in visual analytics to be reproduced later. There exist tools for the connected field multiple criteria decision making, but they are not focused on reproducibility (Hakanen et al., 2022; Shavazipour et al., 2021; Mazumdar et al., 2022).

The aim of this study is to investigate the requirements of industrial decision makers regarding decision support provided by KDO. The requirements are gathered from industrial decision makers on their needs for decision support. Management teams were shown a relevant simulation model and corresponding dataset obtained by SBO through a decision-support system. Results from the requirement analysis were used to guide further developments of the decision-support tool and gain insights into the decision-making process on the higher manufacturing hierarchy levels.

This article is structured as follows: Section 2 details the foundations of SBO, multi-objective optimisation (MOO), KDO, and discusses other research in the area. Section 3 describes the method of gathering the requirements and the optimisation case studies and their results are presented in Section 4. The study is concluded with conclusions and discussions on future work in Section 5.

2 Background

In the background section, MOO datasets and their special properties will be presented in addition to knowledge-extraction methods of MOO datasets. The main knowledge-extraction method is flexible pattern mining (FPM), based on frequent itemset mining and association rule learning, presented in Subsection 2.2. Another method for explicit rule generation is decision trees, detailed in Subsection 2.1.

Creating knowledge by applying data mining methods to MOO data differ from other data mining applications due to the properties of the MOO dataset (Bandaru et al., 2017a). These special properties, and the requirements for the data mining methods applied to generate knowledge from the datasets, can be summarised as follows:

- handling of two separate spaces, objective and decision space

- decision maker involvement
- knowledge representation
- different variable types in the data
- problem parameters.

Visualisations of MOO datasets are performed in the objective space, showing the results from the optimisation, but the extraction of knowledge is handled in the decision space with all the inputs. These two spaces are different, and clusters of objectives identified through clustering methods are most likely not reflected as the same clusters in the decision space. Thought needs to be placed into the application of knowledge-extraction and clustering methods for MOO datasets.

The involvement of a decision maker in a practical MOO problem to select the wanted solution – and possibly also influence the optimisation process by introducing preferences – is an aspect special to MOO. The preferred solutions selected by a decision maker should be used to extract knowledge from their respective decision space, but knowledge can also be gained by using data mining on the non-selected solutions to differentiate decision variables between wanted and non-wanted solutions.

Representing the knowledge mined from the preferred solutions should generate knowledge in an explicit form due to the need to present it to a decision maker and later convert it to an actionable form. This explicit knowledge can be in the form of analytical relationships, decision rules, or association rules. A MOO dataset can include different types of data, both continuous, discrete, and nominal. Data mining methods would need to handle a mixture of these (Bandaru et al., 2017a).

MOO problems can also include problem parameters in addition to variables and objectives. These problem parameters are not altered by the optimisation algorithm but represent external controlling parameters such as constraints, variable bounds, or even objective functions. By altering these parameters and then running the optimisation again, higher-level knowledge can be obtained (Bandaru and Deb, 2013).

2.1 Decision trees

Decision trees have become widely used as classifiers in machine learning, pattern recognition, and most important for this study, data mining (Rokach and Maimon, 2010). This study will focus on classification and regression trees (CARTs) due to their ability to generate both CARTs (Breiman et al., 1984).

The data in decision trees is partitioned to maximise the homogeneity of the response variables by identifying the predicting variable which will best split the data into two homogeneous groups. For each partition, each predictor is evaluated, even those used in previous partitions. Maximising homogeneity in each partition is achieved by minimising heterogeneity, most commonly using the Gini Index, i.e., the probability of a randomly chosen element would be mislabeled if it was randomly labeled according to the distribution of labels in the partition. The base of the inverted tree is called the root node, and the end results are called leaves or terminal nodes. After growing the trees, a pruning step is applied with which the tree is shortened by removing leaves, 'trading accuracy for simplicity' (Rokach and Maimon, 2010). Pruning has been shown to improve the generalisation performance of the decision tree (Breiman et al., 1984).

Applications for the improvement of manufacturing can be found in Thomas et al. (2014, 2015), Bergmann et al. (2017) and Prajapat et al. (2020).

Decision trees are explicit in nature, as they are self-explanatory and can be converted to a set of rules, and can handle both nominal and numeric attributes with discrete and continuous outcomes (Rokach and Maimon, 2010). The main problem with decision trees is their tendency to overfit the data, meaning that the model has become too specialised and will not perform well when encountering new data (Berk, 2009). Reducing overfitting can be accomplished by constructing average results over random samples of the data, which gave rise to an early technique known as bootstrap aggregation or 'bagging'. Bagging grows multiple trees, each using a random sample of the training data, and the predictions from all trees are then averaged (c.f. Breiman, 1996). Bagging eventually led to random forests (Breiman, 2001).

2.2 Pattern mining

Sequential pattern mining was developed through the analysis of market basket (sales) data in an effort to determine which items were commonly bought together by a specific customer (Agrawal and Srikant, 1995). Three concepts are important in sequential pattern mining, itemset, sequence, and support. The itemset is a combination of items bought together at a specific instance, while the sequence is an ordered list of all itemsets bought by the customer. If the sequence of a customer includes a specific sub-sequence, the customer supports that sub-sequence. A list of all sub-sequences with a support value larger than a predefined minimum support value is the target of sequential pattern mining (Höppner, 2010).

The apriori algorithm is one method of association rule mining frequently used in data mining applications (Bandaru et al., 2017b). Starting with a list of all one-itemset sequences L_1, which meets the minimum support value, the algorithm continues by adding one-itemsets together into two-itemset sequences, called L_2, and analysing support for those. This combination is possible through the downward closure property, stated as "every non-empty subset of a frequent sequence is also frequent" (Agrawal and Shafer, 1996). Terminating the algorithm is the condition that L_k is empty, i.e., no frequent k-itemsets can be identified. The list of frequent sequences is then pruned to create sequences called sequential patterns (Agrawal and Shafer, 1996).

Sequential pattern mining is a good method for identifying exact patterns in a dataset, but for elicitation of explicit knowledge, the rules generated by decision trees are more useful. FPM has been developed as an extension to sequential pattern mining to combine the strengths of sequential pattern mining with the rule-generation capabilities of decision trees (Bandaru et al., 2017b). Each solution in the MOO dataset is treated as a customer in sequential pattern mining, and each variable as a transaction. Converting the sequences into a table, where each parameter x, where $x \in \{0, 1\}$, is evaluated for less than, equal, or greater than each of its discrete options, allows for the creation of rules by applying the apriori algorithm to find patterns of ones. The stated benefits of FPM are: an unsupervised implementation, the generation of rules, and does not require class labels. If a preferred area of solutions is expressed by the decision maker, the ratio of support for rules in the selected area to that of rules in the unselected area can be used as ordering (Bandaru et al., 2017b). New methods related to using MOO dataset with machine learning algorithms are usually related to online KDO, meaning using the knowledge learned from the dataset to assist the optimisation to converge faster to some preferred region in the objective space (Mittal et al., 2022). In contrast, the current paper is focused on offline KDO wherein the generated knowledge and visualisations are used to assist human decision makers (i.e., human learning).

3 Method

Using SBO has several benefits, but given the large amounts of data it generates which can be difficult to interpret and present to a decision maker (Amouzgar et al., 2018). Without some proper analysis and data-mining techniques, knowledge about the underlying system will be impossible to attain and some benefits of SBI will be lost. Quick and high-quality visualisations can also aid in the exploration of a dataset when performing an initial analysis of the problem or preparing data for publication. Interactive selection of solutions in a graphical user interface enable users to quickly select a region of interest, compared to creating filter rules for the same solutions. Existing solutions for SBO data exploration and knowledge-extraction are powerful, but many are closed-source. Acquiring, modifying, analysing, and extending closed-source software is difficult or impossible, and therefore, an open-source alternative is beneficial.

The aim of this research is the augmentation of an open-source knowledge-extraction tool by gathering requirements from industrial decision makers, and the application of the open-source tool to two industrial problems. The knowledge generated from the SBO data can benefit the industry by revealing connections between inputs not known previously.

The method section is divided into four subsections. In Subsection 3.1, the decision-support system and the open-source tool based on KDO are presented. Next, in Subsection 3.2 we detail the requirements gathered from a group of industrial decision makers. Then followed by two sections detailing the industrial case studies performed on the supply-chain and manufacturing site levels in Subsections 3.3 and 3.4.

3.1 Decision-support tool

In the previous paper, the open-source tool for reproducible analysis of MOO data and knowledge extraction was presented by reproducing the results of two journal articles (Lidberg et al., 2022). One of those articles, Dudas et al. (2014), presents the SBI process of:

1. simulation-based multi-objective optimisation (SBO)
2. data pre-processing
3. pattern detection
4. interpretation and evaluation

To support the process steps of SBI, the open-source tool has several views shown in Figure 1. Data exported from the SBO step can be used in the application and entered into the first stage of analysis, called data pre-processing. The data pre-processing stage is supported through two views, *import data* and *filter data*. After data pre-processing follows pattern detection, also supported by two views: *cluster data* and *visualise data*. Finally, the last stage of SBI, interpretation and evaluation, is handled in the view *rule extraction*, supported by two methods: SBI and FPM. The resulting data from the various stages can be exported from the final view called *export data*. The source code for the open-source tool is publicly available as version 0.0.1.9000 (Lidberg, 2022).

Figure 1 Knowledge-extraction captured in code, allowing for a repeatable process from Lidberg et al. (2022) (see online version for colours)

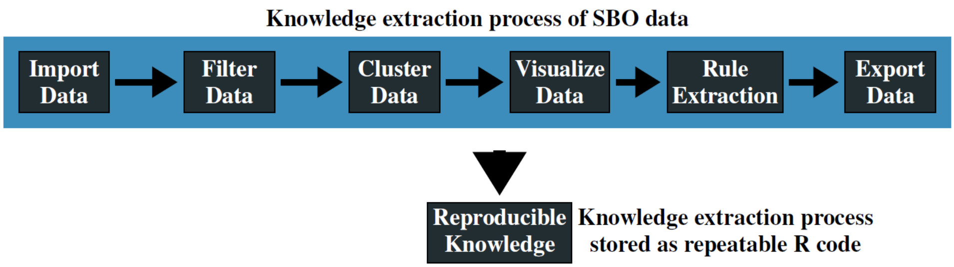
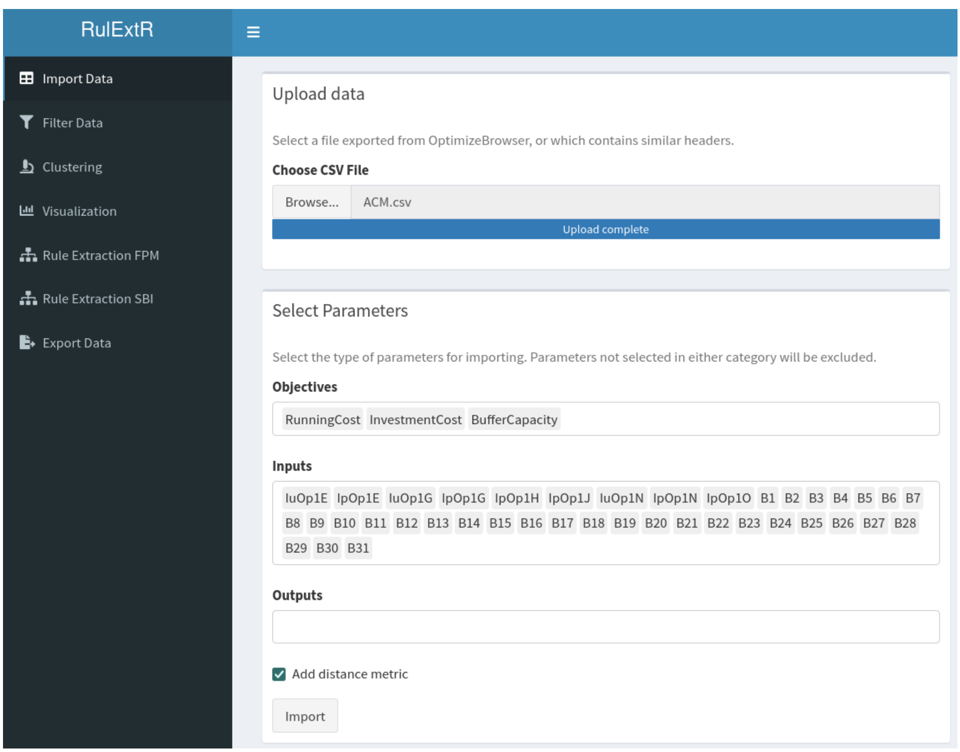


Figure 2 *Import data* view in the application (see online version for colours)



3.1.1 Import data

The intended use of the application is the analysis of optimisation data obtained through SBO. The *import data* view contains two parts, uploading data – formatted as comma separated values (CSV) – and assigning the parameters to one of three groups: objectives, inputs, and outputs as seen in Figure 2. These groupings persist through the remainder of the analysis and influence the methods used. Constraints in the optimisation can be added as outputs or objectives as needed, but are otherwise not recognised as a separate group. The information about the constraints, and whether or not the solutions are feasible, comes from the optimisation backend included in the CSV file.

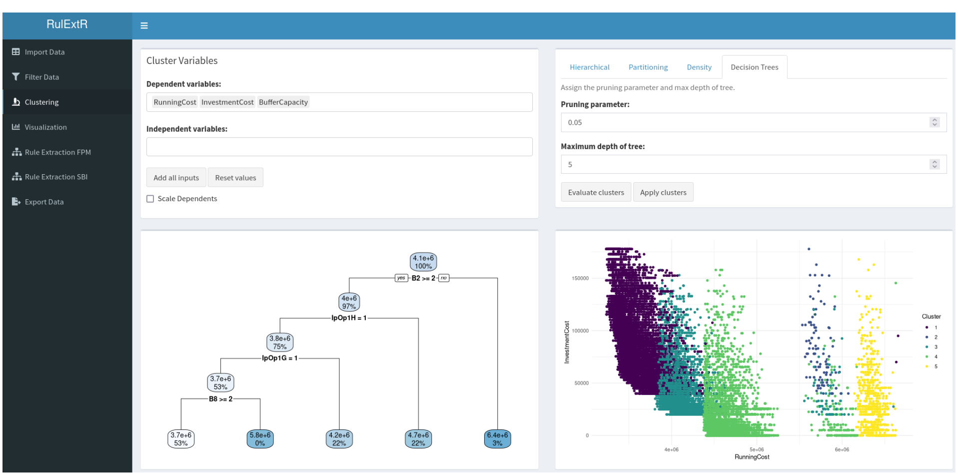
3.1.2 Filter data

The resulting datasets for SBO are usually large, with population numbers and generations combining tens of thousands of solutions. Large amounts of data will complicate the visualisation and hinder understanding for researchers and algorithms. Due to the iterative nature of the genetic algorithms most commonly used for SBO, the solutions early in the optimisation process could be of limited value for the final analysis. The decision maker could also be interested in certain regions of the objective space, or have specific preferences for one objective over another. In these circumstances, filtering the data before clustering, visualising, and extracting knowledge can be beneficial. To support these scenarios, the *filter data* view was implemented in the form of a table view.

3.1.3 Cluster data

Identifying clusters in the data provides knowledge about the connection between the input space and objective space. Four categories of clustering methods organised as tabs are available: hierarchical, partitioning, density, and decision trees, shown in Figure 3. The automatic suggestion for the number of clusters to use is available for some clustering methods, assisting the user in selecting the appropriate number of clusters.

Figure 3 Applying clustering based on decision trees in the cluster data view (see online version for colours)

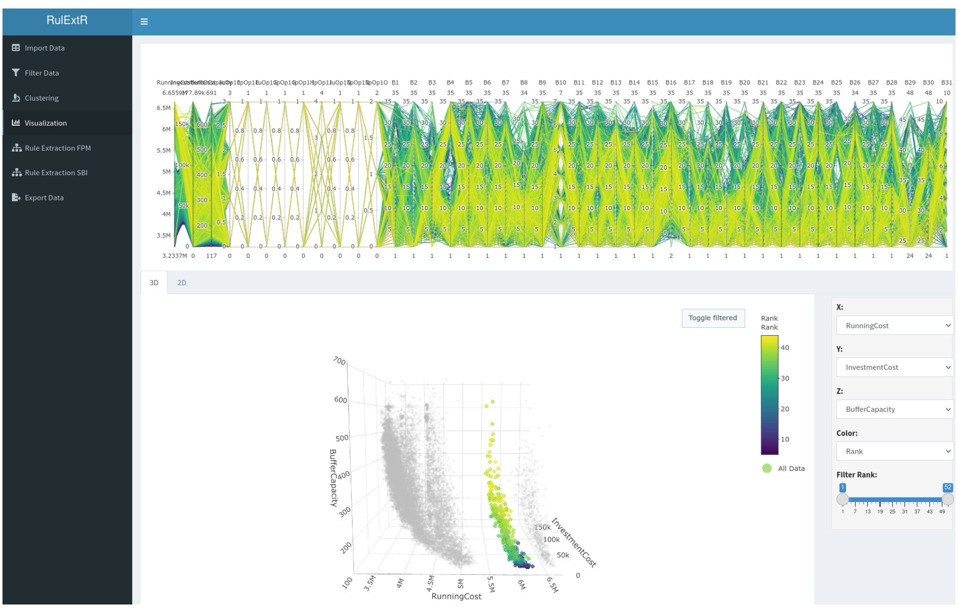


3.1.4 Visualise data

When there are more than three objectives in SBO, visualisation of them becomes difficult. Identifying clusters and structure in data can be achieved using several individual visualisations, although combining them gives even more insight. With brushing and selecting in the visualisation, a decision maker can select the solutions which are relevant and most important to them when taking new decisions based on data.

The *visualise data* view has two components, a parallel coordinates plot on the top, and a 2D/3D-view of the data at the bottom, shown in Figure 4. These components share filtering, colouring, and dimension selection. The parallel coordinates plot allows for filtering and quick visual comparison between the objective space and decision space. The 3D-view offers an alternative view for visual comprehension of the current data, while the 2D-view offers the possibility of interactive selection of interesting areas and/or solutions.

Figure 4 Parallel coordinates chart and visualising an interesting region in the three dimensional scatter plot (see online version for colours)



3.1.5 Rule extraction

Extracting rules from SBO data allows for prescriptive analytics. This shows not only which parameter but also the necessary change to reach a wanted state. The prescriptive rules are presented in order of decreasing importance through the FPM implementation in R, shown in Figure 5. Solutions are either selected in the *visualise data* view or assigned through a reference point. The reference point could be the ideal trade-off point for the decision maker, or a preference in either objective. Rule extraction by decision trees, following the SBI method, is also possible but will not be covered in this study.

Figure 5 Rule generation through FPM in the open-source tool (see online version for colours)

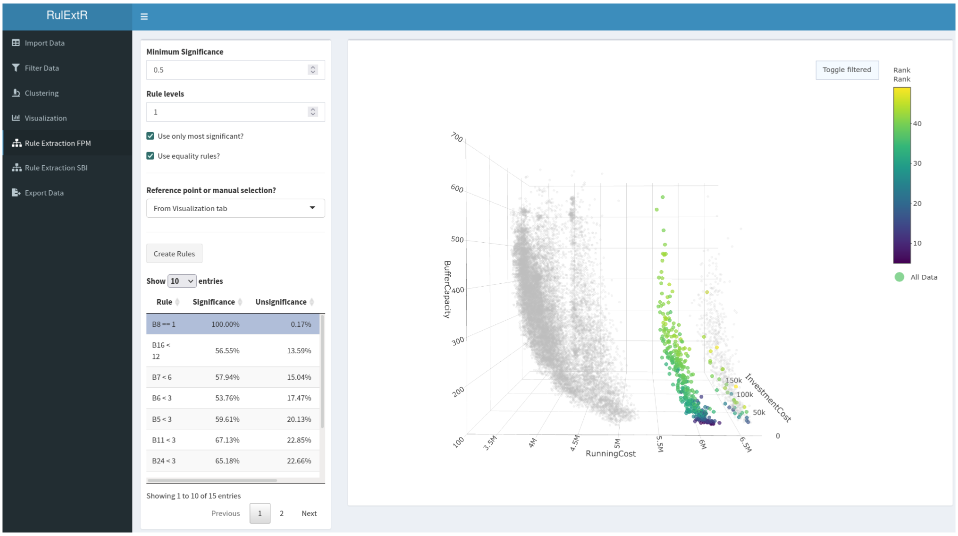
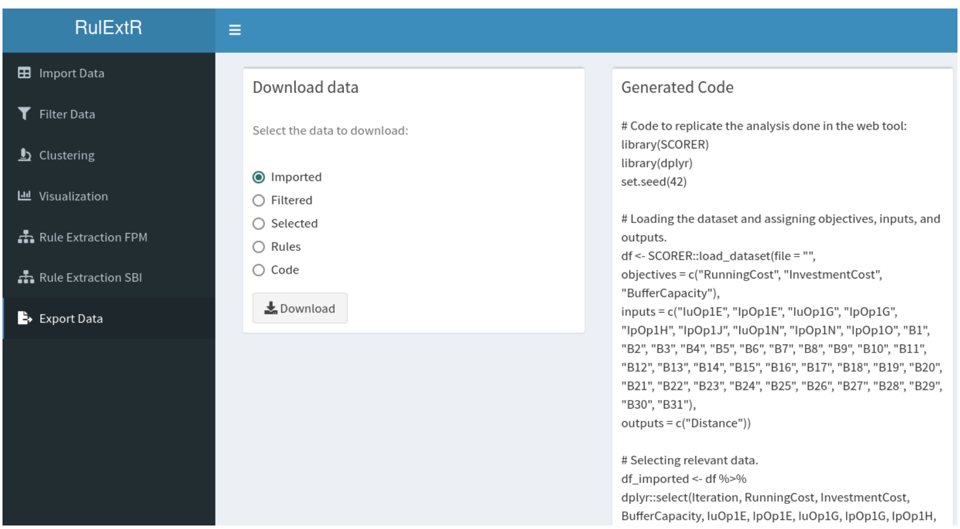


Figure 6 *Export data* view with download options on the left and generated code on the right, cropped for the sake of brevity (see online version for colours)



3.1.6 Export data

At the end of the analysis, four datasets can be exported in the *export data* view as CSV-files, representing the state of data from each of the previous steps. Exporting code allows the user to download an R script file with the generated code, seen in the right half Figure 6. The generated code captures the main steps taken in the application, as changes to the dataset. The code can then be used to reproduce the analysis without the need for an interactive environment. For example, selecting the exact same solutions interactively can be difficult to achieve with consistency. Instead, having the exact solutions stored as code allows for reproducibility of the analysis.

3.2 Industrial requirements analysis

Industrial decision making on the higher levels of the manufacturing hierarchy, i.e., manufacturing site and the supply chain, has a large impact on the entire manufacturing company. Many decisions made also have a large impact on the output of the system and its financial performance. To better understand the decisions made, the range of options and the state of data-based decision making, inputs from actual decision makers are of high relevance.

Industrial decision makers were gathered to get information about the current state of decision-support in addition to their perceived future requirements. The selected participants are members and leaders of the management team for a manufacturing site, including managers for production, maintenance, quality, method development, information technology, production technology, and human resources. A few of the participants had previous experience with simulation and even fewer had experience with KDO.

Smaller groups were formed with three to four participants and each group would discuss open-ended questions and report back on their findings. The context of the discussion was centred around the use cases presented in Subsection 3.3 for the supply-chain case, and in Subsection 3.4 for the manufacturing site case. A deliberate choice was made not to introduce KDO concepts before discussions began in the smaller groups so as to not precondition the participants and limit the range of their discussions. Each group had an external observer with more knowledge about KDO taking notes about the discussions. The discussion outcomes from the smaller groups were then presented to everyone. The questions proposed to the groups were the following:

1 What type of answers are you interested in from a decision-support system?

2 On which key performance indicator (KPI) are you currently basing your decisions?

3 How many alternatives do you consider when making a decision, and how many alternatives are manageable?

4 Do you consider one alternative at a time as a group, or is each participant selecting their own preferred alternative?

The first question was asked to ascertain the type of answers the participants would want from a decision-support system. These answers could range from simple KPI, e.g., lead time, production capacity or costs, to more complex answers, such as identifying bottlenecks, analysing the consequences of introducing new products or taking on new business. The second question is stated to identify the valuable KPI for the decision makers, which could be different depending on the decision making context. For the third question, it was stated to learn how the decision makers handle different alternatives and choose between them. Are the decision makers limiting themselves to just a few alternatives? Is this affected by seniority in the hierarchy and in the chain of decision making? Finally, the last question is posed to investigate the group decision making process. The intent was to investigate the preferences in the number of alternatives, the presentation of alternatives, and the time needed to process the prerequisites of each alternative. Are decision makers willing to perform basic analysis tasks themselves or do they prefer to have the analysis already made for them? This has a large impact if the intent of the decision-support system is to show several alternative trade-off solutions to a decision maker.

3.2.1 Identified requirements

Starting from the first question, the types of answers they would like to get from a decision-support system, they wanted answers to align to already established and set goals. These include KPI for safety, quality, delivery, cost, environment, and people. The scope for the decision-support for this study will be on improving KPI for the delivery goals, which is the easiest to implement in a discrete-event simulation model. The participants suggested that they are not always taking decisions based on data or KPI, but that they wanted to move in that direction.

The participants also requested assistance in answering complex questions about their system. Currently, live data for the manufacturing system is collected and displayed in a dashboard accessible from the web. This allows for a more transparent decision-making process where everyone bases their understanding of the health of the system by a single source of truth. The dashboard has helped decision makers attain descriptive analytics about their system. This view is static and based on historical data. Many decision makers wanted to augment the dashboard with predictive analytic capabilities, with which future performance of the departments and the complete manufacturing site could be displayed.

One of the more complex questions which decision makers wanted assistance regarded scenario handling and what-if analysis. This connects to the third question about the number of manageable alternatives when making a decision. The participants noted that they usually had three alternatives when making a decision; preserving the status quo, the decision maker's wanted state, and a compromise in cost or capability. They also expressed a desire to keep the number of alternative solutions open for as long as possible. In MOO datasets, especially with conflicting objectives, the number of Pareto-optimal solutions can be large. For knowledge-extraction methods, non-Pareto-optimal solutions are also included to get more knowledge from the data, further increasing the number of alternatives.

The biggest problem the participants faced was prioritising the limited resources to increase the utilisation and capability of the system while ensuring that the improvements made are of benefit to the total system output, and not causing sub-optimisations. Identification of the current bottlenecks and the type of problem would enable prescriptive analytics of the system. This, in turn, would create a recipe and the steps needed to reach their wanted state of the complete system.

Trust in the system is an issue brought up by decision makers. Is the data generated from the decision-support system trustworthy, or will people revert to their old and proven ways of analysing their system? This will be out of the scope of this study, but is an important part of the change management process in the digitalisation of industry.

For the fourth and final question, participants stated that much of the decision making was done without a holistic view of the system. Individual departments set their goals and priorities for improvements. As awareness of the state of the system has increased, due to Industry 4.0 initiatives such as the dashboard of live data, decision making has become more transparent. When everyone has aligned their views based on data, the decision making process can be completed, focusing on the output of the complete system, reducing the effect of silos in the organisation.

Concluding this requirements analysis, the decision makers required help with prioritising improvements to increase the current output. Another requirement was to get support in handling a larger number of alternatives. Keeping their options open for as long as possible, and as the time for a decision is coming closer, reduce the number of alternatives to a manageable size and present them in an appropriate context. These requirements will be applied to two optimisation studies described in the next sections.

3.3 Manufacturing footprint case

The first industrial use case is the manufacturing footprint case (MFC). A supply-chain optimisation case where manufacturing of two main products, *product A* and *product B*, can be allocated to three geographically disparate manufacturing sites over the course of six years. The simplified bill of materials can be summarised as; *product A* is produced from raw material arriving from one supplier, while *product B* requires four pieces of *product A* and one piece of a separate raw material arriving from a different supplier. This section will detail the simulation model and the setup of the optimisation with the required customisation to the optimisation algorithm. To protect the intellectual property of the company, specific details will be masked and the model can not be provided.

3.3.1 Simulation model

The discrete-event simulation model is built using Siemens Tecnomatix Plant Simulation, version 2201.003. Each manufacturing site is represented in the model using a similar technique presented in a previous publication (Lidberg et al., 2018). The site is responsible for ordering raw material from one of several possible suppliers depending on available capacity and proximity. Each request for *product A*, *product B*, or raw material, is handled by a global ordering system where the inventory of each product type and site is tracked. Contrary to the previous publication, only one type of transport between different manufacturing sites is utilised in this model, i.e., road transport.

The manufacturing sites and customer sites are permanent, meaning that their location will not change during the optimisation, only where production is placed at the manufacturing sites will differ. The raw material supplier for *product A* will change during the course of the simulation to handle the increase in volume needed from the customers. Additional capacity will open up at a different geographical site than the original source, which will affect the optimisation due to changing transport costs.

3.3.2 Optimisation setup

The optimisation is built in Python, using pymoo version 0.5.0 as the optimisation backend, NSGA-III as the optimisation algorithm, and distributed with the use of the Dask framework (Blank and Deb, 2020). Each manufacturing site, the main interest of this use case, is allowed one to three production lines, $LineA_{1-3}$ for *product A* and $LineB_{1-3}$ for *product B*, where $LineA_N \in \{0, ..., 4\}$ and $LineB_N \in \{0, ..., 8\}$. This means that the input variables are discrete, further complicating the optimisation problem. A demand of these products will need to be fulfilled to three customers over six years (2025–2030). Three possible sites, each with six possible lines, over six years gives in total 108 input variables. These input variables are named using the following format: $Year\_Site\_Line\_N$ with the permitted values shown in Table 1.

Table 1 Optimisation parameter ranges

Parameter	*Value*
Year	2025, ..., 2030
Site	Site1, Site2, Site3
Line	LineA, LineB
N	1, 2, 3

For the optimisation, the goal of the study is to minimise five objectives, each with a corresponding output variable:

1 TransportCost

2 InvestmentCost

3 RunningCost

4 ProductionLoss

5 OverProduction.

The first objective, *TransportCost*, summarises the transportation costs for each year of production, this is a continuous variable and is measured in SEK. *InvestmentCost* summarises the investments needed for setting up new production at the sites, this is also a continuous variable and measured in SEK. Manning costs are summarised in *RunningCost*, a continuous variable measured in SEK. The two last objectives, *ProductionLoss* and *OverProduction*, are constraints used as objectives. *ProductionLoss* quantifies the loss of production at the customer due to insufficient deliveries of *product A* and *product B*, and *OverProduction* quantifies the number of extra products manufactured each year, summarised for each year. By minimising the two last objectives, the solution presented is regarded as resource efficient.

Increasing *RunningCost* for existing production lines or *InvestmentCost* for new production lines will increase the production rate of the specific product. Depending on the location of the new production and where the demand is located, an increase in *TransportCost* can be incurred. On the other hand, if new production lines are opened in closer proximity to the customer, *TransportCost* can be decreased. If the rate of production becomes larger than the demand, this will lead to a negative effect on the *OverProduction* objective.

As previously mentioned, the input variables are discrete, with 0 indicating that the line is not installed nor operational. For $LineA_N$, the setting $LineA_N = \{1, ..., 4\}$ means increasingly longer operational time using shifts shown in Table 2. A higher parameter value means more output, but also more running costs due to higher manning. Investment cost is incurred when moving from $LineA_N = 0$ to another value. A constraint in this model is moving back from $LineA_N = \{1, ..., 4\}$ to $LineA_N = 0$ is not permitted, if the line has been installed it will need to be used.

Table 2 Operating hours for $LineA_N$ and $LineB_N$

Parameter value	*Hours per week*
1 and 5	40
2 and 6	80
3 and 7	120
4 and 8	164

In the second case, $LineB_N$, the parameter values are similar but divided in two capacity steps with $LineB_N = 0$ also representing not installed. The first capacity step is represented by $LineB_N = \{1, ..., 4\}$, with each value representing a different shift pattern and where the working times are identical to $LineA_N$, as shown in Table 2. The second capacity step is represented by the values $LineB_N = \{5, ..., 8\}$, and has doubled the base capacity compared to $LineB_N = \{1, ..., 4\}$. Moving from $LineB_N = 0$ to $LineB_N = \{1, ..., 8\}$ is permitted but not moving from $LineB_N > 4$ to $LineB_N < 4$. If the parameter value is changed from $LineB_N \leq 4$ to $LineB_N > 4$, an investment cost is also incurred, representing the cost of extending the production line. These constraints reduce the feasible space of the optimisation significantly and necessitate the use of customised optimisation operators for crossover, mutation, selection, and repair. The main reason behind customisation of the operators is to reduce the number of infeasible solutions, as each solution needs to be evaluated several times in a computationally expensive simulation model.

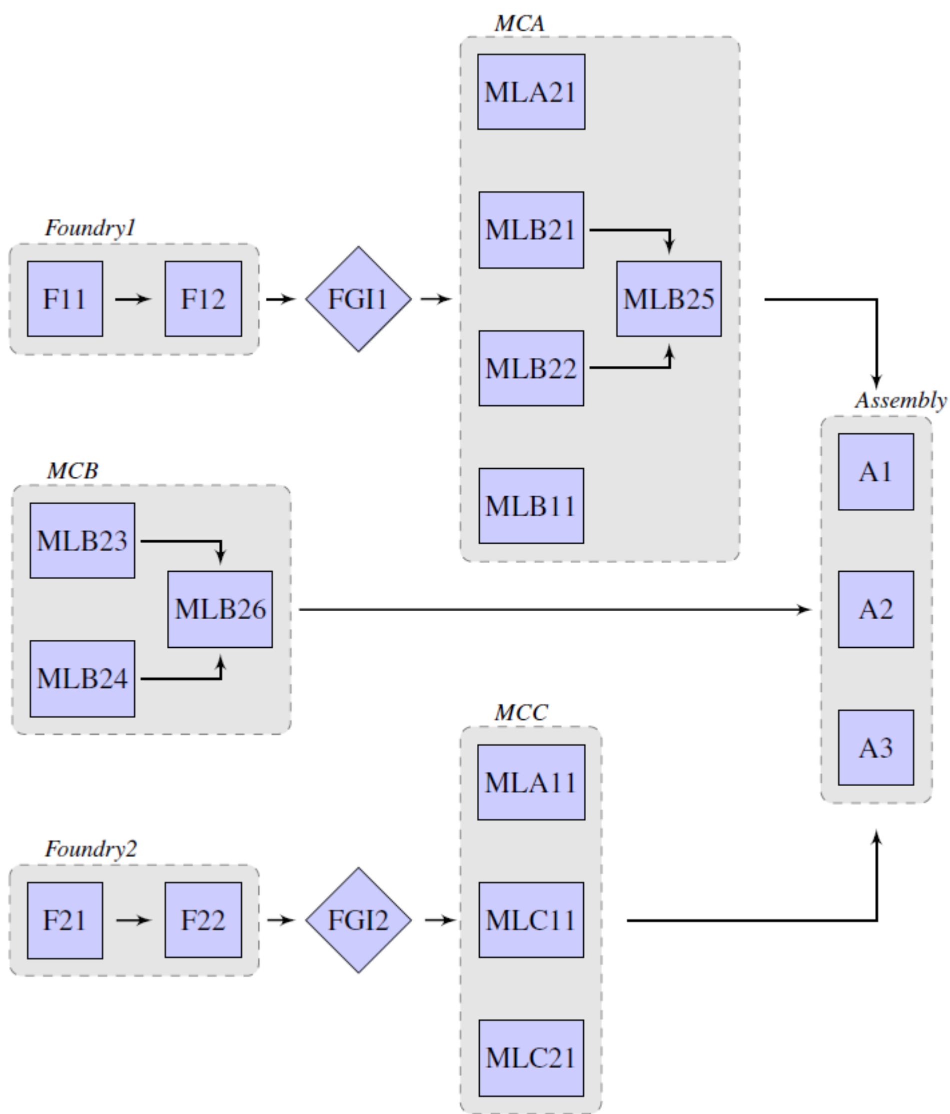
Due to the constraints noted above and the type of parameters, custom crossover and mutation operators had to be created. To keep the feasibility of each production line over all the years, the crossover operator handles input parameters in blocks of six, changing them between parent solutions depending on the crossover probability. Mutation is also affected to preserve the feasibility. If a parameter is selected for mutation, there are two modes depending on if the parameter is zero or larger. If the value is zero, it will randomly assign a number between the limits of the parameter, depending on which production line is selected. If the value is above zero, the mutation operator will increase or decrease by one while respecting the limits of each production line type, i.e., $LineB$ values are not permitted to go from $LineB_N = 4$ to $LineB_N < 4$. As the objective values for *InvestmentCost* and *RunningCost* can be calculated, the main concern is the *TransportCost*, *ProductionLoss*, and *OverProduction* objectives.

3.4 Site bottleneck case

The second industrial use case is the site bottleneck case (SBC). In this section, the simulation model and optimisation will be presented. Sharing of the simulation model is restricted due to the intellectual property of the participating company. The simulation model represents a manufacturing site producing combustion engines and components for heavy duty trucks and machinery. Due to the climate crisis and electrification of the automotive industry, the site has to balance the production of traditional combustion engines with the introduction of new electromobility products. As the complexity increases in the production, SBO and KDO can assist in guiding improvement efforts and transition into prescriptive analytics to better react to an increasingly changing world and market.

A schematic view of the site is shown in Figure 7. The site has two foundries where melted material is cast into blanks for three types of products: cylinder heads, cylinder blocks, and flywheels. After casting, the parts are deburred, cleaned, cylinder blocks are painted, and then the products are machined in a number of machining lines (ML) located in one of three machining centres (MC). Three product types are produced in the site, $P = \{A, B, C\}$, each with two main components denoted as $P1$ and $P2$. For some products, machining is done in parallel with MLs for the same product in several MCs. Each ML or MC, depending on locality, has either a combined or separate finished goods inventory (FGI). For example, in Figure 7, MLB25 denotes the ML for product B, component 2, and line number 5.

Figure 7 Schematic layout of the site for the SBC case study (see online version for colours)



From the MLs, the parts are either dispatched for export to other manufacturing sites or assembled into complete engines in the assembly. Due to the size of the components, the manufacturing area is large and MCs are dispersed requiring forklift transports between the different buildings. Ordering relies on a pull method from assembly, drawing material from intermediate FGI supplied from MLs and foundries. Planning in the foundries is resource optimised to fully utilise the foundry capacity by running batch production. The batch production requires larger storages between foundry and machining to ensure a stable supply.

3.4.1 Simulation model

The discrete-event simulation model is built in Siemens Tecnomatix Plant Simulation version 2201.003 using methods from Lidberg et al. (2019). A modelling technique is utilised with which each manufacturing line can be represented using four simulation objects and seven parameters while still retaining the characteristics of a more complex model. The lower object count reduces the number of generated events in the discrete-event simulation model and subsequently reduces the computational complexity. This leads to a faster simulation model and the total optimisation time is also reduced.

The simulation model represents the main product flows and components in the manufacturing site. Transports are modelled through the use of truck transports with speeds depending on position, scale accurate roadways, and loading times based on historical data. Shift times are specific for each production line, and the differences between them lead to large buffers between the production.

3.4.2 Optimisation setup

The optimisation is built in Python, using pymoo version 0.5.0 as the optimisation backend, NSGA-III as the optimisation algorithm, and distributed with the use of the Dask framework (Blank and Deb, 2020). Identifying improvements to the manufacturing site and its manufacturing facilities, in the form of manufacturing lines, is made through optimisation and the SCORE method (Bernedixen et al., 2015).

The SCORE method uses the theory of constraints by Goldratt and Cox (1984) to prescribe the removal of the limiting factors in the simulation model by SBO. Each parameter value, in this optimisation, is derived from the cycle time, availability, mean leadtime, max work in process, and average work in process of each manufacturing line, and is represented by a Boolean value. A parameter value of zero indicates that the parameter retains its original value, while a value of one sets the parameter to an improved state, representing the removal of a constraint. For example, an improvement for availability could be changing from a base value of 80%, to an improved value of 98%.

This improvement may be impossible to reach in the real system, but will give an indication of the largest bottleneck in the system. For 20 manufacturing lines and six parameters, gives 120 input variables across all products. The sum of these improvements constitute a minimisation objective in the optimisation. The identification of not only the bottleneck line, but also the bottleneck parameter of the line, offers prescriptive analytics and creates a recipe for improvement efforts. The other objectives are maximisation of throughput of the main product, variant B, combining the output from two parallel lines, and the minimisation of leadtime, measured in seconds, of variant B. The objectives are as follows:

1 maximise output

2 minimise leadtime

3 minimise improvements

Maximising the output and minimising leadtime is correlated with maximising the number of improvements. As improving all the parameters is not feasible, neither technically nor financially for the company, the improvement objective should be minimised. Objective three is therefore conflicting with objectives one and two.

The limitations of the optimisation study is the focus on the main product variant. This is to reduce the complexity of the optimisation and the analysis. Product B is the most important product due to the production volume.

4 Results

This section will present the knowledge-extraction process from two optimisation studies on industrial use cases. The first in Subsection 4.1 and the second in Subsection 4.2, using the open-source tool for reproducible knowledge extraction presented in Lidberg et al. (2022) updated with requirements from industrial decision makers. The tool is publicly available as version 0.0.1.9000 from Lidberg (2022).

4.1 Knowledge-extraction for MFC

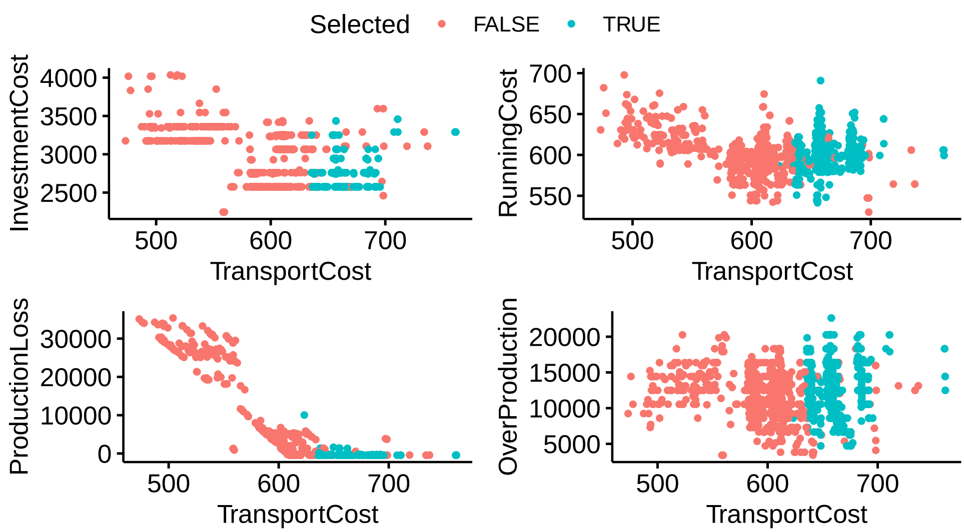
The optimisation was run for 100 generations, with a population of 240, and produced 1076 Pareto-optimal solutions. Initial values were supplied as $Site1\_ProdA\_1 = 4$, $Site2\_ProdA\_1 = 2$, and $Site2\_ProdB\_1 = 5$. Following the steps in Figure 1, the data was imported in the open-source tool. Filtering and clustering were not used in this use case as pre-processing steps.

Knowledge-extraction methods can be utilised to gain prescriptive analytics about the MOO data. The knowledge can be encapsulated as rules or as connections between the decision space and objective space. The *ProductionLoss* objective, i.e., the lack of products delivered to the customer, is positive if products are missing and negative if demands are met. To understand what differentiates solutions with positive and negative values for *ProductionLoss* can help decision makers when deciding on the manufacturing footprint. An analysis of the effect of *ProductionLoss* will be performed with FPM using the open-source tool. The selected solutions matching the most significant rule found by FPM can be seen in Figure 8.

From the requirements stated in Subsection 3.2, the number of alternatives when taking a decision is important. A range of solutions is not desired by decision makers. Instead, they want a limited set of solutions placed in a use case context. After the knowledge extraction has been performed, the dataset is filtered on *ProductionLoss* < 700, as the remaining solutions are not deemed to be relevant to be implemented.

The use case context for MFC is an implementation plan for new production across the different sites. As each use case can be highly unique, the analysis will need to be further developed outside of the open-source tool while still using the R language. An example of this is shown in Figure 9, where Figure 9(a) represents the minimum investment cost, Figure 9(b) the minimum transport cost, while Figure 9(c) represents the minimum running cost in the filtered objective space. The solution in Figure 9(a) shows a centralised approach with production concentrated in *Site1*. To reduce *RunningCost*, investment into more localised production is necessary.

Figure 8 Solutions selected by the most significant rule: $2028\_Site2\_ProdA\_1 = 0$ and $2030\_Site1\_ProdB\_2 > 2$ and $2030\_Site3\_ProdA\_1 = 2$ (see online version for colours)



Comparing the objective function values for the two different solutions with minimum values, shown in Table 3, *InvestmentCost* is directly proportional to *RunningCost* and indirectly proportional to *TransportCost*. As new production capacity is installed, the need for cross-site transportation is lower. For transporting, *product B* is more costly to transport than *product A*.

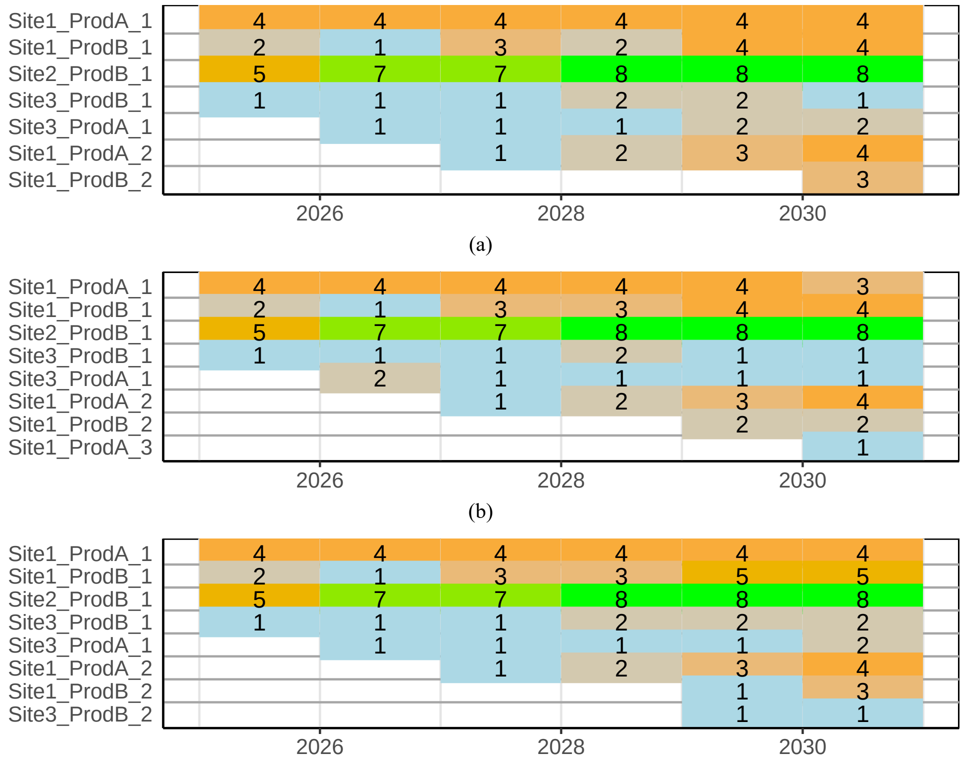
Table 3 Comparing objective values for solutions with minimum *TransportCost* and with minimum *RunningCost*

TransportCost	*InvestmentCost*	*RunningCost*	*Prod.Loss*	*OverProd.*
605.490	3,065	619.3	526	15,068.71
658.063	2,760	559.6	–334	15,253.14

Note: *ProductionLoss* and *OverProduction*, abbreviated for brevity.

All solutions after filtering can be seen in Figure 10, where an elbow effect is visible in the upper-left graph, showing that lower transport costs are impossible to achieve without investing in new production. *InvestmentCost* is primarily discrete due to the costs for installing new production, but it also has some economy of scale, showing as solutions not horizontally aligned with the remaining solutions.

Figure 9 Alternative implementation plans for MFC, showing minimum solutions for three objectives, (a) minimum InvestmentCost (b) minimum TransportCost (c) minimum RunningCost (see online version for colours)



(a) (b) (c)

Note: Filtered to *ProductionLoss* < 700.

The knowledge discovery process, using FPM, resulted in rules on how to minimise the *ProductionLoss* in the manufacturing footprint. Along with the knowledge from these rules, alternative implementation plans can be offered to the decision makers. The use case context can be presented directly from the data in the form of an implementation plan.

4.2 Knowledge-extraction for SBC

To analyse the data, the SBI method begins with a high-level clustering using decision trees. Applying this to the SBC dataset, we get the rules seen in Figure 11, where the leaf nodes represent four sequential cluster groups, seen in Figure 12. The percentages in the leaf nodes represent how many solutions are in each group. Rules are extracted from all available data in the optimisation, including the dominated solutions.

The most divisive rule, i.e., the rule that divides the solutions the most, is F11_Avb. To further divide the dataset in four groups, the rules MLB21_CT and F12_Avb are used. The first and second leaves are connected to a lower *MinLT* and higher *MaxOut* and *MinImp*, while the two rightmost leaves are connected to the opposite. The decision maker can use this information to express their preference for either objective, e.g., wanting a lower number of improvements would therefore lead to improving F11 and MLB21.

Figure 10 Remaining solutions after filtering *ProductionLoss* < 700

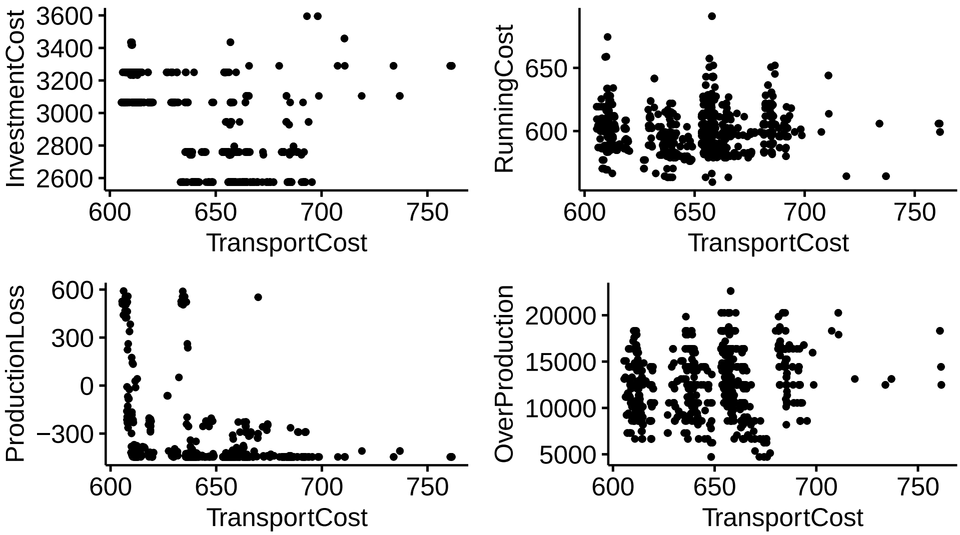


Figure 11 Decision tree showing rules for the SBC (see online version for colours)

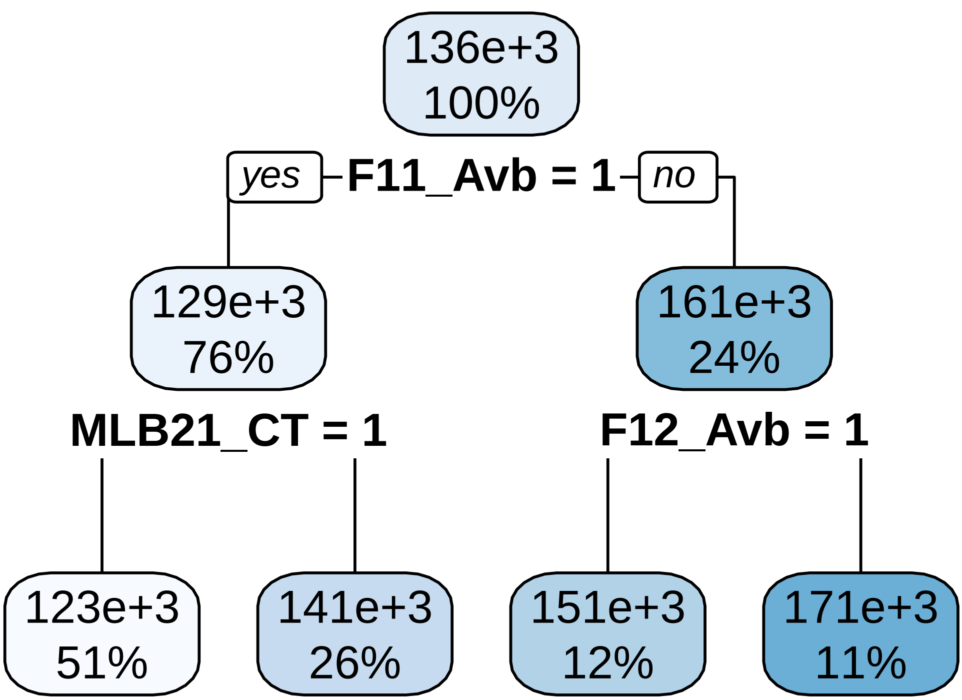


Figure 12 Applied clusters from the decision tree (see online version for colours)

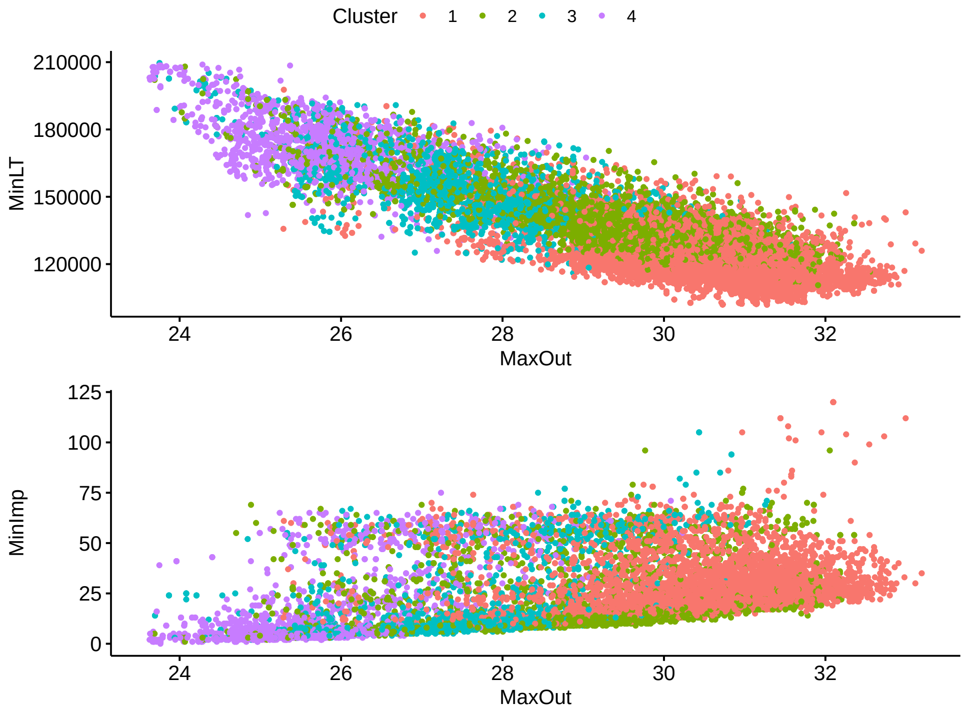


Figure 13 Frequency of bottleneck parameter in Pareto-optimal solutions for SBC

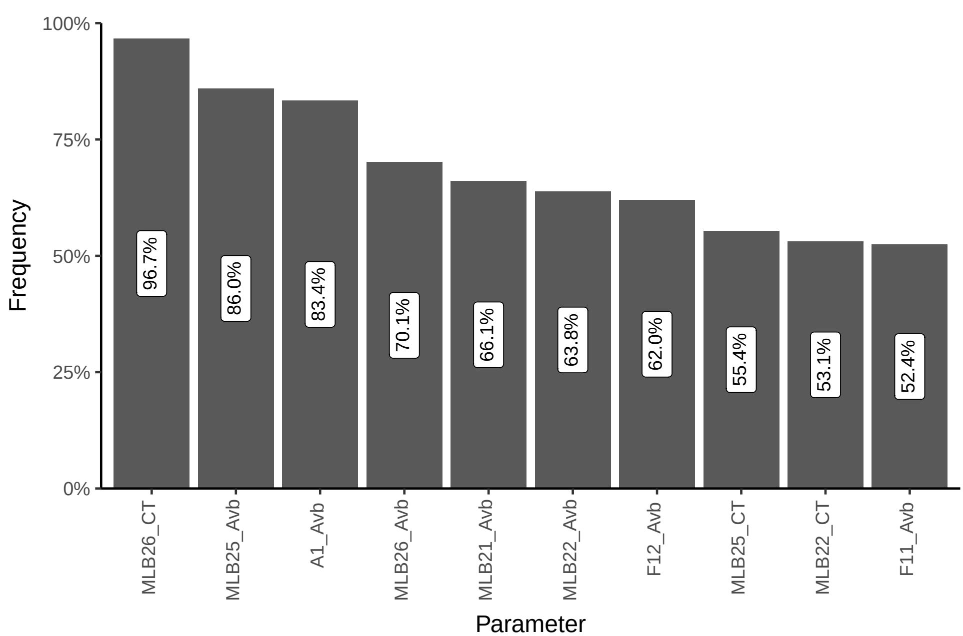
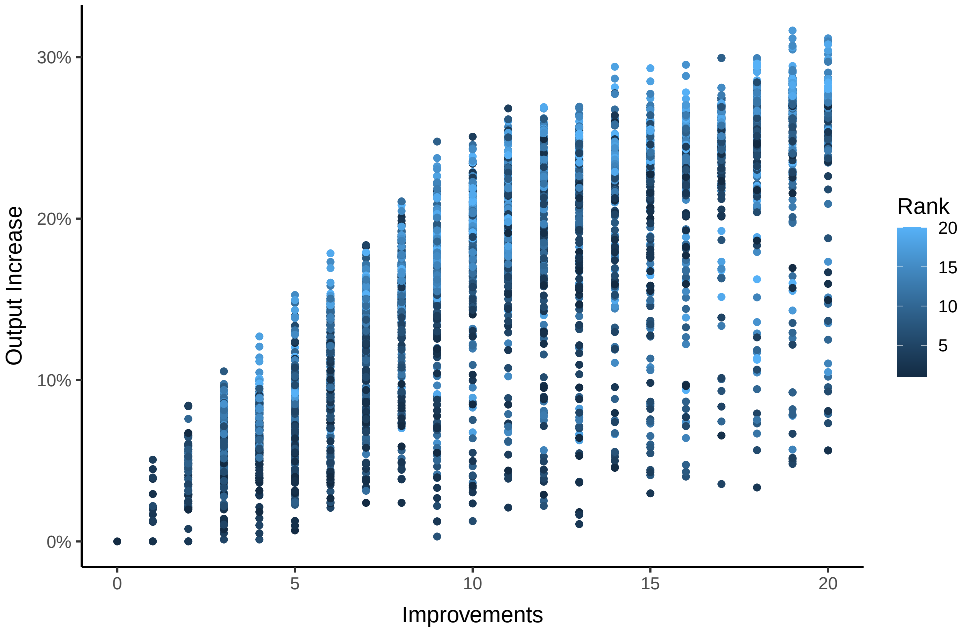


Figure 14 Potential increase in output by removing the biggest bottleneck in turn (see online version for colours)

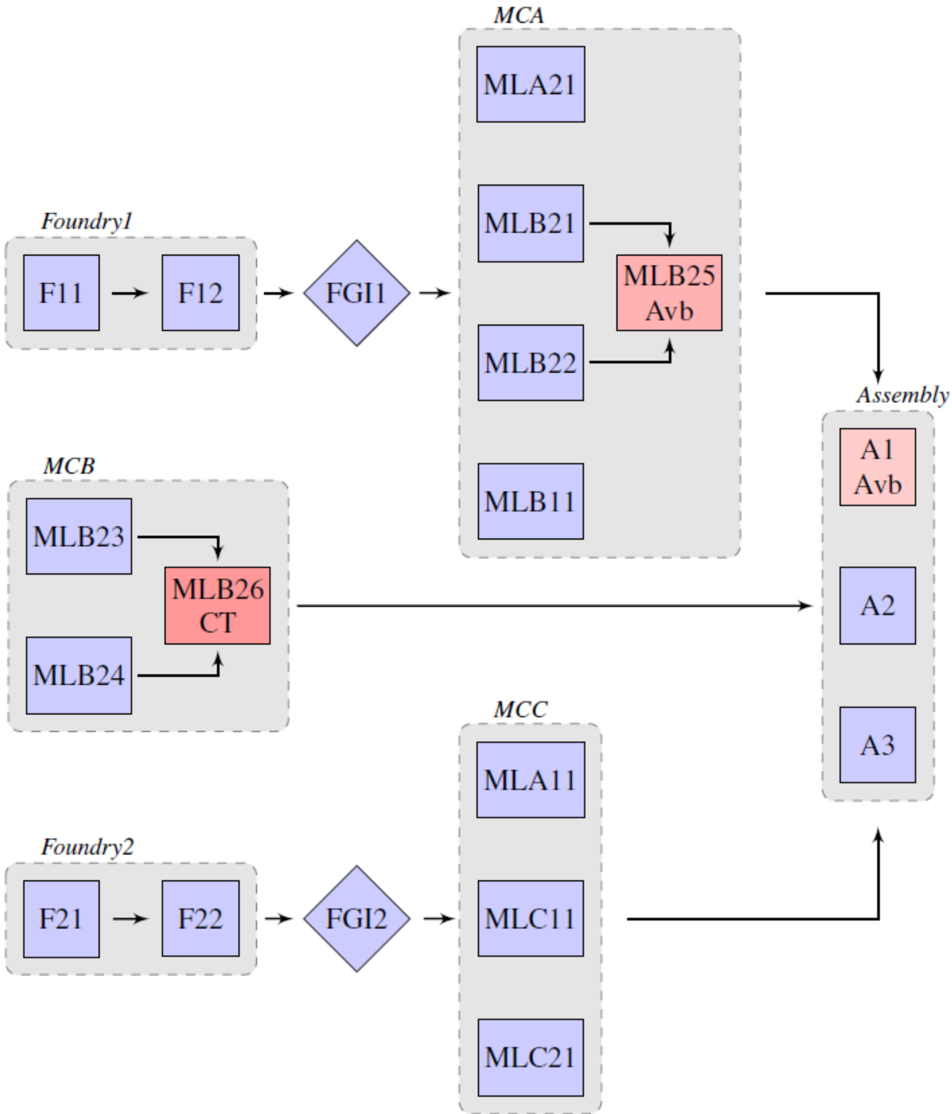


Note: Ranking is for all objectives, and not only for *MaxOut* and *MinImp*.

The second analysis for the SBC data is a SCORE analysis resulting in a ranking of bottlenecks, which allows for a prioritisation of improvements. See Figure 13 for an analysis of the bottleneck parameters in SBC. The frequency bar chart shows the prevalence of a parameter in the Pareto-optimal solutions. The more common a parameter is, the more likely it is to be the biggest bottleneck in the complete system. The most common parameter is MLB26_CT with 96.7% followed by MLB25_Avb with 86.0% and A1_Avb with 83.4% of solutions in the Pareto-optimal front. The main focus should be placed on the MLs, with MLB26 focused on processing time improvements, and MLB25 on increasing the availability of the line. As both MLB25 and MLB26 are production lines following two parallel production lines, disturbances in MLB25 and MLB26 has a higher performance effect on the total system. In assembly of the main product, improvements to availability should be prioritised.

The potential improvements to output is shown in Figure 14 where the number of improvements is compared to the relative increase in output. The potential improvement is attained when completely removing a bottleneck and will probably not be possible to reach in the real system. The improvements also points to the importance of removing bottlenecks in the correct order for improving the output of the complete system. Changing the order of the improvements would lead to suboptimal outcomes, and is most likely the current unintended way of handling improvements. The open-source tool supports these visualisations, but to deliver knowledge to the decision maker additional effort external to the open-source tool is required.

Figure 15 Improvements to be prioritised indicated in red, with parameters to improve for the three greatest bottlenecks (see online version for colours)



The use case context of SBC is an improvement prioritisation for the manufacturing site, seen in Figure 15. Improvements to be made are listed under the name of the production line, along with the prioritisation of the bottleneck. The deeper shade of red, the more prioritised the improvement. This gives a decision maker, aiming to increase the output of the total system, help in prioritising the limited resources for improvement work.

5 Conclusions and future work

Multi-objective optimisation methods, such as NSGA-III, can iteratively optimise simulation models by modifying model inputs. The outcome of the optimisation process is a large optimisation dataset, which can be difficult to interpret. By applying knowledge-extraction methods, such as FPM and SBI, important decision variables can be identified. Knowledge-extraction methods can provide the name, as well as the value, of important parameters in the form of rules. These rules provide recipes to a decision maker on how to optimise their production processes according to the decision maker's preference. Industrial decision makers participating in this study expressed a preference for more data-driven decisions, keeping options open until the decision should be made. When the time for a decision comes, the number of alternatives should be limited and presented in an appropriate context for the case.

Acquiring, modifying, analysing, extending, and improving closed-source software is difficult or impossible and therefore the open-source application created is useful to successfully replicate results. Reproducing the analysis is possible through the transformation of steps in the analysis into R code. This is important for both transparency and validity of research results, and helps mitigate one issue of reproducibility in science. Having knowledge-extraction methods available through an open-source tool also lowers the barrier of entry and is an important contribution to both science and practice of this work. To demonstrate the utility of the open-source tool, it was used in two industrial use cases, through which important knowledge was gained about the systems studied.

For future research, finding not only the parameter to improve on the site level but which equipment to improve on the production line level is of interest. One research approach is automatic model generation for the production line and site level. By collecting data from the actual system, models can be automatically created and updated, allowing for continuously supplying decision support to decision makers. To increase the scale of the optimisation, the removal of proprietary software from the optimisation would also be needed. The prioritisation of SBC could be further improved by supplying specific values, or thresholds, to reach for each parameter in the optimisation instead of current Boolean values. Allowing the decision maker to identify when a bottleneck is improved enough to continue with the bottleneck in the system. In addition, to further improve the open-source tool, industrial practitioners should use and evaluate the tool for their needs and their feedback should be recorded through a questionnaire.

References

Aarts, A.A. et al. (2015) 'Estimating the reproducibility of psychological science', *Science*, August, Vol. 349, pp.aac4716–aac4716.

Agrawal, R. and Shafer, J. (1996) 'Parallel mining of association rules', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, pp.962–969.

Agrawal, R. and Srikant, R. (1995) 'Mining sequential patterns', in *Proceedings of the Eleventh International Conference on Data Engineering*, IEEE Comput. Soc. Press, pp.3–14.

Amouzgar, K., Bandaru, S., Andersson, T. and Ng, A.H.C. (2018) 'A framework for simulation-based multi-objective optimization and knowledge discovery of machining process', *The International Journal of Advanced Manufacturing Technology*, October, Vol. 98, pp.2469–2486.

Bandaru, S. and Deb, K. (2013) 'Higher and lower-level knowledge discovery from Pareto-optimal sets', *Journal of Global Optimization*, October, Vol. 57, pp.281–298.

Bandaru, S. and Deb, K. (2016) 'Metaheuristic techniques', in Sengupta, R.N., Gupta, A. and Dutta, J. (Eds.): *Decision Sciences: Theory and Practice*, Chap. 11, 1st ed., CRC Press, pp.693–749.

Bandaru, S., Ng, A.H. and Deb, K. (2017a) 'Data mining methods for knowledge discovery in multi-objective optimization: part A – survey', *Expert Systems with Applications*, March, Vol. 70, pp.139–159.

Bandaru, S., Ng, A.H. and Deb, K. (2017b) 'Data mining methods for knowledge discovery in multi-objective optimization: part B – new developments and applications', *Expert Systems with Applications*, Vol. 70, pp.119–138.

Bergmann, S., Feldkamp, N. and Strassburger, S. (2017) 'Emulation of control strategies through machine learning in manufacturing simulations', *Journal of Simulation*, February, Vol. 11, pp.38–50.

Berk, R.A. (2009) 'Data mining within a regression framework', in *Data Mining and Knowledge Discovery Handbook*, pp.209–230, Springer, Boston, MA, USA.

Bernedixen, J., Pehrsson, L., Ng, A.H. and Antonsson, T. (2015) 'Simulation-based multi-objective bottleneck improvement: towards an automated toolset for industry', in *Proceedings of the 2015 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers Inc., Piscataway, New Jersey, pp.2183–2194.

Blank, J. and Deb, K. (2020) 'pymoo: multi-objective optimization in Python', *IEEE Access*, Vol. 8, pp.89497–89509.

Breiman, L. (1996) 'Bagging predictors', *Machine Learning*, Vol. 24, No. 2, pp.123–140.

Breiman, L. (2001) 'Random forests', *Machine Learning*, Vol. 45, No. 1, pp.5–32.

Breiman, L., Friedman, J., Stone, C.J. and Olshen, R.A. (1984) *Classification and Regression Trees*, 1st ed., Routledge.

Dudas, C., Ng, A.H., Pehrsson, L. and Boström, H. (2014) 'Integration of data mining and multi-objective optimisation for decision support in production systems development', *International Journal of Computer Integrated Manufacturing*, Vol. 27, No. 9, pp.824–839.

Goldratt, E.M. and Cox, J. (1984) *The Goal: A Process of Ongoing Improvement*, Gower Publishing, Aldershot.

Höppner, F. (2010) 'Association rules', in *Data Mining and Knowledge Discovery Handbook*, pp.299–319, Springer, Boston, MA, USA.

Hakanen, J., Radoš, S., Misitano, G., Saini, B.S., Miettinen, K. and Matković, K. (2022) 'Interactivized: visual interaction for better decisions with interactive multiobjective optimization', *IEEE Access: Practical Innovations, Open Solutions*, Vol. 10, pp.33661–33678, Citation Key: 9739720.

Jain, S., Lechevalier, D., Woo, J. and Shin, S-J. (2015) 'Towards a virtual factory prototype', in Yilmaz, L., Chan, W.K.V., Moon, I., Roeder, T.M.K., Macal, C. and Rossetti, M.D. (Eds.): *Proceedings of the 2015 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ, USA, pp.2207–2218.

Liao, Y., Deschamps, F., Loures, E.d.F.R. and Ramos, L.F.P. (2017) 'Past, present and future of Industry 4.0 – a systematic literature review and research agenda proposal', *International Journal of Production Research*, June, Vol. 55, pp.3609–3629.

Lidberg, S., Aslam, T., Pehrsson, L. and Ng, A.H. (2018) 'Evaluating the impact of changes on a global supply chain using an iterative approach in a proof-of-concept model', in Thorvald, P. and Case, K. (Eds.): *Advances in Manufacturing Technology, Advances in Transdisciplinary Engineering*, IOS Press, Skövde, Sweden, Vol. 32, pp.467–472.

Lidberg, S. (2022) *SCORER* [online] https://zenodo.org/record/7316922, https://github.com/verbalins/SCORER.

Lidberg, S., Aslam, T., Pehrsson, L. and Ng, A.H. (2019) 'Optimizing real-world factory flows using aggregated discrete-event simulation modelling: creating decision-support through simulation-based optimization and knowledge-extraction', *Flexible Services and Manufacturing Journal*, Vol. 32, No. 4, pp.1–25.

Lidberg, S., Frantzén, M., Aslam, T. and Ng, A.H. (2022) 'A knowledge extraction platform for reproducible decision-support from multi-objective optimization data', in Ng, A.H.C., Syberfeldt, A., Högberg, D. and Holm, M. (Eds.): *Proceedings of the 10th Swedish Production Symposium*, IOS Press, April, pp.725–736.

Mahmoodi, E., Fathi, M. and Ghobakhloo, M. (2022) 'The impact of Industry 4.0 on bottleneck analysis in production and manufacturing: current trends and future perspectives', *Computers & Industrial Engineering*, Vol. 174, p.108801.

Mazumdar, A., Otayagich, S. and Miettinen, K. (2022) 'Interactive evolutionary multiobjective optimization with modular physical user interface', in *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO'22*, Association for Computing Machinery, New York, NY, USA, pp.1835–1843.

Mittal, S., Saxena, D.K., Deb, K. and Goodman, E.D. (2022) 'A learning-based innovized progress operator for faster convergence in evolutionary multi-objective optimization', *ACM Transactions on Evolutionary Learning and Optimization*, March, Vol. 2, pp.1–29.

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W. and Ueda, K. (2016) 'Cyber-physical systems in manufacturing', *CIRP Annals*, January, Vol. 65, pp.621–641.

Ng, A.H., Deb, K. and Dudas, C. (2009) 'Simulation-based innovization for production systems improvement: an industrial case study', in Rosén, B-G. (Ed.): *Proceedings of the International 3rd Swedish Production Symposium, SPS'09*, The Swedish Production Academy, Skövde, Göteborg, Sweden, 2–3 December, pp.278–286.

Prajapat, N., Turner, C., Tiwari, A., Tiwari, D. and Hutabarat, W. (2020) 'Real-time discrete event simulation: a framework for an intelligent expert system approach utilising decision trees', *The International Journal of Advanced Manufacturing Technology*, Vol. 110, Nos. 11–12, pp.2893–2911.

Rokach, L. and Maimon, O. (2010) 'Classification trees', in Maimon, O. and Rokach, L. (Eds.): *Data Mining and Knowledge Discovery Handbook*, pp.149–174, Springer, Boston, MA, USA.

Shavazipour, B., López-Ibáñez, M. and Miettinen, K. (2021) 'Visualizations for decision support in scenario-based multiobjective optimization', *Information Sciences*, Vol. 578, pp.1–21.

Thomas, P., Suhner, M-C. and Thomas, A. (2014) 'CART for supply chain simulation models reduction', in Grabot, B., Vallespir, B., Gomes, S., Bouras, A. and Kiritsis, D. (Eds.): *Advances in Production Management Systems, Innovative and Knowledge-Based Production Management in a Global-Local World*, Springer, Berlin, Heidelberg, Vol. 440, pp.530–537.

Thomas, P., Suhner, M-C. and Thomas, A. (2015) 'Reduced simulation model for flow analysis in a sawmill internal supply chain', in Framinan, J., Perez Gonzalez, P. and Artiba, A. (Eds.): *2015 International Conference on Industrial Engineering and Systems Management (IESM)*, IEEE, pp.1319–1328.