SPS2022
A.H.C. Ng et al. (Eds.)

doi:10.3233/ATDE220191

A Knowledge Extraction Platform for Reproducible Decision-Support from Multi-Objective Optimization Data

Simon LIDBERG [a,b,1] Marcus FRANTZÉN [c] Tehseen ASLAM [a] Amos H.C. NG [a]
[a] *School of Engineering Science, University of Skövde, Skövde, Sweden*
[b] *Volvo Group Trucks Operations, Skövde, Sweden*
[c] *Department of Industrial and Materials Science, Chalmers University of Technology, Gothenburg, Sweden*

Abstract. Simulation and optimization enables companies to take decision based on data, and allows prescriptive analysis of current and future production scenarios, creating a competitive edge. However, it can be difficult to visualize and extract knowledge from the large amounts of data generated by a many-objective optimization genetic algorithm, especially with conflicting objectives. Existing tools offer capabilities for extracting knowledge in the form of clusters, rules, and connections. Although powerful, most existing software is proprietary and is therefore difficult to obtain, modify, and deploy, as well as for facilitating a reproducible workflow. We propose an open-source web-based application using commonly available packages in the R programming language to extract knowledge from data generated from simulation-based optimization. This application is then verified by replicating the experimental methodology of a peer-reviewed paper on knowledge extraction. Finally, further work is also discussed, focusing on method improvements and reproducible results.

Keywords. multi-objective optimization, knowledge extraction, industry 4.0, decision-support, industrial optimization

1. Introduction

Industry and academia are increasingly focused on the concept of Industry 4.0 [1], even though interest for Simulation-Based Optimization (SBO) have reduced since 2016 [2]. For industry, adopting the paradigm early means getting a competitive edge over other businesses, and for academia there exists ample funding opportunities in the research area due to national and international focus. Industry 4.0 includes cyber-physical systems, and more importantly, cyber-physical production systems [3]. Simulation models are therefore also increasing in use, and to attain the benefits of prescriptive analysis [4], the addition of optimization and knowledge extraction would need to be made. Analysis of the large datasets from SBO have been described as Simulation-Based Innovization

[1]Corresponding Author: Simon Lidberg, Högskolevägen, BOX 1231, Skövde, Sweden; E-mail: simon.lidberg@his.se

(SBI) or Knowledge-Driven Optimization (KDO) [5,6]. The knowledge generated from the optimization data increases the usefulness of the optimization and can be utilized to prescribe actions increasing the effectiveness of production systems, instead of relying on guesswork or what-if analysis.

Academic research is facing a reproducibility crisis, most famously shown in the field of Psychology [7], where decades' worth of academic output in the form of peer-reviewed literature reported results which are not reproducible. Academic journals are countering this by embracing open science, where transparency and reproducibility are increasingly important; encouraging researchers to publish supplementary data, e.g., raw data, processed data, and code for analysis to reproduce and corroborate findings. Analysis steps only performed in a user interface or particular visualization software are difficult to document and reproduce; instead, each step should be expressed and be reproducible in code. This is important both for publishing, and for the workflow for each researcher. The creation of an open-source knowledge extraction platform is therefore relevant, both from the perspective of industry and academia. It would allow for faster decision-support delivered to strengthen the competitiveness of industry, and record steps taken in visual analytics to later be reproduced, which benefits science and academia.

The goal of this application study is the development of an open-source tool to support the knowledge extraction process of MOO data, with the added benefit of captur- ing the analysis in repeatable code. The open-source tool is then evaluated by reproducing analysis from two different journal articles presenting knowledge extraction methods.

2. Background

In this section, the background for Multi-Objective Optimization (MOO) datasets and their special properties will be presented. This section will also present theoretical background to the implemented methods of this study for knowledge extraction of MOO datasets. SBI is based on decision trees explained in the section Section 2.1, and Flexible Pattern Mining (FPM) is based on frequent itemset mining and association rule learning presented in Section 2.2.

Creating knowledge by applying data mining methods to MOO data differ from other data mining applications due to the properties of the MOO dataset [8]. These special properties, and the requirements to the data mining methods applied to generate knowledge from the datasets, can be summarized as follows:

- Handling of two separate spaces, objective and decision space
- Decision maker involvement
- Knowledge representation
- Different variable types in the data
- Problem parameters

Visualizations of MOO datasets are performed in the objective space, showing the results from the optimization, 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 probably 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 optimization 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 of presenting it to a decision-maker and converting 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 [8].

MOO problems can also include problem parameters in addition to variables and objectives. These problem parameters are not altered by the optimization algorithm but represent external controlling parameters such as constraints, variable bounds or even objective functions. By altering these parameters and then running the optimization again, higher-level knowledge can be obtained [9].

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 [10]. This study will focus on Classification and Regression Trees (CART), due to their ability to generate both classification and regression trees [11].

The data in decision trees is partitioned to maximize 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. Maximizing homogeneity in each partition is achieved by minimizing 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 where the tree is shortened by removing leaves, "trading accuracy for simplicity" [10]. Pruning has been shown to improve the generalization performance of the decision tree [11]. Applications to improvement of manufacturing can be found in [12], [13], [14], and [15].

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 [10]. The main problem with decision trees, is their tendency to overfit the data, meaning that the model has become too specialized and will not perform well when encountering new data [16]. Reducing overfitting can be accomplished by constructing average results over random samples of the data, where an early development was 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 [17], eventually leading to Random Forests [18].

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 [19]. 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 [20].

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 [21]. 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 is evaluated (0,1) 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 [21]. 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 optimization to converge faster to some preferred region in the objective space [22]. In contrast, the current paper is focused on offline KDO that the generated knowledge and visualizations are used to assist human decision-makers (i.e., human learning).

3. Method

Using SBO has several benefits, but generates large amounts of data, which can be difficult to interpret and present to a decision-maker [23]. Without proper analysis and data-mining, knowledge about the underlying system will be impossible to attain and some benefits of SBI will be lost. Quick and high-quality visualizations can also aid in the exploration of a dataset when performing an initial analysis of the problem or preparing data for publication. Existing solutions for SBO data exploration and knowledge extraction are powerful, but many are closed source. Acquiring, modifying, analyzing, and extending closed source software is difficult or impossible and therefore an open-source alternative is beneficial.

The goal of this application study is the development of an open-source tool to support the knowledge extraction process of SBO data, with the added benefit of capturing the analysis in repeatable code. This includes analyzing SBO data and performing the steps from SBI using the R language, revealing new knowledge about the data and the underlying system. To evaluate the efficacy of the open-source tool, two studies will be reproduced; the study presenting the SBI method [24] and FPM [21]. Both journal articles utilize the ACM simulation model and data [25]. The model is based on an industrial

problem where the interesting objectives are minimization of total annual running cost, minimization of total investment cost for improvements, and minimizing total buffer capacities. These objectives are evaluated through settings for improvements, either processing time Ip or up-time Iu, summarized by $SumI$, and the number of buffer places in the buffers $B_{1...31}$. The original article [24] follows the SBI process of:

1. Simulation-based multi-objective optimization
2. Data pre-processing
3. Pattern detection
4. Interpretation and evaluation

Running the SBO is not the focus of this study and is therefore left to other tools. The open-source tool has a number of different views supporting the process steps of SBI, as 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 *Visualize Data*. Finally, the last stage of SBI, Interpretation and evaluation, is handled in the view *Rule Extraction*. The resulting data from the various stages can be exported from the final view called *Export Data*. For FPM, *Import Data*, *Visualize Data*, and *Rule Extraction* are used.

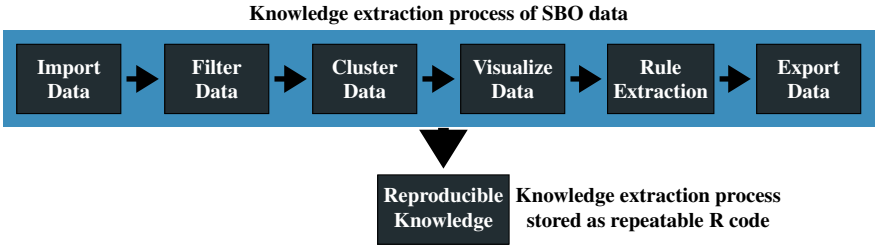

Figure 1. Knowledge extraction captured in code, allowing for a repeatable process.

4. Results

The following section will present comparisons made between the R implementation of SBI and FPM, and their original publications [24,21]. The source code for the open-source tool is publicly available as version 0.0.0.9000 [26].

4.1. Import Data

The intended use of the application is the analysis of optimization data obtained through SBO. The *Import Data* view contains two parts, uploading data and assigning the parameters to one of three groups: Objectives, Inputs, and/or Outputs seen in Figure 2. These groupings persist through the remainder of the analysis.

Uploading data lets the user choose a file, restricted to semicolon Comma Separated Values (CSV)-files as the supported format. The Distance metric [24] can be added to the dataset by computing the Euclidean distance for each point to the closest point on the Pareto front.

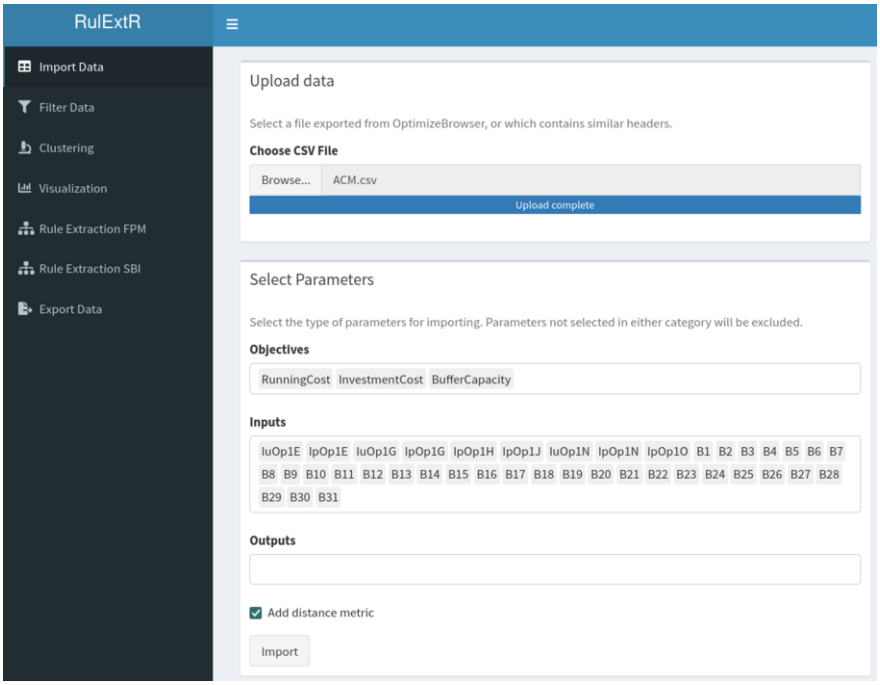

Figure 2. *Import Data* view in the application.

4.2. Filter Data

The resulting datasets for SBO are usually large, with an iteration count in the tens of thousands. Large amounts of data will complicate the visualization and hinder understanding for analysts and algorithms. Due to the iterative nature of the genetic algorithms most commonly used for SBO, the solutions early in the optimization 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, visualizing, and extracting knowledge can be beneficial. To support these scenarios, the *Filter Data* view was implemented, in the form of a table view. No filtering was applied to the dataset for this application study.

4.3. Cluster Data

Identifying clusters in the data provides knowledge about which parameters influence the result. Four categories of methods are available: Hierarchical, Partitioning, Density, and Decision Trees, each tab containing different algorithms, shown in Figure 3. Each method has different parameters and the inputs will change to reflect that when selecting a new method. The automatic suggestion for the number of clusters to use is available for some clustering methods, assisting the user to select the appropriate number of clusters.

Comparison of the results for this evaluation will be conducted using the data from the finished SBO; thus, step one is not relevant here. As an additional step for this case study, a top-level decision tree analysis was conducted to separate the solutions. The ar-

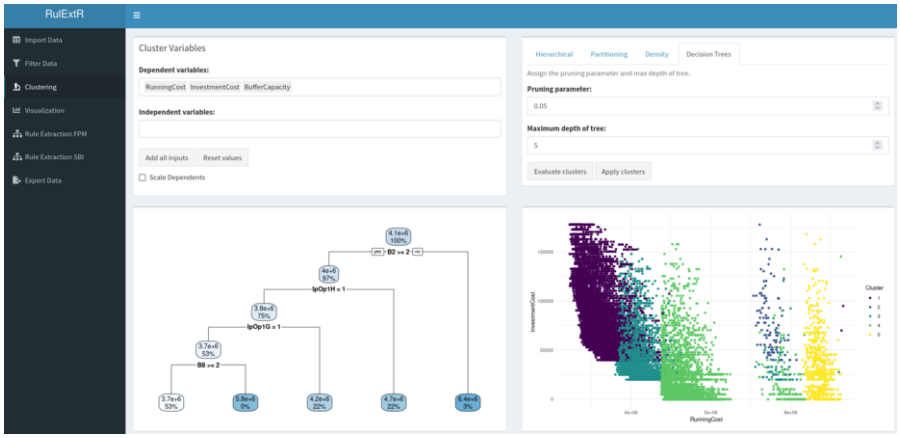

Figure 3. Applying clustering based on decision trees in the *Cluster Data* view.

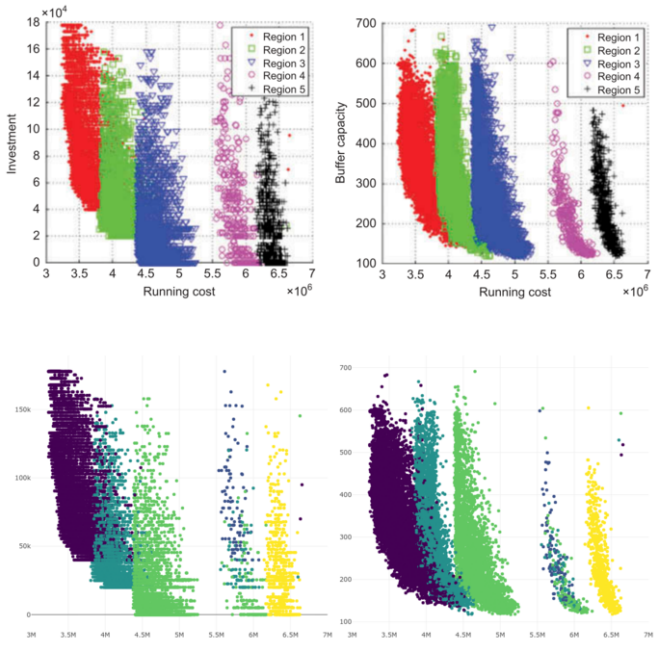

Figure 4. Clustering comparison with [24] on top, and the results from the application study below.

ticle suggested 0.05 as the threshold value when analyzing differences between decision variables, along with a max depth of four for the decision tree. Worth noting is the difference between the identified decision variables between this experiment and the article. The experiment, in Figure 4, shows Ip_{op1H} and Ip_{op1G} as important, while the article notes Ip_{op1O} and Ip_{op1N}, showing the problem with replicating results where code is not available. Clustering for FPM is not required, and is therefore skipped.

4.4. Visualize Data

Objectives in SBO are often conflicting, and visualization is difficult with more than three objectives, i.e., dimensions. Identifying clusters and structure in data can be achieved using several individual visualizations, combining them gives even more insight. With brushing and selecting in the visualization, a decision-maker can select the solutions which are relevant and most important to them when taking new decisions based on data.

The *Visualize 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 5. These components share filtering, coloring, 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 selecting interesting areas and solutions.

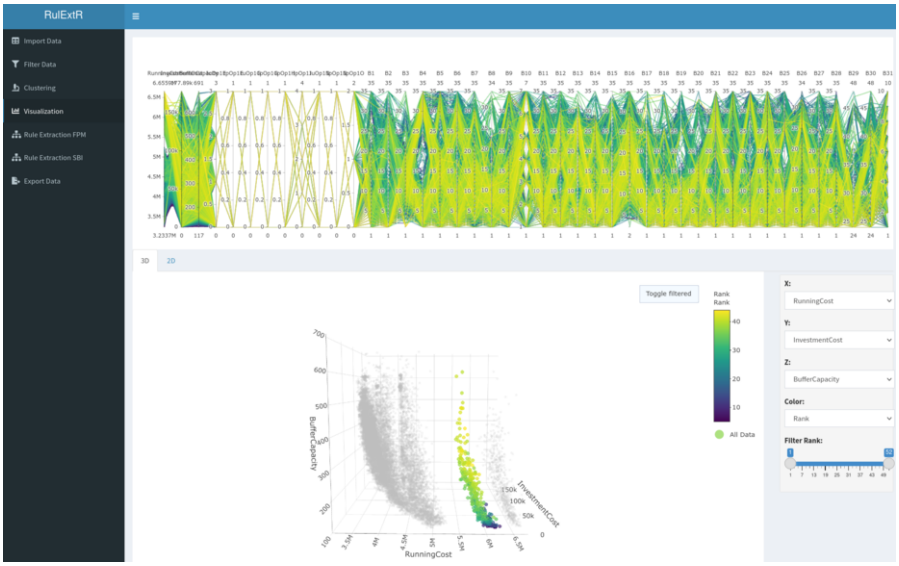

Figure 5. Parallel Coordinates chart and visualizing an interesting region in the 3D scatter plot.

The 2D tab shows $n-1$ two-dimensional scatter plots, where n is the number of objectives set in the *Import Data* view. As shown in Figure 6, the X-axis is always the first objective, while the Y-axis changes to show the remaining objectives.

4.5. 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 application study incorporates both rule generation by SBI and FPM.

4.5.1. FPM

For FPM, two parameters are utilized when creating rules, minimum significance and rule level. Minimum significance is an internal parameter to FPM setting the threshold

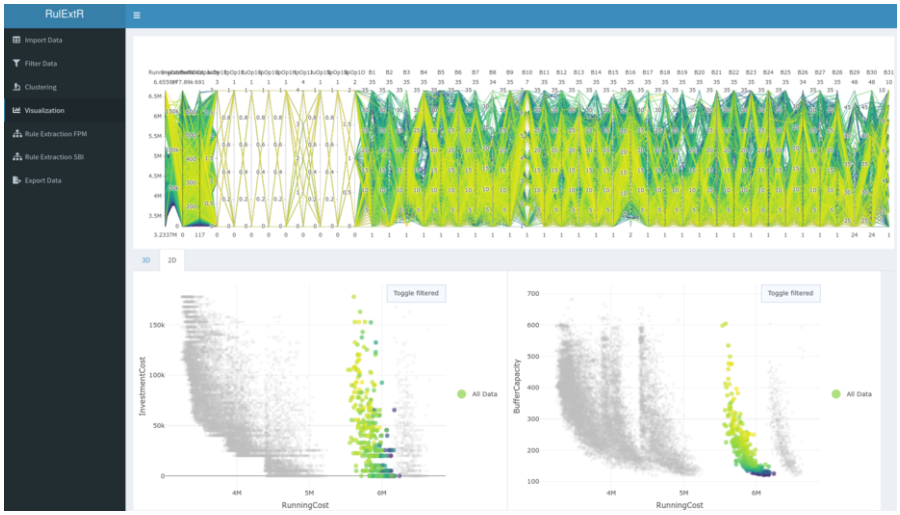

Figure 6. Selecting an interesting region of the dataset on the 2D plot.

of support for each rule to be included in the final rule list. The second parameter, rule levels, sets the level of rules to be generated. The default level, one, generates simple rules such as $B_1 = 1$. Increasing the level, means that FPM will consider combinations of two rules, i.e., $B_1 = 1 \ \& \ Ip_{op1O} = 0$, when evaluating support. The levels are inclusive, meaning that when set to two, rules of size one are also included in the evaluation. The rules are presented in order of decreasing importance through the FPM implementation in R, shown in Figure 7. Solutions are either selected in the *Visualize Data* view or assigned through a reference point. Rule generation through FPM shows $B8 = 1$, with 100 % matching the selected solutions, and 0.17 % for the unselected solutions. This exactly matches the rule generated in Table 17 [21, p. 137] for the same dataset.

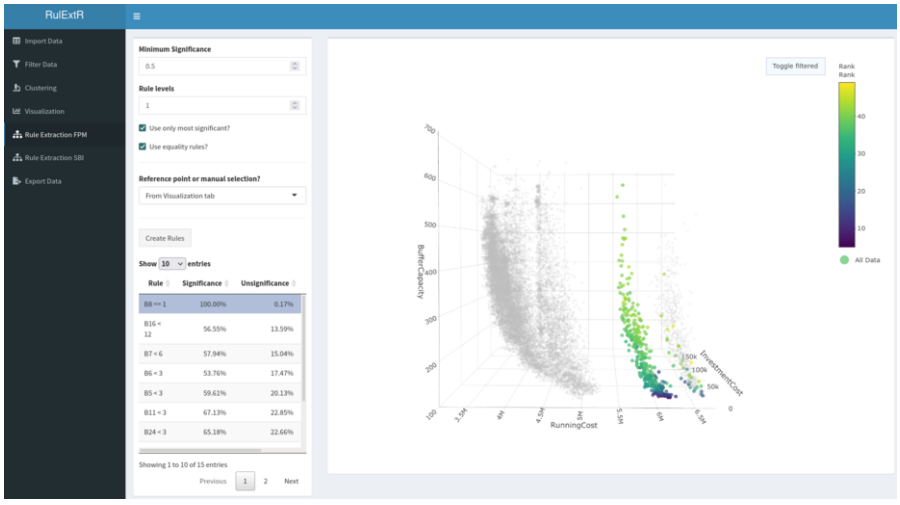

Figure 7. Rule generation through FPM showing $B8 == 1$, with 100 % matching the chosen solutions.

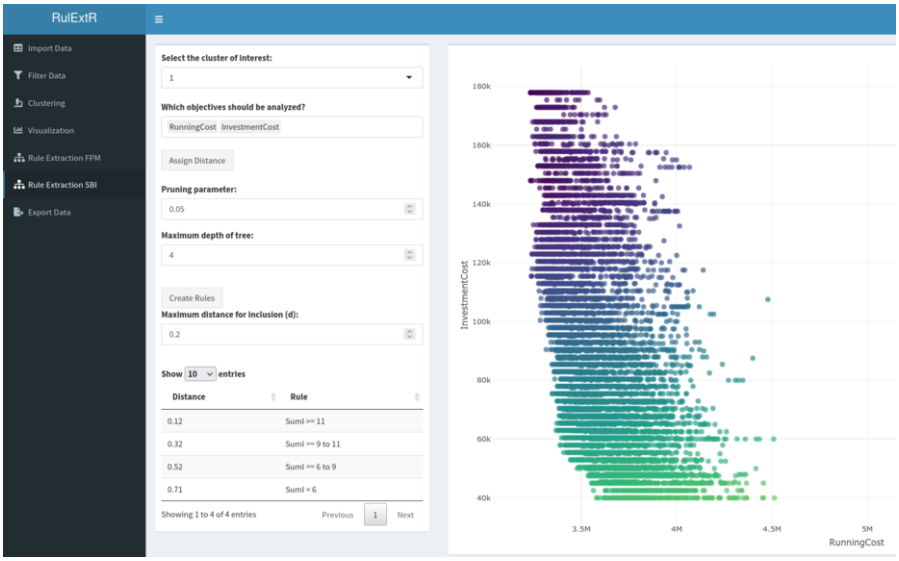

Figure 8. Rule generation by SBI on the first sub cluster of RCI, showing $\sum(I_u + I_p) > 11$ (SumI) as the most important rule.

4.5.2. SBI

Rule generation by SBI utilizes the Distance metric, i.e., the shortest distance from one solution to a solution on the Pareto optimal front. Due to issues with determining the selected solutions in the original publication, only one sub-cluster (RCI-1) will be analyzed in this paper, c.f. [24, p. 834]. By selecting the two topmost Pareto solutions and recalculating the Distance metric against the selected solutions, decision trees can be used to generate rules for those areas. The application shows the rule $\sum(I_u + I_p) > 11$ (SumI) – in addition to the rules generated for the top-level analysis in Section 4.3 – as important to reach the selected area, in accordance with [24]. Further comparisons are difficult due to the lack of implementation details in the original article.

4.6. Export Data

After completing the analysis, the *Export Data* view can be used to export four datasets, representing the state of data from each of the previous steps, and an R script file. All the data for the first four steps will be exported as CSV-files, using semicolon as the delimiter and dot as the decimal mark. Exporting code allows the user to download an R script file with the generated code, seen in Figure 9. The main steps taken in the application, regarding changes to the dataset, are recorded in the code output. This allows for the reproduction of the result without the need for the interactive environment. For example, selecting the exact same solutions interactively can be difficult to achieve with consistency, having the exact solutions stored as code allows for exact reproducibility.

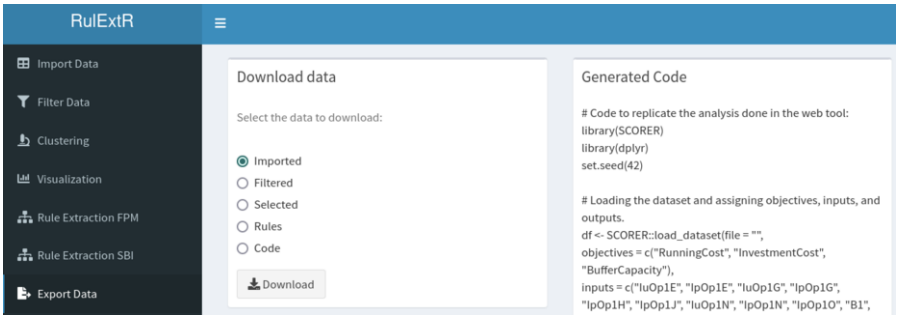

Figure 9. *Export Data* view with download options on the left and generated code on the right, cropped for the sake of brevity.

5. Conclusions and Future Work

Multi-objective optimization methods, such as NSGA-II, can iteratively optimize simulation models by modifying model inputs. The outcome of the optimization process is a large optimization 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 optimize their production processes according to the decision-makers' preference. Acquiring, modifying, analyzing, and extending 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.

The experiments presented in this study show that the knowledge extraction methods implemented in R, for both SBI and FPM, are accurate. Although the validation is difficult when only working with reported data in the papers and not exact code. Performance of the specific method can be further improved by tuning the parameters. Only CART has been used for partitioning by decision trees, other methods could also be included to improve accuracy. By offering alternative implementations of knowledge extraction methods in other languages, the methods can be more widely adopted and more easily integrated in projects. Future work will be focused on efforts to include these methods in a larger decision-support framework in R. The inclusion of Random Forests and Gradient Boosted Machines to improve rule generation of SBI is another avenue of interest, as well as evaluating the framework with industrial decision-makers on industrial problems.

References

[1] Liao Y, Deschamps F, Loures EdFR, Ramos LFP. Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal. International Journal of Production Research. 2017 jun;55(12):3609-29.

[2] Trigueiro de Sousa Junior W, Barra Montevechi JA, de Carvalho Miranda R, Teberga Campos A. Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review. Computers and Industrial Engineering. 2019 feb;128:526-40.

[3] Monostori L, Kádár B, Bauernhansl T, Kondoh S, Kumara S, Reinhart G, et al. Cyber-physical systems in manufacturing. CIRP Annals. 2016 jan;65(2):621-41.
[4] Jain S, Lechevalier D, Woo J, Shin SJ. Towards a Virtual Factory Prototype. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD, editors. Proceedings of the 2015 Winter Simulation Conference. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers Inc.; 2015. p. 2207-18.
[5] Ng AHC, Deb K, Dudas C. Simulation-based Innovization for production systems improvement : An industrial case study. In: Rosén BG, editor. Proceedings of The International 3rd Swedish Production Symposium, SPS'09, Göteborg, Sweden, 2-3 December 2009. Skövde: The Swedish Production Academy; 2009. p. 278-86.
[6] Bandaru S, Deb K. Metaheuristic techniques. Decision Sciences: Theory and Practice. 2016:693-749.
[7] Aarts AA, Anderson JE, Anderson CJ, Attridge PR, Attwood A, Axt J, et al. Estimating the reproducibility of psychological science. Science. 2015 aug;349(6251):aac4716-6.
[8] Bandaru S, Ng AHC, Deb K. Data Mining Methods for Knowledge Discovery in Multi-Objective Optimization: Part A - Survey. Expert Systems with Applications. 2017 mar;70:139-59.
[9] Bandaru S, Deb K. Higher and lower-level knowledge discovery from Pareto-optimal sets. Journal of Global Optimization. 2013 oct;57(2):281-98.
[10] Rokach L, Maimon O. Classification Trees. In: Maimon O, Rokach L, editors. Data Mining and Knowledge Discovery Handbook. Boston, MA: Springer US; 2010. p. 149-74.
[11] Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. CRC Press; 1984.
[12] Thomas P, Suhner MC, Thomas A. CART for Supply Chain Simulation Models Reduction. In: Grabot B, Vallespir B, Gomes S, Bouras A, Kiritsis D, editors. Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World. vol. 440. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014. p. 530-7.
[13] Thomas P, Suhner MC, Thomas A. Reduced simulation model for flow analysis in a sawmill internal supply chain. In: Framinan JM, Perez Gonzalez P, Artiba A, editors. 2015 International Conference on Industrial Engineering and Systems Management (IESM). IEEE; 2015. p. 1319-28.
[14] Bergmann S, Feldkamp N, Strassburger S. Emulation of control strategies through machine learning in manufacturing simulations. Journal of Simulation. 2017 feb;11(1):38-50.
[15] Prajapat N, Turner C, Tiwari A, Tiwari D, Hutabarat W. Real-time discrete event simulation: a framework for an intelligent expert system approach utilising decision trees. The International Journal of Advanced Manufacturing Technology. 2020;110(11-12):2893-911.
[16] Berk RA. Data Mining within a Regression Framework. In: Data Mining and Knowledge Discovery Handbook. Boston, MA: Springer US; 2009. p. 209-30.
[17] Breiman L. Bagging predictors. Machine Learning. 1996;24(2):123-40.
[18] Breiman L. Random forests. Machine Learning. 2001;45(1):5-32.
[19] Agrawal R, Srikant R. Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering. IEEE Comput. Soc. Press; 1995. p. 3-14.
[20] Höppner F. Association Rules. In: Data Mining and Knowledge Discovery Handbook. Boston, MA: Springer US; 2010. p. 299-319.
[21] Bandaru S, Ng AHC, Deb K. Data Mining Methods for Knowledge Discovery in Multi-Objective Optimization: Part B - New Developments and Applications. Expert Systems With Applications. 2017;70:119-38.
[22] Mittal S, Saxena DK, Deb K, Goodman ED. A Learning-based Innovized Progress Operator for Faster Convergence in Evolutionary Multi-objective Optimization. ACM Transactions on Evolutionary Learning and Optimization. 2022 3;2:1-29.
[23] Amouzgar K, Bandaru S, Andersson T, Ng AHC. A framework for simulation-based multi-objective optimization and knowledge discovery of machining process. The International Journal of Advanced Manufacturing Technology. 2018 oct;98(9-12):2469-86.
[24] Dudas C, Ng AHC, Pehrsson L, Boström H. Integration of Data Mining and Multi-Objective Optimisation for Decision Support in Production Systems Development. International Journal of Computer Integrated Manufacturing. 2014;27(9):824-39.
[25] Pehrsson L, Ng AHC, Bernedixen J. Multi-objective Production Systems Optimisation with Investment and Running Cost. In: Wang L, Ng AHC, Deb K, editors. Multi-objective Evolutionary Optimisation for Product Design and Manufacturing. London: Springer London; 2011. p. 431-53.
[26] Lidberg S. SCORER; 2022. Available from: https://github.com/verbalins/SCORER.