Knowledge Discovery in Production Simulation
By Interleaving Multi-Objective Optimization and Data Mining

Amos H.C. Ng¹, Catarina Dudas¹,², Leif Pehrsson¹,³ and Kalyanmoy Deb¹,⁴
¹Virtual Systems Research Centre, University of Skövde, Sweden
²Volvo Group Trucks Technology, Sweden
³Volvo Car Corporation, Sweden
⁴Department of Mechanical Engineering, Indian Institute of Technology Kanpur, India

KanGAL Report Number 2012019

*Received Best Paper Award in SPS12 Conference held in Linköping, Sweden, 6-8 November 2012

ABSTRACT
This paper introduces a novel methodology for the optimization, analysis and decision support in production systems development. The methodology is based on the innovization procedure, originally introduced for unveiling new and innovative design principles in engineering design problems. The innovization (innovation via optimization) procedure stretches beyond an optimization task and attempts to discover new design/operational rules/principles relating to decision variables and objectives, so that a deeper understanding of the underlying problem can be obtained. By integrating the concept of innovization with simulation and data mining techniques, a new set of powerful tools can be developed for general systems analysis. The uniqueness of the approach introduced in this paper lies in the decision rules extracted from the multi-objective optimization (MOO) using data mining (DM) are used to modify the original optimization so that faster convergence to the desired solution of the decision maker can be achieved. In other words, faster convergence and deeper knowledge of the relationships between the key decision variables and objectives can be obtained by interleaving the MOO and DM processes. In this paper, such an interleaved approach is illustrated through a set of experiments carried out to a simulation model developed in a real-world production system improvement project.

Keywords: Production System Simulation, Multi-objective Optimization, Data Mining, Innovization.

1. INTRODUCTION
Optimization involves the process of finding one or more solutions which correspond to the minimization or maximization of one or more objectives. In a single optimization problem, a single optimal solution is sought to optimize a single objective function and in a multi-objective optimization (MOO) problem the optimization involves more than one objective function. In most MOO problems, especially those found in real world, these objective functions are in conflict with each other, so that to seek a single best solution to optimize all of them simultaneously is impossible because improving one objective would deteriorate the other [1]. This scenario gives rise to a set of optimal compromised (trade-off) solutions, largely known as Pareto-optimal solutions. The so-called Pareto Front (PF) in the objective space is constituted of solutions in the Pareto-optimal solution set.

Despite the existence of multiple trade-off solutions, in most cases only one of them will be chosen as the solution for implementation, for example, in a product or system design. Therefore, two equally important tasks are usually involved in solving an MOO problem: (1) searching for the PF solution set so that the decision maker can acquire an idea on the extent and nature of the trade-off among the objectives, and (2) choosing a particular preferred solution from the Pareto-optimal set. While the first task is computationally-intensive, and can be fully automated using an MOO algorithm, the second task usually necessitates a manual decision-making process with the preference information of the decision maker. It is interesting to note that an MOO problem can be easily converted into a single-objective optimization problem by formulating a weighted-sum objective function which composed of the multiple objectives so that a single trade-off optimal solution can be sought effectively. However, the major drawback is that the trade-off solution obtained by using this
procedure is very sensitive to the relative preference vector. Therefore, the choice of the preference weights and thus the obtained trade-off solution is highly subjective to the particular decision maker. Firstly, without the detailed knowledge about the product or system under study, it is a very difficult task to select the appropriate preference vector. Secondly, converting an MOO problem into a simplistic single-objective problem puts decision making before the best possible trade-offs are known, which is undesirable in a decision-making process. In other words, thanks to the generation of multiple trade-off solutions, an MOO procedure can contribute to support the decision-making when compared to a single-objective optimization procedure. On one hand, the decision maker is provided with more “best” alternatives to consider before making the final choice. On the other hand, since these optimal (or precisely near-optimal) solutions are “high-performing” with respect to at least one objective, conducting analysis to answer “What make these solutions optimal?” can provide very important information, or knowledge, to the decision maker, which cannot be obtained if only one single solution is sought in the optimization task. The idea of deciphering knowledge, or knowledge discovery, by the post-optimality analysis of Pareto-optimal solutions from an MOO was first proposed by Deb and Srinivasan [2]. They coined the term innovization (innovation via optimization) to describe the task of discovering the salient common principles present in the Pareto-optimal solutions so that deeper knowledge/insights on the behavior/nature of the problem can be gained. The innovization task employed in earlier publications involved the manual identification of the important relationships among decision variables and objectives that are common to the obtained trade-off solutions. Recent studies using data mining techniques so that innovization procedures can be performed automatically [3,4] have been shown to be promising in various engineering problems. In these innovization tasks, the efficient evolutionary multi-objective optimization (EMO) algorithm, NSGA-II [5], has been applied to generate the Pareto-optimal solutions. Due to their population-based approach and wide-spread applicability, EMO algorithms are in general very suitable for the optimization task in an automated innovization procedure. Research in combining MOO and data mining has attracted increasing interest in the last decade. Obayashi and Sasaki [6] used Self-Organizing Maps (SOM) to cluster the design space in order to gain more information about design trade-offs in the objective space. Jeong et al. [7] applied a combination of SOM and analysis of variance (ANOVA) in the design process for aerodynamic optimizations problems. SOM was used to analyze the key design variables found in the ANOVA for further examination to get insight on how they influence the objective functions. Sugimura et al. [8] explored the use of decision trees and rough sets to analyze the optimal solutions in order to get insight on the design rules for the blower efficiency and stability of the inflow for a diffuser. In Oyama et al. [9], the dominant design features were extracted by decomposition of the shape and flow data into a set of orthogonal base vectors describing the optimal design of an aerodynamic transonic Aerofoil. In addition to data-mining techniques, data visualization techniques like 4D-plots, Parallel Coordinates, hyper-radial visualization have been used to analyze the Pareto-optimal solutions, applied to a range of automotive engineering problems [10].

It can be summarized that most of the above-mentioned related studies were focused on engineering or product design problems. As a matter of fact, by integrating the concept of innovization with simulation and data mining techniques, the innovization procedure can be applied for systems design and analysis in general. In particular, by applying MOO on discrete-event simulation models, the innovization task can be used effectively for the analysis and decision-making support in the system design/development of industrial-scale production or supply-chain systems. Such a so-called Simulation-based Innovization (SBI) procedure has been proposed in some of our previous work [11-13]. In contrast to other automated innovization procedure using data-mining techniques [3,4], the uniqueness of our proposed SBI approach lies on:

- Use decision trees/rules as the induction techniques, instead of mathematical formula, to capture the relationship of decision variables and objectives, which is believed to be easier for the decision maker to grasp and understand the extracted knowledge.

- The SBI approach emphasizes to compose the data set in data mining with both optimal and non-optimal solutions so that investigation on “What distinguish the Pareto-optimal solutions and non-optimal solutions?” can be made. This is different from other existing innovization procedures, which putting focus only on unveiling the salient common principles of the Pareto-optimal solutions.

Related to the latter point, it is logical to assume a non-optimal solution, which is closer to the PF, to possess the attributes that are closer to a Pareto-optimal solution than a solution which is far away from it. This is particularly apparent for an MOO problem with continuous decision variables and objective functions. Therefore, a distance-based approach to perform pre-processing on the data set generated from MOO has been proposed [14]. With such a distance-based pre-processing, the subsequent data mining task becomes a regression problem in which the distance of the solutions in the data set to the PF are used as the dependent continuous variable. The overall aim of this knowledge discovery in simulation procedure is therefore to decipher attributes or patterns of what distinguish a solution with small distance from a solution with a large distance from the PF, in order to portray the optimality of the PF solutions and acquire deeper knowledge of the designed system, before a final decision is made.
Very recently, the EMO literature has highlighted the importance of using a local search procedure along with an EMO procedure [15]. In that article, a procedure of serial innovization and local search approach, whereas the common principles present in EMO solutions are first deciphered and then used as heuristics in the local search operation, is proposed. The basic idea of this approach is that the relationships among the decision variables and the objectives derived from an innovization task can be used as heuristics in the local search procedure to obtain a faster convergence than a single application of EMO to the problem. We believe that this concept can be extended when considering a common decision-making scenario would require the decision maker to go through the following process iteratively, before a decision can be made: (1) run MOO to gain a rough idea of the extent of the PF; (2) select some specific interested region(s) on the PF; (3) discover the attributes of the solutions in the selected region(s); (3) perform some local search to further explore other possible solutions in the interested region(s), e.g., using reference-point based approach [16]. In other words, an advanced SBI procedure should be able to support the decision maker so that optimization, knowledge discovery and decision-making tasks can be applied in an interleaved manner, before a final knowledgeable/confident decision is made.

The aim of this paper is to introduce such an advanced SBI procedure in which MOO and data mining are interleaved. The usefulness of such a procedure to solve real-world system design problem is illustrated through a case study in an industrial production system improvement project. The rest of the paper is arranged as follows: Section 2 introduces the original SBI process and then the extended interleaved approach. Section 3 presents the industrial case study and results from earlier SBI analysis. Section 4 presents the experiments and results for the evaluation of the interleaved approach in improving the efficiency of the same optimization problem in the industrial case study. Conclusions of the paper can be found in Section 5.

2. SIMULATION-BASED INNOVIZATION

The SBI process is schematically described in Fig. 1, showing the four main steps: (1) SMO, (2) data preprocessing, (3) pattern detection, and (4) interpretation and evaluation. In essence, these four steps do not deviate from a standardized process model of an ordinary knowledge discovery in databases (KDD) process [14]. The major difference, when comparing SBI with a KDD process, is that the data used for the pattern detection is not from a data source with historical data, but from experimental data generated from SMO. Simulation is a common approach to solve industrial problems. The level of details of the simulation model can vary from a conceptual model to a model with detailed operation logics, but stochastic processes are commonly used in almost all production simulation models to model variability, e.g., due to machine failures. It is such a challenge imposed from stochastic simulation, commonly found in production system simulation that leads to the design of a novel distance based data mining approach specifically for this type of innovization applications [14].

![Fig. 1: Simulation-based Innovization Process](image-url)
decision variables are in common with the solutions in the non-dominated set. This information can be used by the decision maker to extract knowledge about the system under study. On the other hand, by analyzing the design variables together with the objective function values, the discriminating factors for separating good solutions from less good ones can also be determined. There are two issues that complicate the later analysis:

- Due to the stochastic behavior of the simulation model, the binary classification of a solution as either non-dominated or dominated is not entirely trustworthy.
- A solution closer to the PF has higher possibility to possess more similar attribute values to a non-dominated solution than a solution which is far away from this front. This is particularly true for an MOO problem with continuous decision variables and objective functions.

To address these issues, a distance-based approach to perform pre-processing on the dataset generated from SMO, which differs significantly from other innovization approaches, has been devised. Instead of treating dominated and non-dominated solutions as belonging to two different classes, and hence finding discriminatory factors by building classification models, the problem is converted into a regression problem, in which the distance to the PF is used as the dependent continuous variable in the subsequent data mining process. The task is therefore to find factors for distinguishing solutions with small distances to the PF from solutions with large distances.

For such a distance-based approach, the non-dominated solutions are used as input to the numerical interpolation of a Pareto curve/surface in the normalized objective space. Depending on the number of objectives, the interpolant will take different forms. For a two-dimensional problem it will be a curve, but for a problem with \( n \) objectives the interpolant will be a surface in the \( n \)-dimensional space. The non-dominated solutions will have a zero distance, the close-to-non-dominated solutions a rather short distance, while the solutions far away will have a long distance. The interpolant is used as the basis for the calculation of the Euclidean distance for each solution. The points of the interpolant are stored in an \( m \)-by-\( n \) matrix, \( \mathbf{Y} \), where \( n \) is the dimensionality of the problem and \( m \) is the number of points in the interpolation. The shortest distance between a point \( x \) (a solution to the optimization problem) and all possible points \( y \) on the interpolation curve/surface points in \( R_n \) is given by Eq. 1:

\[
\text{min}_i d_i = \text{min}_i \| x - y_i \| = \text{min}_i \sqrt{\sum_{l=1}^{n} |x_l - y_{il}|^2}
\]

...(1)

The distance \( d_i \) is calculated between the solution \( x \) and the interpolation point \( y_i \) and the minimization is taken over all points, for all \( i \), of the interpolation curve/surface. As an example, a two-dimensional interpolation curve for a two-objective MOO problem is illustrated in Fig. 2 in which the minimum Euclidian distances of the solutions to the curve are represented by the color scale.

![Fig. 2: Color representing the minimum Euclidian distance of solutions to the interpolation curve in a 2-objective MOO problem.](image)

2.2 Knowledge Discovery

Data mining techniques have provided powerful algorithms to automatically discover the most influencing design variables which affect the quality of solutions. Among the various data mining models, decision trees [17] are appealing for the purpose of innovization because they provide interpretable (non-opaque) predictive models. In the context of the distance-based approach, the extracted decision tree models can provide insights into how the decision variables should be configured in order to obtain a close-to-optimal solution. In other words, the purpose of using data mining in this context is to find relationships (or patterns) of the decision variables to explain why certain solutions are closer than others to the Pareto surface. It is also important to note, unlike other innovization approaches in the literature, e.g. [3], by feeding the data mining algorithms with all the solutions generated in an MOO, both Pareto and non-Pareto ones, the extracted models can also provide information about the values and relationships of the decision variables that constitute the poor solutions, which are distant from the interpolation surface.

The main purpose of SBI is to find innovative principles and to present novel information to the user. It is hence of great interest to pay attention to the visualization step of the SBI process. This step can be divided into two parts: the first part is to interpret the results from the decision tree analysis and the second step is to present the discovered information to the user in a comprehensible way. In terms of extracting important rules from the decision tree analysis, each node (internal or leaf) in the decision tree corresponds to one rule, which is represented by two parts: an antecedent
set of conditions and the consequent averaged regression value \( r_c \). The elements in the antecedent set of conditions consists of design variables \( x_1, \ldots, x_n \) and its corresponding value \( v_1, \ldots, v_n \) which is linked by an operator \( \text{op} \), as shown in Eq. 2:

\[
\text{Rule } j: \text{IF } (x_1 \text{ op } 1 v_1) \text{ AND } \ldots \text{ AND } (x_n \text{ op } n v_n) \text{ THEN } r_c = d
\]

For a rule to be “interesting”, \( r_c = d \) \( d \) is the predicted Euclidian distance) must be sufficiently small. Therefore, all nodes are checked in order to determine all the rules with \( d \) below a certain threshold of interestingness. Such a threshold for ensuring the high interestingness of the selected rules can be predetermined or determined at a later stage by the decision maker in conjunction with the visualization of the extracted rule sets.

2.3 Interleaving MOO and Knowledge Discovery

Re-define the MOO problem using the rules extracted as constraints

![Fig. 3: An interleaved MOO and DM approach.](image)

In general, since EMO algorithms do not use any mathematical optimality conditions in their operators, the obtained solutions after a finite number of computations are not guaranteed to be optimal, although an asymptotic convergence for EMOs with certain restrictions has been proven in the past [18]. To enhance the convergence properties of EMO algorithms, one common approach is to first apply an EMO and then the solutions obtained are modified by using a local search procedure one at a time. Although this hybrid procedure is commonly employed, the overall computational effort needed in executing the local search from each EMO solution can be burdensome. Therefore, an alternative strategy, which involves the hybrid use of MOO and DM, schematically illustrated in Fig. 3, is proposed in this paper. First, an MOO is run to generate a data set of sufficient size for the innovization study using DM techniques. Since the derived rules can reveal salient properties present in the MOO solutions, the obtained rules can then be used to modify the original optimization MOO problem so that a faster convergence than a single application of EMO to the original problem can be achieved.

With the introduction of the distance-based approach based on the preference region selected by the decision maker, it is deemed that faster and “better” optimization can be achieved in an interactive and interleaved manner. By interactive, it means the decision maker can select and subsequently change the preferred region by choosing different reference points. By interleaving, it implies that several iterations of the MOO-DM cycle can be repeated in order to obtain important rules with respect to the preference of the decision maker and simultaneously enhance the efficiency of the optimization by having the searching converges faster to the preferred region. Such an interleaved approach has been tested on some theoretical benchmarking functions, but in the context of this paper we are interested on the results when it is applied to simulation-based optimization studies for real-world production systems.

3. INDUSTRIAL CASE STUDY AND EARLIER SBI RESULTS

The application case study presenting here is part of an industrial improvement project conducted in an automotive manufacturer. One objective of the project was to verify and evaluate the combined use of SMO and innovization as a new and innovative manufacturing management toolset to support decision making [19]. For the SMO and the subsequent SBI methodology to be useful to solve real-world manufacturing decision-making problems, optimization of production systems via simulation models that take into account both productivity and financial factors related to the decision-making process is essential. In other words, the formulations of some financial optimization objectives, like running costs and investment costs into the SMO, was the first step of the case study.
The production line in this industrial case study suffered from high running costs due to capacity loss. It was believed in the beginning of the project that the system capacity was restricted by several constraints, which could be relieved if investments were made to improve the availability and/or reduce the processing times of some of the workstations. The purpose of the optimization study was therefore to choose the best alternative to reduce running cost with minimal investments (improvement costs) that could achieve the required capacity, in terms of system throughput, by the application of SBI. A discrete-event simulation model of the production line was developed using FACTS Analyzer and validated with accurate correspondence to the real average line throughput in the early phase of the project. The model is shown in Fig. 4 with the workstations and inter-workstation buffers labeled.

In the case study, three objectives were considered, namely, minimizing total annual running cost \( (C_r) \), minimizing total (improvement) investment cost \( (I) \), and minimizing total buffer capacities \( (B) \), as expressed with the following formulas:

\[
\min \left( C_r = C_i + \sum_{i=1}^{n} \Delta C_{r_i} + \sum_{j=1}^{n} \Delta C_{a_j} + V_p \sum_{i=1}^{n} \Delta C_{u_i} \right) \quad \ldots(3)
\]

\[
\min \left( I = \sum_{i=1}^{n} I_{a_i} + \sum_{j=1}^{n} I_{u_j} \right) \quad \ldots(4)
\]

\[
\min \left( B = \sum_{j=1}^{n} B_{c_j} \right) \quad \ldots(5)
\]

The total investment cost function \( (I) \) is simply the summation of the cost for all processing time related investment \( (I_p) \) and availability or up-time improvement \( (I_u) \). The function related to minimizing the total number of buffers in the line, \( B \), is the summation of inter-workstation buffers, \( B_i \). The estimation of the annual running cost requires detailed elaborations that beyond the scope of the current paper. Hence, we refer the readers to [19] for the complete description of the running cost modeling.

3.1 Decision variables

There are two groups of decision variables considered in the case study: the amount of investments \((I_u \text{ or } I_p)\) and the capacities for the inter-workstation buffers \((B_j)\), where \(i\) and \(j\) denote the position of the operation and buffer in the simulation model respectively (see Fig. 4). The investments were specified by different levels: zero meant no change and hence no investment to be made and their upper bounds, denoted as positive integers, were dependent on the maximum step of changes allowed by the production manager who ordered the optimization study. Since each step of change corresponded to an amount of investment, a look-up table of changes to investment cost was needed as the input data for conducting the optimization. In other words, the total amount of investment \((Eq. 4)\) was a function of the total investment steps specified by the positive values of the decision variables, i.e. \(I_u \text{ and } I_p\).

The complete step of change to investment cost table for the six workstations selected by the production manager in the case study is included in Tab.1 below.

Tab.1: Improvement steps and their associated costs

<table>
<thead>
<tr>
<th>Operation and Improvement</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Op1E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{u_{Op1E}})</td>
<td>91%</td>
<td>5.443</td>
<td>95%</td>
<td>2.500</td>
<td>96%</td>
</tr>
<tr>
<td>(I_{p_{Op1E}})</td>
<td>14%</td>
<td>10.000</td>
<td>14%</td>
<td>10.000</td>
<td>13%</td>
</tr>
<tr>
<td>Op1G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{u_{Op1G}})</td>
<td>91%</td>
<td>5.000</td>
<td>95%</td>
<td>2.500</td>
<td>95.5</td>
</tr>
<tr>
<td>(I_{p_{Op1G}})</td>
<td>145.3</td>
<td>20.000</td>
<td>138.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Op1H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{u_{Op1H}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{p_{Op1H}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Op2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{u_{Op2}})</td>
<td>15%</td>
<td>2.500</td>
<td>91%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(I_{p_{Op2}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Op3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{u_{Op3}})</td>
<td>93%</td>
<td>5.000</td>
<td>95%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(I_{p_{Op3}})</td>
<td>95%</td>
<td>20.000</td>
<td>81%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Op31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{u_{Op31}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{p_{Op31}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Op32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{u_{Op32}})</td>
<td>125%</td>
<td>20.000</td>
<td>85%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(I_{p_{Op32}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab.1 is interpreted as follows: if \( I_{u_{Op1E}} = 1 \) and \( I_{p_{Op1E}} \) denote the level of investment put to respectively improve the availability and reduce the processing time of workstation \( Op1E \) in the production line (see Fig. 4), to improve its availability by 1 step, from 91% to 95%, will cost $5,443 (\( I_{u_{Op1E}} = 1 \)) and $2,500 extra to raise 1% further to 96% (\( I_{u_{Op1E}} = 2 \)). Similarly, it costs $10,000...
(e.g. to equip new tooling) to reduce the processing time of Op1E from 145sec. to 143sec. ($i_{Op1E}=1$); it costs another $10,000 to further reduce the cycle time to 133sec. ($i_{Op1E}=2$). As a matter of fact, very often engineering/maintenance teams are good at proposing different improvements at various locations with their associated costs. Their difficulty lies on finding the optimal combinations of investment when other objectives, like in this case running cost and total buffer capacity, need to be optimized.

Excluding $i_{Op1H}$, $i_{Op1J}$ and $i_{Op1O}$, nine decision variables were used to represent the level of investments for the six workstations listed in Tab.1. Together with all the 31 inter-workstation buffers in the line, there were in total 40 discrete decision variables included in the optimization and subsequent innovization study. For the ease of interpreting the extracted rules, the same nomenclature of the decision variables will be used for the rest of the paper.

### 3.2 Results from earlier SBI study

MOO with the three objectives described earlier was performed using the built-in NSGA-II implementation in FACTS Analyzer. A set of 20,000 solutions were generated whereof only around 500 solutions are non-dominated, with respect to the three optimization objectives, in the first SBI analysis which supported the decision making in the improvement project. Although the objective space can be visualized in 3-D plots, the distribution of Pareto solutions can be shown more clearly with two 2-D plots. Because of the limited space in this paper, we opt to present the results mainly in 2D plots and analysis related to only running cost against buffer capacity, so-called RCB analysis hereafter. This was accepted in the project because the priority of the decision maker was in finding the suitable investment level to minimize the running cost and total buffer capacity so that the total investment could be treated as a secondary objective.

The results from a top-level decision tree analysis on the entire solution set are presented in Tab.2. Combined with the color-coded visualization of the extracted rules in Fig. 6, it can be easily discovered that $B_2$ and $B_8$ must have a storage capacity > 1 in order for the system to reach a Pareto optimal solution. Recall that $B_j$ is the buffer capacity for buffer $j$, and $i_{Op}$ is the investment to reduce processing time at the $i$th workstation.

**Tab.2: Results from initial decision tree analysis**

<table>
<thead>
<tr>
<th>Region</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(B_2 &amp; B_4) &gt; 1 &amp; i_{Op1H} = 0 &amp; i_{Op1O} = 0$</td>
</tr>
<tr>
<td>2</td>
<td>$(B_2 &amp; B_4) &gt; 1 &amp; i_{Op1H} = 0 &amp; i_{Op1O} = 0$</td>
</tr>
<tr>
<td>3</td>
<td>$(B_2 &amp; B_4) &gt; 1 &amp; i_{Op1H} = 0$</td>
</tr>
<tr>
<td>4</td>
<td>$B_2 &gt; 1 &amp; B_3 = 1$</td>
</tr>
<tr>
<td>5</td>
<td>$B_2 = 1$</td>
</tr>
</tbody>
</table>

With an acceptable level of investment, the decision maker was more interested in finding a solution with a very low running cost and total buffer capacity. Therefore, solutions in region one in Fig.5 represented the preference region for the decision maker to further analysis in the innovization study. It was also in this that the largest number of solutions could be found: 10,500 solutions of which 340 are non-dominated.

**Fig.5: Color-coded visualization of the 5 clusters.**

**Fig.6: Interpolation curve and colour-coded distance for two sub-clusters in the RCB analysis.**

MATLAB® functions were used for determining the interpolation curves and for calculating the minimum distances based on the formula in Eq.1. The Pareto solutions were used to determine the interpolation curve where the total number of Pareto-optimal solutions in the RCB analysis was 31.

Within region one in Fig.5, there exist sub-clusters of Pareto solutions. From a decision maker’s point of view,
it was interesting to understand not only the reason for why a solution is Pareto optimal or close-to-optimal but also to discover what are the main features that separate the Pareto solutions into these sub-clusters. Color-coded scatter plots of the Pareto solutions were used to provide information about how the Pareto space is clustered. For the RCB case, there are two sub-clusters in region 1 that fulfilled the preference of the decision maker (see Fig. 6). As can be seen in that figure, the distance of a solution to the Pareto front will depend on the selected segment of the interpolation curve. It is important to note that all distances are based on normalized objective function values so that the unit or magnitude of the objectives will not affect the distance calculations.

DM was used for generating decision trees with the 40 decision variables as the input variables and the calculated distance to the respective interpolation segments on the PF as the dependent (or regression) variable. A decision tree was generated for each segment and there are five sub-clusters in the RCB analysis. The essential part of the decision tree analysis was to find rules that can describe the characteristics of each specific sub-cluster. Nodes in the decision trees with an average distance less than the predefined threshold, \(d < 0.05\), had been selected for further analysis. Two thirds of the dataset was used for training the decision tree models and the remaining one third was used as a test set.

The results from the decision trees were the extracted rule set for each sub-cluster, where the average distance was less than the predefined threshold, 0.05, which meant that the average distance from a solution to the interpolated curve segment on the PF is 0.05 in the normalized objective space. The rule set for each cluster in the RCB analysis is given in Tab.3.

**Tab.3: Decision rules generated from the RCB analysis**

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Rule sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(\sum_l \leq 14, B_3 &gt; 26, B_1 &gt; 29, B_{15} &lt; 9, B_{13} = [7, 11], B_{15} &gt; 12)</td>
</tr>
<tr>
<td>2</td>
<td>(\sum_l \leq 19, \sum_l \geq 6, B_{15} &lt; 5, B_{15} &gt; 5)</td>
</tr>
<tr>
<td>3</td>
<td>(\sum_l \leq 8, \sum_l \geq 5, [x_{n+1} = 0, B_3 \leq 10, B_3 \leq 5, B_{13} &lt; 6, B_{27} \leq 4, B_{27} &gt; 0, B_{27} &lt; 25, B_{30} &lt; 6)</td>
</tr>
<tr>
<td>4</td>
<td>(B_3 \leq 5, B_3 &lt; 8, B_3 &lt; 4, B_{23} = 4)</td>
</tr>
<tr>
<td>5</td>
<td>(\sum_l \leq 4, \sum_l = 0, B_3 \leq 5, B_3 \leq 4, B_3 &lt; 6, B_{23} &lt; 11, B_{23} &lt; 3, B_{23} &lt; 1)</td>
</tr>
</tbody>
</table>

The rule sets found and presented in the previous section can be validated by using color-coded visualization. This means that the rule sets are applied on the dataset and the remaining solutions are plotted, with one color for each rule set in Fig. 7 for the RCB analysis.

**Fig.7: Color-coded visualization of the rule sets extracted for the RCB analysis.**

By consulting this graphical plot, the decision maker can obtain useful information based on the mapping of the generated rule sets on the solutions in the objective space. The rule set 2 (green solutions) extracted represents the best trade-off between running cost and total buffer capacity. As a matter of fact, the decision maker was targeting solutions with total buffer capacity lower than 300 and running cost lower than 3.4x10^6. Therefore, in the improvement project, the final decision was made using the knowledge represented with decision rule 2 in Tab.2. More details on how the final decision making was supported and the final implementation and validation of the industrial case study can be found in [14] and [19]. In the coming section, we will illustrate how the extended SBI process that uses the interleaved MOO and DM approach to improve the efficiency, in terms of finding “better” PF solutions with the same evaluation budget, of the same optimization problem.

**4. APPLYING THE INTERLEAVED APPROACH ON THE INDUSTRIAL PROBLEM**

A set of extended experiments illustrated in Fig.8 has been carried out to test the extended SBI approach, namely the interleaving of MOO and DM on the same simulation model developed for the earlier SBI analysis. In the figure, LHC represents Latin Hypercube design and MOO\(_{\text{ex}}\) represents the optimization using the extracted rules as the constraints, i.e. constrained optimization. The numbers represent the numbers of evaluations (one evaluation = a simulation run with \(x\) resampling) for that step. One can relate each red circle to an MOO-DM cycle represented in Fig.3. Results from a cycle are the rules as direct knowledge for the decision maker and as useful rules for re-defining the original MOO problem. While Opt.1 does not involve a new MOO run after the DM step, Opt.2 and Opt.3 use the rules from DM as inputs to the subsequent MOO\(_{\text{ex}}\) step. It can be seen that the only difference between
Opt.2 and Opt.3 is in an extra MOO step in the first MOO-DM cycle. With such an experimental design we aim at verifying how the quality of the rules from an MOO-DM cycle would affect the efficiency of the optimization. It is also important to point out that since the case study for supporting decision making in the industrial project was fully completed based on the earlier SBI analysis, these experiments are done mainly to test and verify the efficiency of the new interleaved approach.

Opt.2 and Opt.3 is in an extra MOO step in the first MOO-DM cycle. With such an experimental design we aim at verifying how the quality of the rules from an MOO-DM cycle would affect the efficiency of the optimization. It is also important to point out that since the case study for supporting decision making in the industrial project was fully completed based on the earlier SBI analysis, these experiments are done mainly to test and verify the efficiency of the new interleaved approach.

Tab. 4: Rules generated in different DM steps for the two experiments.

<table>
<thead>
<tr>
<th>Method</th>
<th>Rule set</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPT2 (LHC &amp; MOO)</td>
<td>$I_{OPT2} = I_{OPT3} = I_{OPT1} = 1, B_1 &lt; 9, B_{12} &lt; 6, B_{13} &lt; 8, B_{14} &lt; 6, B_{15} &lt; 10, B_{16} &lt; 16, B_{17} &lt; 14, B_{18} &lt; 14, B_{19} &lt; 11, B_{20} &lt; 11, B_{21} &lt; 14$</td>
</tr>
<tr>
<td>OPT2 (LHC &amp; MOO &amp; MOOc)</td>
<td>$I_{OPT2} = I_{OPT3} = I_{OPT1} = 1, B_1 &lt; 9, B_{12} &lt; 6, B_{13} &lt; 8, B_{14} &lt; 6, B_{15} &lt; 10, B_{16} &lt; 16, B_{17} &lt; 14, B_{18} &lt; 14, B_{19} &lt; 11, B_{20} &lt; 11, B_{21} &lt; 14$</td>
</tr>
<tr>
<td>OPT3 (LHC)</td>
<td>$I_{OPT2} = I_{OPT3} = I_{OPT1} = 1, B_2 &gt; 3, B_3 &gt; 3, B_4 &gt; 2$</td>
</tr>
<tr>
<td>OPT3 (LHC &amp; MOOc)</td>
<td>$I_{OPT2} = I_{OPT3} = I_{OPT1} = 1, B_2 &gt; 3, B_3 &gt; 3, B_4 &gt; 2$</td>
</tr>
</tbody>
</table>
4.1 Experimental Results

With the common reference point, [3.25x10^6, 275], chosen based on the preference of the decision maker in the earlier SBI analysis, the rules generated from different DM steps for the two experiments, Opt.2 and Opt.3 are presented in Tab. 4. The most interesting outcome from this experimental study is presented in Fig.9, showing the comparison of the quality of the optimization results. Opt.2 which with DM rules extracted from an additional step of MOO, can produce better rules that capture the attributes of the decision variables closer to the reference point. This has led to faster convergence of the subsequent MOO towards the reference point. In other words, better optimal solutions with respect to the reference point can be found more efficiently in Opt.2 than Opt.3 and Opt.1, as shown in Fig. 9.

5. CONCLUSIONS

This paper has described an extension of the SBI procedure for extracting knowledge from SMO. The SBI process is based on the post-optimality analysis of Pareto-optimal solutions, to discover knowledge, in terms of rules/principles that relate key influencing decision variables and objectives. Recent work in using data mining techniques to automate the post-optimality analysis of Pareto-optimal solutions has shown that some engineering design problems can be successfully handled. Recently, we proposed a distance-based data pre-processing approach specifically for generating high-quality rules from stochastic simulation models of real-world production systems. In this paper, this approach is further extended by the introduction of interleaving some MOO-DM cycles in order to enhance the efficiency of the optimization, in terms of faster convergence to the preference region selected by the decision maker in the objective space. Such an enhanced MOO procedure has been demonstrated with a simulation model developed previously for a full-scale industrial cost optimization case study. In our future work we will apply the methodology to address even more complex production system (e.g. assembly) as well as further comparing with other innovization approaches quantitatively.

6. ACKNOWLEDGEMENT

This work is partially financed by VINNOVA, Sweden, through the FFI-HSO project. The authors gratefully acknowledge their provision of research funding, and the industrial partners, including Volvo Car Corporation, Volvo AB and Scania, for their support in the project.

7. REFERENCES


