On the propagation of quality requirements for mechanical assemblies in industrial manufacturing*

Paul Alexandru Bucur[a,**], Philipp Armbrust[a], Philipp Hungerländer[a]

[a]*Department of Mathematics, University of Klagenfurt, Universitätsstraße 65-67, Klagenfurt 9020, Austria*

Abstract

A frequent challenge encountered by manufacturers of mechanical assemblies consists of the definition of quality criteria for the assembly lines of the subcomponents which are mounted into the final product. The rollout of Industry 4.0 standards paves the way for the usage of data-driven, intelligent approaches towards this goal. In this work, we investigate such a scenario originating in the daily operations of *thyssenkrupp Presta AG*, where new vibroacoustic quality specifications must be derived for the assembly line producing ball nut assemblies, based on the feedback offered by the vibroacoustic quality test of the steering gear, the final mechanical assembly they are mounted into. We first present a Mixed Integer Linear Programming (MILP) formulation for the problem and show that small instances of the corresponding available dataset can be solved to optimality. Upon ascertainment of the unsuitability of the MILP approach for industrial daily operations due to its long computation time, we propose a heuristic solving approach based on genetic algorithms and measure the performance gap between them in terms of achieved solution quality and computation time. Finally, we additionally propose a greedy heuristic designed to outperform the genetic algorithm in terms of computation time while still featuring a comparable solution quality. The practical relevance of the results is guaranteed, since the best solution reached by the genetic algorithm reduces the scrap costs with respect to the method currently employed by the company by 49.91%.

Keywords: Mixed integer linear programming, Genetic algorithm, Rotating

*A short paper containing a relaxation of the problem considered in this work and a genetic algorithm as a solving approach appeared in (Bucur, Frick & Hungerländer, 2019b).

**Corresponding author

Email addresses: pabucur@edu.aau.at (Paul Alexandru Bucur), philipp.armbrust@aau.at (Philipp Armbrust), philipp.hungerlaender@aau.at (Philipp Hungerländer)

multi-body system, Ball nut assemblies, Steering gears

1. Introduction

Industrial manufacturing has been undergoing a fundamental paradigm shift, away from the traditional mass production framework of the past towards a reality consisting of large product variety (ElMaraghy et al., 2013) and a shorter time to market. A critical phase in the manufacturing cycle associated with each product occurs during ramp-up, shifting the production from test batches to production at the maximum required capacity. The necessary reconfiguration of the production systems to accommodate the new products, paired with the simultaneous increase in production output, lead to potential instabilities which translate in extra costs for unplanned maintenance activities, quality and capacity loss (Colledani, Tolio & Yemane, 2018; Almgren, 1999, 2000).

This potential problem is further exacerbated in industries which produce mechanical assemblies, where the final product consists of multiple, potentially hundreds of subcomponents which interact with each other in a non-linear manner. In this production context, a fault in the final assembly may originate either due to faulty subcomponents, due to subcomponent interactions or due to a suboptimal assembly process. Furthermore, the manufacturing of the subcomponents on different machines spread across multiple countries in a global production chain often complicates the tracking and management of quality issues. Another aspect worth considering is that meeting technical specifications does not always guarantee a high perceived quality from the customers (Stylidis, Madrid, Wickman & Söderberg, 2017), a problem aggravated in domains in which quality can be subjective. Psychoacoustics, the study of sound perception by humans, constitute a relevant example in the automotive industry, with many studies trying to bridge the gap between subjective perception and measurable quantities (Huang, Huang, Li, Lim & Ding, 2016; Swart & Bekker, 2019; Kwon, Jo & Kang, 2018; Gaspar, Fontul, Henriques, Ribeiro, Silva & Valverde, 2016).

Since most manufacturer-customer relations involving mechanical assemblies are business-to-business, quality feedback from the end customers is also scarce. In the manufacturing process, the quality specifications for the production of mechanical assemblies are thus mostly defined either by, or together with the customer business partner and typically ensured by an End-Of-Line (EOL) test, paired with an analytical decision support framework (Hirsch, Reimann, Kirn & Mitschang, 2018). The resulting challenge for the mechanical assembly manufacturer is the definition of quality specifications for the subcomponent production lines and their own EOL

modules. The sheer complexity of this production scenario, paired with its inherent uncertainties, render the building of an exact mathematical model impossible. However, the consistent implementation of Industry 4.0 standards is guaranteeing an increasing volume of manufacturing data which can be used for smart manufacturing (Tao, Qi, Liu & Kusiak, 2018) and industrial prognosis tasks (Diez-Olivan, Ser, Galar & Sierra, 2019). The availability of large amounts of quality-relevant data, paired with large scale monitoring capabilities and intelligent data and sensor fusion strategies, shifts thus the focus of the practitioners towards a data-driven modeling of the complex production chains and an early identification of quality issues. (Lieber, Stolpe, Konrad, Deuse & Morik, 2013) offer such an example from the steel industry, where 218 features from 470 manufacturing processes were used as input for the prediction of a binary label describing the quality of the final product.

In the specific domain of quality control for mechanical assemblies, (Wang, Liu, Ge, Ling & Liu, 2015) propose the consideration of three research directions: assembly quality optimization, assembly performance prediction and quality control systems. Our work provides an analysis framework for the quality requirements propagation problem down the production chain of a mechanical assembly and could thus be ordered in both the quality optimization and performance prediction categories. Specifically, we assume two measurement curves from the EOL modules at the end of the production line of the mechanical assembly and of a specific subcomponent, respectively, together with an additional binary quality label describing whether the mechanical assembly was tested and found to be faulty due to the specific subcomponent or not. In our setting, the implication that a mechanical assembly fault can be traced back to a subcomponent fault can only be asserted with certainty by domain experts. The objective is to learn features from the subcomponent's EOL measurement curve which help predict the corresponding mechanical assembly's binary quality, in order to ensure that potential defects which may negatively impact the final product are recognized early in the production chain. The main challenge in our work is thus the fact that the final quality decision is achieved after the subcomponent is mounted into the mechanical assembly, while the quality screening of the subcomponent is desired before it reaches the assembly. For model learning purposes, both EOL measurement curves are available; at test time, the prediction shall be based on the subcomponent's EOL measurement curve alone.

The described scenario was previously examined by (Bucur, Hungerländer & Frick, 2019c; Bucur, Frick & Hungerländer, 2019a) via machine learning methods in the context of a specific engineering dataset from the automotive industry consisting of EOL vibroacoustic measurements encoded as order spectra for steering gears and ball nut assemblies (BNA). An alternative solving approach, still extensively used for

rotating parts in the aforementioned industry, prescribes the partition of the order spectra in different so-known quality windows, characterized by their left and right borders and their upper threshold. If the maximum of the curve in any of the windows violates the respective upper threshold, the complete curve is assigned a label of 1. This simple approach uses thus only the subcomponent's EOL curve and the quality label of the mechanical assembly as input, disregarding the mechanical assembly's EOL curve on which the quality label is assigned. Although ignoring the EOL curve of the final assembly could lead to a potentially worse quality of the subcomponent quality classification results, the relative model simplicity ensured its continued use especially in ramp-up operational phases, in which an explainable model is crucial. In the present work, we investigate this scenario thoroughly in the aforementioned ball nut assembly and steering gear context and provide a Mixed Integer Linear Programming (MILP) formulation for the problem, as well as a genetic algorithm and a greedy heuristic which achieve a high performance with respect to solution quality and computation time. To the best of our knowledge, no MILP model exists in literature for the considered problem. Our results further extend those from the previous short paper (Bucur et al., 2019b), in which a genetic algorithm was used to optimize the so-called quality windows which covered the complete order spectrum curve, implicitly assuming that each spectral order contributes information relevant for the classification. The optimization variables after this relaxation consisted of the change points on the spectral order grid and the upper threshold values between them, instead of the left and right window borders and corresponding upper thresholds.

The main contributions of this work thus expand the previously obtained results (Bucur et al., 2019b,c,a) and can be summarized as follows:

1. MILP formulation for the optimization problem seeking so-called quality windows consisting of left, right and upper borders on vibroacoustic order spectra for the BNA quality classification problem, using the steering gear quality labels as a quality indicator.

2. Redesigned genetic algorithm which overcomes multiple limitations of the design presented in (Bucur et al., 2019b) by introducing new crossover and mutation mechanisms; furthermore, it is suitable for use by practitioners due to its, with respect to the MILP approach, shorter computation time and comparable solution quality.

3. Measurement of the performance gap between the redesigned genetic algorithm and the MILP approach on subsets of the engineering dataset introduced in (Bucur et al., 2019c).

4. Proposal of a greedy heuristic which provides fast results, usable as initial solutions for the genetic algorithm or the MILP approach.

5. High practical relevance due to the achieved scrap cost reduction of 49.91% in an industrial environment via methods which provide a less complex and more interpretable alternative to the machine learning methods proposed in (Bucur et al., 2019c).

The remainder of this paper is organized as follows: in Section 2 we provide a formal description of the optimization problem seeking so-called optimal quality windows for the BNA quality classification problem, while in Section 3 we present a MILP formulation for it, as well as two other solving techniques based on genetic algorithms and greedy heuristics. In Section 4 we describe the dataset used for the validation of the different techniques, as well as their implementation details. The results and their implications are discussed in Section 5. Finally, the work is concluded in Section 6 with a review of our main achievements and suggestions for potential further research questions.

2. Problem Formulation

Let $X \in \mathbb{R}^{N \times K}$ denote the matrix containing the spectral amplitudes of N BNA order spectra sampled at K equidistant orders, with the spectral order grid defined as $S = [s_1, \ldots, s_K]$ for ordered integer indexes $1, \ldots, K$. For a specific order spectrum curve $x \in X$, we further denote its excerpt between two orders s_a (inclusive) and s_b (exclusive) by $x_{[s_a, s_b)}$. The set of binary labels $y \in \{0, 1\}^N$ offers insights into the quality of the BNA after being assembled into the steering gear: a label of 0 indicates that the corresponding steering gear passed its own quality test, implying that the assembled BNA was also qualitatively good. A label of 1 points out that acoustic domain experts traced the root cause why the steering gear failed its own quality test back to the BNA, thus suggesting its poor quality. A visualization of the BNA order spectra for a subset of the data, colored to reflect the quality of the final mechanical assembly they were mounted into, is offered in Figure 2. Further representing by Z the number of quality windows which shall be found and applied as a quality criterion to the BNA order spectra X, we further require each of the windows to feature a minimum width of $t_{\min}$ units on the spectral order grid. This can be understood as a regularization mechanism to be used against overfitting, in order to prevent the algorithms from reaching solutions which focus on single order spectrum peaks.

We introduce for $u \in \mathbb{N}$ the notation $[u] := [1, \ldots, u]$ and describe the problem solution as $\Gamma = \{(\gamma_z,\ \tau_z) | z \in [Z]\}$. This expression represents the set of pairs of the Z non-overlapping quality windows consisting of an upper threshold γ_z and corresponding interval $\tau_z = [s_{\alpha_z}, s_{\beta_z})$ on the grid S. The application of Γ as a quality criterion to X outputs a binary quality prediction vector $y^{'}$: if the spectral amplitudes in X violate the upper threshold of any quality window on the corresponding window interval, they receive a predicted label of 1 (0 otherwise). The comparison of the true labels y and the predicted labels $y^{'}$ allows the computation of the confusion matrix and thus of the number of false positives $\Gamma_{fp}(X)$ and false negatives $\Gamma_{fn}(X)$. Since the number of false positives and false negatives have a different business impact, the fitness of the solution Γ is computed by means of a weighted sum which prioritizes the identification of false positives with a positive integer factor ε:

$$\Gamma_{\text{cost}}(X) = \varepsilon \cdot \Gamma_{fp}(X) + \Gamma_{fn}(X). \tag{1}$$

3. Methodology

We propose three solving approaches for the problem described in Section 2: a Mixed Integer Linear Programming model and two heuristic approaches by means of a genetic algorithm and a greedy heuristic.

3.1. Mixed Integer Linear Programming Formulation

In addition to the non-overlapping condition imposed on the intervals on the grid S of the different quality windows, we introduce a further restriction in form of a maximum allowed interval width, which is denoted by $t_{\max}$. The observed maximum value of the spectral amplitudes in the dataset is denoted by $X_{\max} = \max_{\substack{1 \leq n \leq N, \\ 1 \leq k \leq K}} X_{n,k}$. The conditions imposed on the left and right boundaries of each quality window lead thus for $\alpha_z, \beta_z \in \mathbb{N}, \forall z \in [Z]$, to the following first set of constraints:

$$\begin{aligned} 1 \leq \alpha_z \leq K - t_{\min} + 1, \qquad t_{\min} + 1 \leq \beta_z \leq K + 1, \qquad \forall z \in [Z], \\ \alpha_z < \alpha_{z+1}, \qquad \beta_z < \beta_{z+1}, \qquad \beta_z \leq \alpha_{z+1}, \qquad \forall z \in [Z-1], \\ \alpha_z + t_{\min} \leq \beta_z, \qquad \beta_z \leq \alpha_z + t_{\max}, \qquad \forall z \in [Z]. \end{aligned}$$

For further modeling purposes, each spectral order $s_k \in S$ must be mappable to one of the quality windows (or to none of them) based on the window intervals spanned on S. To this effect we introduce the decision variables $\mu_{k,z}$ and $\lambda_{k,z}$ for all

$k \in [K]$ and $z \in [Z]$. Due to the fact that $\mu_{k,z}$ and $\lambda_{k,z}$ are both binary but cannot be 0 at the same time, we introduce the binary variables $\psi_{k,z}$ as their sum minus one, for all $k \in [K]$, $z \in [Z]$:

$$\mu_{k,z} = \begin{cases} 1, & \text{if } k \geq \alpha_z, \\ 0, & \text{otherwise.} \end{cases} \quad \forall k \in [K],\, z \in [Z], \tag{2}$$

$$\lambda_{k,z} = \begin{cases} 1, & \text{if } k < \beta_z, \\ 0, & \text{otherwise.} \end{cases} \quad \forall k \in [K],\, z \in [Z], \tag{3}$$

$$\psi_{k,z} = \mu_{k,z} + \lambda_{k,z} - 1, \quad \forall k \in [K],\, z \in [Z].$$

A spectral order $s_k \in S$ can thus be mapped uniquely to a window (γ_z, τ_z) if $\psi_{k,z} = 1$, for all $k \in [K]$, $z \in [Z]$. To describe the upper threshold set for each spectral order, we introduce the variables b_k, where $k \in [K]$. If the spectral order is mapped to a quality window, b_k shall contain the upper threshold of the window, otherwise a large enough threshold which does not have any influence:

$$b_k = \begin{cases} \gamma_z, & \text{if } \psi_{k,z} = 1, \\ X_{\max} + 1, & \text{otherwise.} \end{cases} \quad \forall k \in [K].$$

In the following, we proceed with a case-by-case analysis to determine the number of false positives and false negatives arising from the binary classification of the order spectra in X.

We first direct our focus to the false positives and set:

$$g_{n,k} = \begin{cases} 1, & \text{if } b_k - X_{n,k} \geq 0, \\ 0, & \text{otherwise.} \end{cases} \quad \forall n \in [N],\, k \in [K] \text{ where } y(n) = 1,$$

$$u_n = \begin{cases} 1, & \text{if } \sum_{k=1}^{K} g_{n,k} = K, \\ 0, & \text{otherwise.} \end{cases} \quad \forall n \in [N] \text{ where } y(n) = 1.$$

The number of false positives to be used in the cost function in Equation 1 is

then obtained by summing over u_n: $\Gamma_{fp}(X) = \sum_{\substack{n\in[N]\\ y(n)=1}} u_n$.

We then focus on the false negatives and set:

$$h_{n,k} = \begin{cases} 1, & \text{if } X_{n,k} - b_k > 0, \\ 0, & \text{otherwise.} \end{cases} \quad \forall n \in [N],\, k \in [K] \text{ where } y(n) = 0,$$

$$v_n = \begin{cases} 1, & \text{if } \sum_{k=1}^{K} h_{n,k} > 0, \\ 0, & \text{otherwise.} \end{cases} \quad \forall n \in [N] \text{ where } y(n) = 0.$$

We obtain the number of false negatives by summing over v_n: $\Gamma_{fn}(X) = \sum_{\substack{n\in[N]\\ y(n)=0}} v_n$.

Finally, we state all constraints and the objective function:

$$\begin{aligned}
\min \quad & \sum_{\substack{n\in[N]\\ y(n)=1}} \varepsilon \cdot u_n + \sum_{\substack{n\in[N]\\ y(n)=0}} v_n && && (4)\\
\text{s.t.:} \quad & 1 \le \alpha_z \le K - t_{\min} + 1, && \forall z \in [Z], && (5)\\
& t_{\min} + 1 \le \beta_z \le K + 1, && \forall z \in [Z], && (6)\\
& \alpha_z < \alpha_{z+1}, && \forall z \in [Z-1], && (7)\\
& \beta_z < \beta_{z+1}, && \forall z \in [Z-1], && (8)\\
& \beta_z \le \alpha_{z+1}, && \forall z \in [Z-1], && (9)\\
& \alpha_z + t_{\min} \le \beta_z, && \forall z \in [Z], && (10)\\
& \beta_z \le \alpha_z + t_{\max}, && \forall z \in [Z], && (11)\\
& \alpha_z - M(1 - \mu_{k,z}) \le k, && \forall k \in [K],\, z \in [Z], && (12)\\
& k < \alpha_z + M \cdot \mu_{k,z}, && \forall k \in [K],\, z \in [Z], && (13)\\
& \beta_z - M \cdot \lambda_{k,z} \le k, && \forall k \in [K],\, z \in [Z], && (14)\\
& k < \beta_z + M(1 - \lambda_{k,z}), && \forall k \in [K],\, z \in [Z], && (15)\\
& \mu_{k,z} + \lambda_{k,z} - 1 = \psi_{k,z}, && \forall k \in [K],\, z \in [Z], && (16)\\
& \gamma_z - M(1 - \psi_{k,z}) \le b_k, && \forall k \in [K],\, z \in [Z], && (17)\\
& b_k \le \gamma_z + M(1 - \psi_{k,z}), && \forall k \in [K],\, z \in [Z], && (18)\\
& X_{\max} + 1 - M \cdot \psi_{k,z} \le b_k, && \forall k \in [K],\, z \in [Z], && (19)\\
& b_k \le X_{\max} + 1 + M \cdot \psi_{k,z}, && \forall k \in [K],\, z \in [Z], && (20)\\
& X_{n,k} - M(1 - g_{n,k}) \le b_k, && \forall n \in [N],\, k \in [K],\, y(n) = 1, && (21)
\end{aligned}$$

$$b_k < X_{n,k} + M \cdot g_{n,k}, \qquad \forall n \in [N],\, k \in [K],\, y(n) = 1, \tag{22}$$

$$K \cdot u_n \leq \sum_{k=1}^{K} g_{n,k}, \qquad \forall n \in [N], \tag{23}$$

$$\sum_{k=1}^{K} g_{n,k} \leq K + u_n - 1, \qquad \forall n \in [N], \tag{24}$$

$$b_k - M \cdot (1 - h_{n,k}) < X_{n,k}, \qquad \forall n \in [N],\, k \in [K],\, y(n) = 0, \tag{25}$$

$$X_{n,k} \leq b_k + M \cdot h_{n,k}, \qquad \forall n \in [N],\, k \in [K],\, y(n) = 0, \tag{26}$$

$$v_n \leq \sum_{k=1}^{K} h_{n,k}, \qquad \forall n \in [N], \tag{27}$$

$$h_{n,k} \leq v_n, \qquad \forall n \in [N],\, k \in [K], \tag{28}$$

$$u_n,\, v_n \in \{0, 1\}, \qquad \forall n \in [N], \tag{29}$$

$$g_{n,k},\, h_{n,k} \in \{0, 1\}, \qquad \forall n \in [N],\, k \in [K], \tag{30}$$

$$\mu_{k,z},\, \lambda_{k,z},\, \psi_{k,z} \in \{0, 1\}, \qquad \forall k \in [K],\, z \in [Z], \tag{31}$$

$$\alpha_z \in [K - t_{\min} + 1] \qquad \forall z \in [Z], \tag{32}$$

$$\beta_z \in \{t_{\min} + 1, \ldots, K + 1\} \qquad \forall z \in [Z]. \tag{33}$$

The parameter M corresponds to the 'Big M' notation and is defined in Section (4.2.2). The objective function (4) minimizes the weighted sum of false positives and false negatives arising from the binary classification of the BNA vibroacoustic order spectra in the matrix X. The constraints (5) and (6) ensure that the left and right interval boundaries lie on the order grid S in a suitable manner, while the constraints (7) and (8) make certain that they also represent ordered sets with increasing values. The non-overlapping condition imposed on the quality windows is ensured by the constraints (9), while the minimum, respectively maximum interval width conditions are guaranteed by the constraints (10) and (11).

The variables $\mu_{k,z}$, $\lambda_{k,z}$ and $\psi_{k,z}$ are used to provide the mapping from each spectral order to a quality window (or to none). Their feasible values are modeled with the help of constraints (12) - (16) for all $k \in [K]$. The inequalities (12) and (13) ensure that the variables $\mu_{k,z}$ are correctly defined for all $k \in [K]$, given the if-condition (2). Furthermore, the constraints (14) and (15) are used to set $\lambda_{k,z}$ to the values defined in the if-condition (3) for all $k \in [K]$. Finally, the equations (16) ensure that $\psi_{k,z}$ is set to 1 if the spectral order s_k lies within the quality window z and 0 otherwise.

In order to determine the values of the variables b_k describing the upper threshold set for each spectral order we need for all $k \in [K]$ the constraints (17) - (20). The inequalities (17) and (18) set γ_z as threshold value if $\psi_{k,z}$ is equal to 1. If $\psi_{k,z}$ is equal to 0, then the upper threshold value b_k at the spectral order s_k has the value $X_{\max} + 1$, which is ensured by the inequalities (19) and (20).

By focusing on the false positives part of the objective function, we consider all

order spectra $x \in X$ with true label 1. If $y(n) = 1$, the constraints (21) and (22) guarantee that the following holds for the binary variables $g_{n,k}$: if $X_{n,k}$ does not exceed the upper threshold set for the respective s_k, then the value of $g_{n,k}$ is 1 and 0 otherwise. If for a fixed $n \in [N]$ and all $k \in [K]$ the variables $g_{n,k}$ are equal to 1, then no violation of the upper thresholds occurs and the predicted label of the order spectrum is 0. The corresponding feasible values for the variables u_n are modeled with the constraints (23) and (24).

Switching our focus to the false negatives, we consider all order spectra $x \in X$ with true label 0. If $y(n) = 0$, the feasible values of the binary variables $h_{n,k}$, which indicate whether a violation of the upper thresholds occurs, are modeled with the inequalities (25) and (26) for all $k \in [K]$, $n \in [N]$. As reflected by the constraints (27) and (28), a single violation of the threshold of any quality window suffices for an order spectrum to receive a predicted label of 1.

Finally, in constraints (29), (30) and (31) all binary variables are introduced as such. The constraints (32) and (33) state the fact that the left window borders α_z and right window borders β_z are positive integers in their defined range.

3.2. Greedy Heuristic

The proposed greedy heuristic consists of two main phases: the *computation phase* processes the information in the matrix of spectral amplitudes X to construct a decision basis, which is used then in the *decision phase* to compute the cost associated with each quality window option and iteratively select the next best window choice. A pseudocode version of the algorithm is offered in Algorithm 2, which we proceed to describe in detail.

The algorithm input consists of the spectral amplitude matrix X, the associated quality labels $y \in \{0, 1\}^N$, the minimum width $t_{\min}$ of a quality window on the spectral order grid and the required number Z of quality windows which the algorithm should find. After its computations, the algorithm returns the solution as the set of quality windows $\Gamma = \{(\gamma_z, \tau_z) | z \in [Z]\}$. A noteworthy property of the proposed algorithm is that it returns a lower number than Z of quality windows if additional ones would not contribute to a further cost reduction, avoiding thus redundancies or irrelevant windows.

Algorithm 2 The greedy heuristic designed to iteratively find quality windows for the BNA vibroacoustic quality classification problem which was formulated in Section 2. The two main phases of the algorithm, the *computation* and *decision* phases, are extensively discussed in Section 3.2.

procedure RUN (X, y, $t_{\min}$, Z)
    $C^{\dagger} \leftarrow$ Computation phase $(X, y, t_{\min})$
    $\Gamma^{*} \leftarrow$ Decision phase $(C^{\dagger}, X, y, Z)$
end procedure

function COMPUTATION PHASE (X, y, $t_{\min}$)
    $C^{\dagger} \leftarrow \emptyset$
    for $i \leftarrow 1, K - t_{\min} + 1$ **do**
        $\tau \leftarrow [s_i, s_{i+t_{\min}})$
        $C \leftarrow \{\max(x_\tau) | x \in X\}$
        $\eta_\tau \leftarrow \min\{x_\tau^{'} | x_\tau^{'} \in C \wedge y(x) = 1\}$
        $C^{\dagger} \leftarrow C^{\dagger} \cup \{x_\tau^{'} | x_\tau^{'} \in C \wedge x_\tau^{'} \geq \eta_\tau\}$
    end for
    return $C^{\dagger}$
end function

function DECISION PHASE ($C^{\dagger}$, X, y, Z)
    $f \leftarrow 0$
    $\Gamma^{*} \leftarrow \emptyset$
    repeat
        γ_f^{*}, $\tau_f^{*} \leftarrow$ MINIMALCOST $(C^{\dagger}, y, \Gamma^{*}, X)$
        $f \leftarrow f + 1$
        extend interval τ_f^{*}
        $\Gamma^{*} \leftarrow \Gamma^{*} \cup \{(\gamma_f^{*}, \tau_f^{*})\}$
        $C^{\dagger} \leftarrow \{x_\tau^{'} | x_\tau^{'} \in C^{\dagger} \wedge \tau \cap \tau_f^{*} = \emptyset\}$
        $C^{\dagger} \leftarrow \{x_\tau^{'} | x_\tau^{'} \in C^{\dagger} \wedge \Gamma^{*}(x) = 0\}$
        if $C^{\dagger} = \emptyset$ **then**
            break
        end if
    until $f = Z$
    return Γ^{*}
end function

```
function MINIMALCOST (C†, y, Γ*, X)
    γ* ← ∞
    τ* ← [s_1, s_{K+1})
    Γ = ∅
    if Γ* = ∅ then
        Γ ← {(γ*, τ*)}
    else
        Γ ← Γ*
    end if
    X' ← {x|x ∈ X ∧ Γ(x) = 0}
    mincost ← ε · Γ_fp(X')
    for j ← 1, K − t_min + 1 do
        τ_j ← [s_j, s_{j+t_min})
        D ← {x'_τ|x'_τ ∈ C† ∧ τ = τ_j}
        D_1 ← {x'_τ|x'_τ ∈ D ∧ y(x) = 1}
        for all x'_τ ∈ D_1 do
            γ = x'_τ
            Γ ← {(γ, τ_j)}
            X'' ← {x|x ∈ X' ∧ x ∈ D}
            cost ← ε · Γ_fp(X'') + Γ_fn(X'')
            if cost < mincost then
                mincost ← cost
                γ* ← γ
                τ* ← τ_j
            end if
        end for
    end for
    return γ*, τ*
end function
```

In the *computation phase*, we initially focus on the first viable interval spanning $t_{\min}$ units on the spectral order grid. Since the deciding factor for the quality outcome is whether the maximum amplitude value between the left and right window boundaries violates the upper threshold, we first compute the maximum amplitude of each order curve on this interval and denote it by x'_τ. In order to narrow down the search space, we only contemplate the maximum amplitudes x'_τ of the curves with label 1 as potential upper thresholds for this interval: a threshold below the lowest

$x_\tau^{'}$ with label 1 would only potentially generate further false negative predictions and thus increase costs without any additional benefit. Bearing in mind that only curves with maxima above the lowest $x_\tau^{'}$ with label 1 could potentially lead to false positives and false negatives, we keep record solely of the maxima of these curves over the interval. The described process is repeated for multiple consecutive intervals on the spectral order grid, with the choice of the overlap between intervals influencing the size of the search space and solution quality; in the following, we denote by θ the parameter describing the hop length on the spectral grid between two consecutive intervals.

In the *decision phase*, we repeat the search procedure for the next optimized quality window until either Z windows are found or additional windows would not help further reduce costs. The search procedure operates in a rather simple manner: it loops over all intervals recorded in the *computation phase* and within them over all maximum values $x_\tau^{'}$ of the distinct spectra with label 1, considering each one as a potential upper threshold on its interval. For each upper threshold possibility, the search procedure then computes the number of false positives and false negatives arising from the application of the threshold, considering only curves with maxima on the interval above the lowest $x_\tau^{'}$ with label 1. By computing the associated cost from the number of false positives and false negatives, it is possible to identify the interval and threshold which most lower the costs arising from yet undetected order spectra with label 1. In further iterations, the search procedure does not consider anymore intervals overlapping with the previously selected ones. Due to the fact that a violation of a single window is already sufficient for a negative quality assessment, each order spectrum already classified as true or false negative by the selected quality window is then removed from cost computations in further iterations.

Despite its major strength in terms of simplicity, the proposed heuristic has a drawback: it only searches for windows featuring the minimum allowed length $t_{\min}$. Although this limits the search space, it also potentially limits the solution quality. In order to alleviate this problem, we propose a simple procedure which runs after the next most promising window is selected and aims to extend the interval corresponding to the quality window, provided that this further reduces costs. To this effect, the left and right interval boundaries are gradually extended by a single unit on the spectral order grid: each such extension is only accepted if it contributes to a further cost reduction and does not lead to overlap, otherwise the extension procedure is interrupted.

3.3. Genetic Algorithm

In a previous contribution (Bucur et al., 2019b), a genetic algorithm was proposed as a solving approach for the problem described in Section 2. Despite having been able to reduce the scrap costs by 24.16% on the corresponding test data (the used dataset is not identical to the one used in this work), it was subject to multiple limitations:

1. The mathematical model of the solution as a piecewise constant function with multiple change points and covering the complete spectral order grid S implicitly assumed that each order contributes information relevant for the classification and should thus be subject to a threshold.

2. The Nelder-Mead downhill simplex method, a gradient-free maximization technique, was employed in the *update* step to optimize the upper threshold values while keeping the change points constant. However, the gains in terms of increased solution quality after this relatively computationally expensive technique were significantly diminished after a random mutation or after the averaging crossover mechanism, which acted as a serious perturbation.

3. The crossover mechanism ran an agglomerative clustering algorithm on the merged change points of the parent functions, with the change points of the offspring individual then chosen as the centroids of the clusters resulting from the cut dendrogram. The upper threshold values between the new change points were obtained by building the average over the upper thresholds of the parent functions on the respective intervals. If the parent solutions however focused on different order intervals responsible for different root faults leading to a label of 1 and thus faulty parts, the crossover technique is unlikely to find itself a root fault by clustering and averaging.

In the present work, we overcome these shortcomings by imposing upper thresholds only on select areas of the order spectrum and by renouncing to the use of the computationally heavy Nelder-Mead method. Furthermore, we also introduce a more efficient crossover mechanism which selects the best windows from both parents based on their achieved cost reduction. Instead of introducing variation through an averaging of the upper thresholds in the crossover mechanism, the current approach uses mutations which are only accepted if they help reduce cost and whose mutation strength is diminished linearly with each further generation, finally reaching a value of 0 for the last generation.

In a first step, the population individuals are initialized in a random manner. To this effect and for all $z \in [Z]$, the α_z values representing the left window borders

are first sampled from the uniform distribution over the spectral order grid in such a manner that after ordering them, the difference between two consecutive α_z values is greater than or equal to the minimum allowed window length $t_{\min}$. The right window borders denoted by β_z are chosen randomly for all $z \in [Z-1]$ between consecutive α_z values, again respecting the minimum window length $t_{\min}$. Last, the initial upper threshold γ_z of each window $z \in [Z]$ is set to the maximum amplitude value of all order spectra with label 0 plus a random variation of maximum $\pm 3\%$.

In the *evolution* phase of each generation a certain proportion of the best population individuals is selected, while the remaining ones are discarded. Upon addition of multiple randomly generated new individuals, the population is refilled by choosing pairs of individuals at random with replacement and crossing them over to generate new individuals. Subsequently, a random selection of the population is subject to the mutation mechanism. In the following we explain the crossover and mutation methodology.

The *crossover* mechanism follows a simple cost minimization approach: first, it selects two parents at random with replacement from the population. For each window of each parent, it computes the confusion matrix resulting from the application of the window's upper threshold to the amplitudes of the order spectra and the associated cost resulting from the number of false positives and false negatives. In an iterative procedure repeated until Z windows are found, it then selects the next most promising window which offers the highest further cost reduction. After a window is selected, other overlapping windows are disregarded. Furthermore, the confusion matrices of all other window possibilities are updated, removing from the cost calculation the order spectra already violating previous windows. The reason behind this approach is twofold: first, the best non-overlapping windows from either parent are selected. Second, the negative cost impact of a window, measured in terms of generated false positive and false negative curves, is diminished if the false negatives were already misclassified by another window; as described in Section 2, the cost for a curve is already maximum if it violates any window.

In the *mutation* step, the α_z, β_z and γ_z values of a random window $z \in [Z]$ of an individual are randomly shifted with a certain probability. The strength of the shifts diminishes linearly with each generation from its initial value to a final value of 0, in order to ensure that significant changes to the individuals are made rather early in the optimization process. Furthermore, the mutation result is only accepted if it does not increase the cost associated with the individual and does not lead to overlapping windows. Since the mutation mechanism cannot worsen the quality of an individual, the algorithm tries to mutate each window of the best individual in a generation.

In order to avoid overfitting during the training process we also employ the technique known as Early-Stopping (Goodfellow, Bengio, Courville & Bengio, 2016). After each generation, the cost of the best individual in the population is evaluated on validation data, chosen as a fixed, stratified and random subset of the training data. If the cost on the validation data begins to increase again and continues to do so after a number of generations marked as the patience parameter, the training process is stopped.

4. Computational Experiments

The computational infrastructure on which the benchmarking tests were conducted consists of a server running the *Windows Server* 2016 operating system and featuring 256 GB RAM and two Intel Xeon E5 − 2643− v4@ 3.40 GHz processors. The proposed methodology was validated on the same dataset used for the validation of the machine learning-based methods in (Bucur et al., 2019c), which is different than the one employed in (Bucur et al., 2019b).

4.1. Dataset

4.1.1. Full Instance

The dataset employed for the validation of the genetic algorithm and of the greedy heuristic was split into a training and a test dataset, consisting of 9291 and 3966 BNA order spectra, respectively. The number of BNA to which the steering gear faults could be traced back to and which thus, as described in Section 2, receive a label of 1 are 410 and 160, respectively, indicating a faulty percentage of 4.41% in the training and of 4.03% in the test dataset. Unlike in our previous contribution (Bucur et al., 2019c), we refrain from rebalancing the training dataset. Instead, we ensure as described in Section 3 that the initial solutions are already in a feasible area of the search space and compute at each training iteration the induced cost on the complete training data, instead of batch-wise.

4.1.2. Mixed Integer Linear Programming Data Subsets

First computational experiments showed that solving the MILP model requires a long time for calculations: in Table 1 we can observe that solving the problem to optimality for only 3 quality windows and a random subset of the data consisting of 15 order spectra from class 0 and 15 from class 1 required a total of 48.89 hours. Due to the prohibitive computation time length, we renounced to obtain an exact solution for the complete training data and focused instead on gaining insights on the performance gap in terms of solution quality and computation time between the

MILP approach and the genetic algorithm on handpicked subsets of the training data.

To this effect, 12 small datasets were created, with the order spectra belonging to them handpicked from the training data to ensure that the problem could be solved to optimality. The different small datasets contain between 5 and 15 order spectra from each class and are always balanced (they feature the same number of samples from each class). The exact number of order spectra composing each small dataset, together with the number of computed quality windows and the resulting cost on the same dataset are protocoled in Table 1.

4.2. Implementation Details

The implementation of the proposed methods is straightforward, with a manageable number of parameter choices. The most important parameter is ε: common to all methods, it denotes the positive integer factor which prioritizes the recognition of false positives in detriment of false negatives in the cost function. In the present work, this parameter was offered as input by *thyssenkrupp Presta AG* and set to $\varepsilon = 10$. The greedy heuristic presented in Section 3.2 only takes three parameters as input: θ as the hop length parameter, $t_{\min}$ as the required minimum window width and Z as the number of windows which it shall find. In order to provide a thorough benchmark, we protocol the results of the greedy heuristic in terms of solution quality and computation time for all feasible parameter combinations with $\theta \in \{1, 5, 10, 20\}$, $t_{\min} \in \{5, 10, 20, 30, 40, 50\}$ and $Z \in [15]$. In the following, we describe the implementation of the genetic algorithm and the MILP formulation.

4.2.1. Genetic Algorithm

In order to provide a direct comparison to the quality of the greedy heuristic solutions, the results of the genetic algorithm in terms of solution quality and computation time were computed for the same range of values for the $t_{\min}$ and Z parameter: $t_{\min} \in \{5, 10, 20, 30, 40, 50\}$ and $Z \in [15]$. The rest of the parameter required for the implementation were chosen by means of an offline parameter tuning approach. To this effect, a subset of the training data was selected, consisting of all the samples from the minority class and an equal number of samples from the majority class, obtained by means of fast, random undersampling. The best parameter choices were then determined via a 5-fold cross-validated grid-search over a specific parameter grid. The number of generations was optimized to a value of 300, with each generation featuring a total of 100 individuals. The percentage of fittest individuals preserved during the evolution phase is 30%, to which a total of 20% of the population size randomly generated new individuals are added. Finally, the mutation

probability was optimized to an initial value of 0.8 and decreased linearly with each generation, until reaching a final value of 0.

4.2.2. Mixed Integer Linear Programming Formulation

We solve the dataset instances described in Section 4.1.2 with the MILP formulation by making use of the open source software "Google OR-Tools" (Google OR-Tools v7.0) and the Coin-or Branch and Cut solver (Forrest et al., 2018). The minimum window width was set to $t_{\min} = 10$, while the maximum allowed width was chosen as $t_{\max} = 30$. Furthermore, the parameter M corresponding to the 'Big M' notation was set to $5 \cdot \lceil X_{\max} \rceil$.

5. Results

In the following, we first present the results of the MILP approach on the small data subsets described in Section 4.1.2 and compare them to the results obtained by the genetic algorithm on the same data. Despite the limited significance of the tests, constrained by the very long computation time of the MILP approach and thereby by the small number of samples in the datasets, we observe that the genetic algorithm reaches the same solution quality as the MILP approach in terms of cost, computed as described in Equation 1.

Second, we benchmark the genetic algorithm against the proposed greedy heuristic, establishing the superiority of the former with respect to achievable solution quality. This benchmarking study is performed for a different number of quality windows, each with multiple minimum window length possibilities. The missing entries in the result tables are due to the fact that the corresponding combinations of number of quality windows and minimum window length would lead to overlapping windows in violation of the feasible solution requirements, as stated in Section 2.

5.1. Mixed Integer Linear Programming Formulation

The benchmarking results for the MILP approach and the genetic algorithm on the training data subsets described in Section 4.1.2 are presented in Table 1. The extent of the benchmarking tests is constrained by the long computation time of the MILP approach and thus limited to 12 data subsets, solvable to optimality for either 3 or 5 quality windows. In all 12 data scenarios, the genetic algorithm was able to match the solution quality of the MILP approach in terms of incurred cost in a fraction of the time. The parameters used for the genetic algorithm implementation in each data scenario were tuned in an offline manner, with their values protocoled aside the results.

	Number of spectra	Time [s]		GA parameters					
		MILP	GA	a	b	c	d	e	f
Optimal value: 0 Number of windows: 3	10	302.43	14.53	0.4	0.1	0.5	0.7	100	100
	16	31274.52	17.26	0.4	0.1	0.5	0.7	100	100
	20	5397.47	16.05	0.2	0.5	0.3	0.8	100	100
	24	5876.58	17.08	0.4	0.2	0.4	0.7	100	100
Optimal value: 1 Number of windows: 3	10	5405.22	46.76	0.3	0.3	0.4	0.7	200	100
	16	65968.40	14.31	0.4	0.3	0.3	0.5	100	100
	18	66945.91	120.86	0.3	0.2	0.5	0.4	500	200
	30	175994.50	57.62	0.3	0.3	0.4	0.7	200	200
Optimal value: 0 Number of windows: 5	10	207.54	16.58	0.4	0.1	0.5	0.7	100	100
	12	56644.21	58.83	0.3	0.2	0.5	0.7	200	200
	16	37592.43	1463.14	0.4	0.3	0.3	0.8	1000	1000
Optimal value: 1 Number of windows: 5	20	57721.81	90.02	0.4	0.3	0.3	0.7	300	200

Table 1: Comparison of the results of the MILP approach and of the genetic algorithm (GA) in terms of solution quality and computation time on the 12 distinct datasets described in Section 4.1.2 using either 3 or 5 quality windows. It is noteworthy that in all cases, the genetic algorithm determined the optimal solution in a fraction of the time. The protocoled total number of order spectra for each data scenario contains the same number of data samples from each class. The parameters used for the implementation of the genetic algorithm for each small data subset are protocoled aside the results and have the following meaning: a) proportion of best individuals in the population kept during the evolution phase b) proportion of the population which is refilled with random individuals during the evolution phase c) proportion of the population which is refilled with crossed over individuals during the evolution phase d) mutation probability e) number of generations f) number of individuals in the population.

5.2. Greedy Heuristic

The results of the greedy heuristic with respect to achieved cost on the test data and time needed for the computations on the training data are protocoled for different values of the hop parameter θ in multiple tables: Table 2 for $\theta = 1$, Table 3 for $\theta = 5$, Table 4 for $\theta = 10$ and Table 5 for $\theta = 20$.

It is noteworthy that due to the design of the greedy heuristic and without the window boundary extension mechanism in Algorithm 2, line 21, each solution with a specific hop length θ_i would be guaranteed to be at least as good as solutions with hop lengths θ_j which are multiples of θ_i, since the solution point in the high-dimensional search space of the latter is also guaranteed to have been explored in the search with the lower θ_i value. This would imply that the best results in terms of solution quality achieved by the greedy heuristic correspond to the value of $\theta = 1$, which can be encountered in Table 2. However, the usage of the window boundary extension mechanism after the iterative selection of the next best quality window can

Number of windows	Minimum window length											
	5		10		20		30		40		50	
	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]
1	1466	232.93	1369	224.27	1298	214.48	1315	209.95	1383	201.65	1280	191.20
2	1422	269.31	1330	258.48	1218	240.60	1276	233.06	1362	222.83	1236	207.30
3	1387	296.86	1336	281.48	1200	259.59	1250	250.25	1380	235.26	1265	212.16
4	1300	315.87	1246	300.50	1192	273.78	1227	260.42	1302	238.69	1265	212.05
5	1231	333.09	1169	316.61	1201	283.05	1180	264.95	1303	240.31	1265	212.25
6	1248	349.14	1172	329.99	1202	290.31	1185	268.47	1303	239.86		
7	1238	362.20	1176	341.76	1221	296.22	1185	268.64				
8	1241	376.30	1193	352.17	1221	300.75	1185	269.07				
9	1242	388.71	1200	362.15	1221	301.66						
10	1242	401.70	1201	370.23	1221	301.97						
11	1250	412.86	1201	378.39	1221	301.68						
12	1250	423.58	1205	385.59	1221	301.45						
13	1247	434.34	1206	392.10								
14	1250	443.19	1206	394.72								
15	1250	449.69	1206	394.69								

Table 2: Results of the greedy heuristic described in Section 3.2, implemented with a hop length value of $\theta = 1$. The results reflect the attainable solution qualities in terms of incurred cost on the test data and computation time on the training data for a different number of quality windows and multiple minimum window lengths.

Number of windows	Minimum window length											
	5		10		20		30		40		50	
	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]
1	1443	48.38	1443	46.50	1406	43.63	1324	43.02	1396	41.21	1280	39.55
2	1355	57.41	1364	54.33	1376	50.38	1284	49.21	1369	46.72	1236	44.75
3	1310	65.93	1324	60.92	1336	55.94	1257	54.37	1386	52.13	1264	46.66
4	1218	73.67	1228	68.20	1320	60.94	1233	58.52	1368	54.79	1264	46.63
5	1190	80.08	1200	74.08	1320	65.43	1186	61.65	1368	54.83	1264	46.58
6	1200	85.98	1202	79.12	1320	69.46	1186	61.75	1368	54.80		
7	1186	91.95	1207	82.83	1320	70.58	1186	61.92				
8	1187	97.60	1219	87.59	1320	70.44	1186	61.78				
9	1195	103.24	1225	92.31	1320	70.35						
10	1195	108.97	1225	96.87	1320	70.58						
11	1195	114.37	1226	101.53	1320	70.54						
12	1195	119.83	1226	102.22	1320	70.45						
13	1195	121.82	1226	102.38								
14	1195	121.82	1226	102.36								
15	1195	121.70	1226	102.53								

Table 3: Results of the greedy heuristic described in Section 3.2, implemented with a hop length value of $\theta = 5$. The results reflect the attainable solution qualities in terms of incurred cost on the test data and computation time on the training data for a different number of quality windows and multiple minimum window lengths.

lead to initially counterintuitive, yet favorable results. Even though it is guaranteed that the *computation* phase of the greedy heuristic has also explored for $\theta = 1$ all window possibilities achievable with the other hop lengths and the *decision* phase has selected the most promising among them, it is possible that a suboptimal selection (in the case of higher hop lengths, with respect to a hop length of 1) proves itself to

Number of windows	Minimum window length											
	5		10		20		30		40		50	
	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]
1	1443	24.55	1443	24.35	1406	22.42	1420	22.32	1392	21.29	1280	20.30
2	1355	30.56	1370	30.09	1376	26.94	1395	26.27	1392	25.51	1231	23.34
3	1290	36.49	1387	34.73	1350	31.07	1364	30.16	1411	28.85	1259	25.01
4	1295	41.09	1339	40.97	1334	35.13	1340	33.49	1411	31.48	1260	26.13
5	1279	45.68	1342	45.59	1334	39.04	1345	36.56	1393	33.96	1260	26.09
6	1267	50.46	1327	50.17	1334	39.75	1345	36.68	1393	33.92		
7	1268	54.51	1338	54.50	1334	39.74	1345	36.69				
8	1268	58.69	1339	58.95	1334	39.68	1345	36.66				
9	1268	60.39	1345	63.64	1334	39.77						
10	1268	60.25	1346	65.87	1334	39.60						
11	1268	60.29	1346	70.22	1334	39.65						
12	1268	60.17	1346	70.70	1334	39.64						
13	1268	60.22	1346	70.87								
14	1268	60.23	1346	70.94								
15	1268	60.27	1346	70.74								

Table 4: Results of the greedy heuristic described in Section 3.2, implemented with a hop length value of $\theta = 10$. The results reflect the attainable solution qualities in terms of incurred cost on the test data and computation time on the training data for a different number of quality windows and multiple minimum window lengths.

Number of windows	Minimum window length											
	5		10		20		30		40		50	
	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]
1	1684	12.18	1408	12.75	1428	11.91	1420	11.46	1276	11.46	1280	10.20
2	1648	15.74	1420	16.88	1406	14.92	1395	14.34	1228	13.96	1222	12.71
3	1667	19.43	1355	20.67	1391	17.95	1371	17.60	1252	15.47	1224	14.98
4	1683	22.91	1357	24.40	1357	21.30	1372	20.82	1252	17.81	1225	15.97
5	1598	26.37	1358	28.17	1375	22.94	1372	20.75	1258	19.66	1225	16.06
6	1560	29.64	1354	31.78	1375	25.96	1372	20.86	1258	19.81		
7	1561	32.97	1357	35.36	1394	28.72	1372	20.71				
8	1561	33.70	1357	35.99	1395	31.61	1372	20.86				
9	1561	33.62	1357	35.89	1395	32.04						
10	1561	33.58	1357	35.98	1395	31.94						
11	1561	33.68	1357	35.97	1395	31.93						
12	1561	33.87	1357	35.85	1395	32.01						
13	1561	33.67	1357	35.82								
14	1561	33.68	1357	35.87								
15	1561	33.52	1357	35.88								

Table 5: Results of the greedy heuristic described in Section 3.2, implemented with a hop length value of $\theta = 20$. The results reflect the attainable solution qualities in terms of incurred cost on the test data and computation time on the training data for a different number of quality windows and multiple minimum window lengths.

be better *after* the window boundary extension mechanism, leading to better results.

5.3. Genetic Algorithm

The results of the genetic algorithm, parametrized as described in Section 4.2.1, with respect to achieved cost on the test data and computation time on the training data are recorded in Table 6. Note that the results represent the average values over

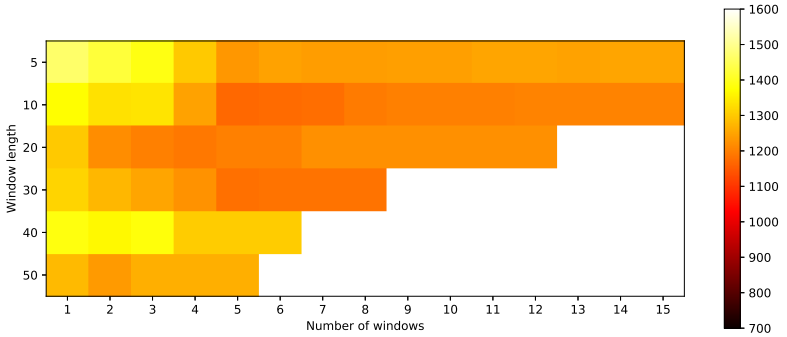
Number of windows	Minimum window length											
	5		10		20		30		40		50	
	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]	Cost	Time [s]
1	1388.67	138.41	1235.17	81.64	1230.33	104.29	1229.17	95.07	1266.50	84.33	1222.83	83.06
2	1095.83	166.98	1089.67	140.43	1099.50	212.09	1091.83	176.93	1091.83	158.96	1102.17	152.93
3	1058.67	247.61	1074.67	220.12	1075.17	362.42	1088.50	281.47	1092.33	258.08	1096.17	260.85
4	1011.50	295.41	1021.33	287.71	1031.00	511.69	1020.83	325.29	1050.83	327.59	1113.17	331.27
5	997.83	357.72	1037.50	347.34	1030.67	623.63	1057.00	353.56	1108.83	397.18	1090.33	329.29
6	956.17	436.35	988.83	424.36	1011.83	868.24	1056.17	497.46	1059.83	451.06		
7	947.00	524.56	948.17	506.52	1034.67	839.72	1007.33	535.40				
8	939.33	607.68	946.00	570.59	1019.17	808.69	1057.00	524.85				
9	895.50	721.20	928.50	796.97	984.17	936.60						
10	844.17	795.09	886.67	1254.24	1018.00	1106.82						
11	869.17	859.47	887.67	1587.75	1003.50	1030.41						
12	849.50	924.95	938.17	1454.65	1009.50	1091.94						
13	857.67	1014.22	900.17	1445.38								
14	805.17	1092.32	944.33	1993.66								
15	801.50	1184.60	879.33	1574.93								

Table 6: Results of the genetic algorithm described in Section 3.3 and parametrized as described in Section 4.2.1. For each combination of required number of quality windows and minimum window length, the results give insights into the obtained solution quality in terms of incurred cost on the test data and computation time on the training data. The best solution features a cost of 801.50, which implies a scrap cost reduction of 49.91% with respect to the currently employed BNA quality classification method.

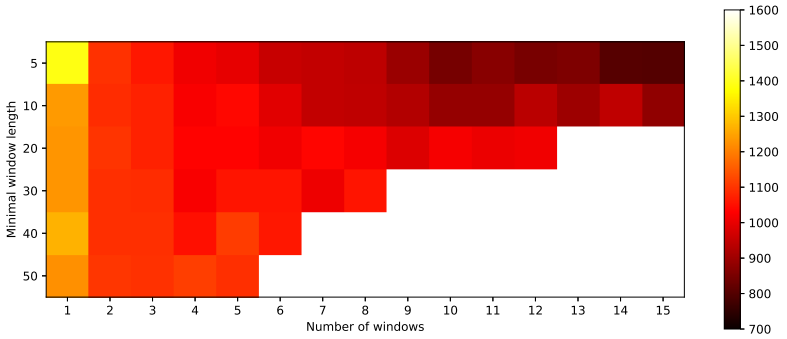
10 algorithm runs with different initial random seeds. A comparison of the incurred costs on the test data by the genetic algorithm and the greedy heuristic using the smallest hop length of $\theta = 1$ is offered in Table 7, while Figure 1 offers further insights by means of a visual comparison of the results.

Number of windows	Minimum window length											
	5		10		20		30		40		50	
	Genetic	Greedy	Genetic	Greedy	Genetic	Greedy	Genetic	Greedy	Genetic	Greedy	Genetic	Greedy
1	1388.67	1466	1235.17	1369	1230.33	1298	1229.17	1315	1266.50	1383	1222.83	1280
2	1095.83	1422	1089.67	1330	1099.50	1218	1091.83	1276	1091.83	1362	1102.17	1236
3	1058.67	1387	1074.67	1336	1075.17	1200	1088.50	1250	1092.33	1380	1096.17	1265
4	1011.50	1300	1021.33	1246	1031.00	1192	1020.83	1227	1050.83	1302	1113.17	1265
5	997.83	1231	1037.50	1169	1030.67	1201	1057.00	1180	1108.83	1303	1090.33	1265
6	956.17	1248	988.83	1172	1011.83	1202	1056.17	1185	1059.83	1303		
7	947.00	1238	948.17	1176	1034.67	1221	1007.33	1185				
8	939.33	1241	946.00	1193	1019.17	1221	1057.00	1185				
9	895.50	1242	928.50	1200	984.17	1221						
10	844.17	1242	886.67	1201	1018.00	1221						
11	869.17	1250	887.67	1201	1003.50	1221						
12	849.50	1250	938.17	1205	1009.50	1221						
13	857.67	1247	900.17	1206								
14	805.17	1250	944.33	1206								
15	801.50	1250	879.33	1206								

Table 7: Direct comparison of the solution quality in terms of incurred cost on the test data for the genetic algorithm results from Table 6 and the greedy heuristic results from Table 2. The results quickly lead to the conclusion that the genetic algorithm clearly outperforms the greedy heuristic by a significant margin across the different number of windows and minimum window length possibilities.



(a) Greedy heuristics cost for a hop length of $\theta = 1$.



(b) Genetic algorithm cost.

Figure 1: Intuitive overview of the solution quality in terms of incurred cost on the test data, computed for a different number of quality windows and minimum window lengths using the weighted sum of false positives and false negatives introduced in Equation 1. It is quickly evident from the obtained results that the genetic algorithm implementation outperforms the greedy heuristic. For an intuitive visualization of the best solution achieved by the genetic algorithm we refer to Figure 2.

Using these results, we conclude that the genetic algorithm clearly outperforms the proposed greedy heuristic in terms of solution quality by a clear margin. Furthermore, the parametrization of the genetic algorithm which was described in Section 4.2.1 also ensures its competitiveness in terms of computation time on the training data. As an example, we consider the best solution cost achieved by the greedy heuristic with a hop length of $\theta = 1$ in Table 2: after 316.61 seconds and using 5 windows each with a minimum width of 10, the cost on the test data was 1169. Using the same number of windows and the same minimum window length, the ge-

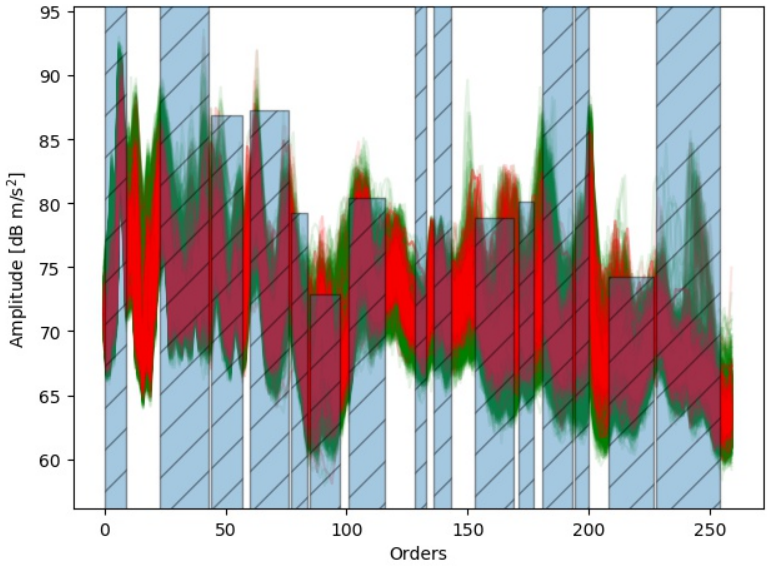


Figure 2: Overview of the best solution obtained by the genetic algorithm, displayed as a set of quality windows each consisting of a left and right border and an upper threshold. The visualized curves represent the vibroacoustic order spectra corresponding to a random subset of the test data: the red order spectra are representative of faulty BNA which belong to class 1, while the green order spectra characterize BNA with a good quality which thus belong to class 0.

netic algorithm needed slightly more time, 347.34 seconds, to achieve a better cost of 1037.5. However, keeping the minimum window width fixed at 10, the genetic algorithm could also achieve a cost of 1089.67 in 140.43 seconds with only 2 used windows, demonstrating its superiority.

The best solution achieved in the present work was obtained on the test data by the genetic algorithm in 1184.60 seconds using 15 windows with a minimum window length of 5, for which the incurred cost amounted to 801.50. From a practitioner's perspective, this result is highly important: for an initial number of 160 false positives, the currently employed BNA quality classification method features a cost of 1600 on the test data, computed using the same factor of $\varepsilon = 10$ in Equation 1. The usage of the genetic algorithm would thus lead to a reduction of 49.91% of the scrap costs in the BNA vibroacoustic EOL test. An intuitive visualization of the best performing solution, together with a random data subset containing samples from both classes, is offered in Figure 2.

6. Conclusion

Our work addresses a common operational problem faced by manufacturers of mechanical assemblies, who must define and set quality criteria for the assembly lines of the subcomponents of the assembly, in order to ensure the quality of the final product. We used a real-world situation encountered by *thyssenkrupp Presta AG* as a practical example, where quality criteria must be defined for the ball nut assemblies' vibroacoustic behavior in order to guarantee the quality of the final products they are assembled into, the steering gears. For this specific problem and extending previous work (Bucur et al., 2019b,c,a), we first proposed a MILP formulation for the data-driven optimization problem seeking so-called quality windows for the ball nut assemblies' vibroacoustic order spectra, with the outcome of the steering gear quality test as the corresponding quality labels. Upon ascertainment of the unsuitability of the MILP approach for daily operations due to its long computation time, we further offered two heuristic techniques as potential solving approaches. The first approach consists of a redesigned genetic algorithm which improves the original design proposed in (Bucur et al., 2019b) with new crossover and mutation mechanisms, while the second approach features a greedy heuristic, suitable for obtaining quick initial solutions. Solving the ball nut assemblies' vibroacoustic quality classification problem via quality windows imposed on the order spectra definitely represents a restriction in the feature space due to the consideration of solely the order spectra maxima within the borders of the quality windows as features. This restriction was disregarded in (Bucur et al., 2019c), where multiple machine learning-based methods were used to learn features from the order spectra. Still, the achieved scrap cost reduction of 49.91% in the present work, paired with the easy interpretability of the quality windows approach when compared to machine learning approaches, represent a simple, yet attractive, alternative to practitioners.

A potential direction for future applied research consists in our opinion of the usage of the solutions of the greedy heuristic as starting solutions for the genetic algorithm and for the MILP approach. Carefully ensuring that the genetic algorithm does not converge to local minima, it would be interesting to measure the computation time gain achieved with the usage of greedy heuristic solutions as initial population individuals.

7. Declaration of Interest

The present work was developed during a cooperation project with the company *thyssenkrupp Presta AG*. During the project, both Mr. Paul Alexandru Bucur and

Mr. Philipp Armbrust were at the same time employees of *thyssenkrupp Presta AG* and students at the University of Klagenfurt.

References

Almgren, H. (1999). Towards a framework for analyzing efficiency during start-up: An empirical investigation of a Swedish auto manufacturer. *International Journal of Production Economics*, *60-61*, 79–86.

Almgren, H. (2000). Pilot production and manufacturing start-up: The case of Volvo S80. *International Journal of Production Research*, *38*, 4577–4588.

Bucur, P. A., Frick, K., & Hungerländer, P. (2019a). Correlation Analysis Between the Vibroacoustic Behavior of Steering Gear and Ball Nut Assemblies in the Automotive Industry. In *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization* (pp. 1253–1262). Cham: Springer International Publishing.

Bucur, P. A., Frick, K., & Hungerländer, P. (2019b). Predicting the Vibroacoustic Quality of Steering Gears. In *Operations Research Proceedings 2018* (pp. 309–315). Cham: Springer International Publishing.

Bucur, P. A., Hungerländer, P., & Frick, K. (2019c). Quality classification methods for ball nut assemblies in a multi-view setting. *Mechanical Systems and Signal Processing*, *132*, 72–83.

Colledani, M., Tolio, T., & Yemane, A. (2018). Production quality improvement during manufacturing systems ramp-up. *CIRP Journal of Manufacturing Science and Technology*, *23*, 197–206.

Diez-Olivan, A., Ser, J. D., Galar, D., & Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0. *Information Fusion*, *50*, 92–111.

ElMaraghy, H., Schuh, G., ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., & Bernard, A. (2013). Product variety management. *CIRP Annals*, *62*, 629–652.

Forrest, J., Ralphs, T., Vigerske, S., LouHafer, Kristjansson, B., jpfasano, Edwin-Straver, Lubin, M., Santos, H. G., rlougee, & Saltzman, M. (2018). coin-or/cbc: Version 2.9.9. Accessed: 24.4.2019.

Gaspar, J., Fontul, M., Henriques, E., Ribeiro, A., Silva, A., & Valverde, N. (2016). Psychoacoustics of in-car switch buttons: From feelings to engineering parameters. *Applied Acoustics*, *110*, 280–296.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* volume 1. MIT press Cambridge.

Google OR-Tools v7.0 (2019). Google OR-Tools v7.0. Accessed: 24.4.2019.

Hirsch, V., Reimann, P., Kirn, O., & Mitschang, B. (2018). Analytical Approach to Support Fault Diagnosis and Quality Control in End-Of-Line Testing. *Procedia CIRP*, *72*, 1333–1338. 51st CIRP Conference on Manufacturing Systems.

Huang, H. B., Huang, X. R., Li, R. X., Lim, T. C., & Ding, W. P. (2016). Sound quality prediction of vehicle interior noise using deep belief networks. *Applied Acoustics*, *113*, 149–161.

Kwon, G., Jo, H., & Kang, Y. J. (2018). Model of psychoacoustic sportiness for vehicle interior sound: Excluding loudness. *Applied Acoustics*, *136*, 16–25.

Lieber, D., Stolpe, M., Konrad, B., Deuse, J., & Morik, K. (2013). Quality Prediction in Interlinked Manufacturing Processes based on Supervised and Unsupervised Machine Learning. *Procedia CIRP*, *7*, 193–198. Forty Sixth CIRP Conference on Manufacturing Systems 2013.

Stylidis, K., Madrid, J., Wickman, C., & Söderberg, R. (2017). Towards Overcoming the Boundaries between Manufacturing and Perceived Quality: An Example of Automotive Industry. *Procedia CIRP*, *63*, 733–738. Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems.

Swart, D., & Bekker, A. (2019). The relationship between consumer satisfaction and psychoacoustics of electric vehicle signature sound. *Applied Acoustics*, *145*, 167–175.

Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). Data-driven smart manufacturing. *Journal of Manufacturing Systems*, *48*, 157–169. Special Issue on Smart Manufacturing.

Wang, X., Liu, M., Ge, M., Ling, L., & Liu, C. (2015). Research on assembly quality adaptive control system for complex mechanical products assembly process under uncertainty. *Computers in Industry*, *74*, 43–57.