Comparison of grouping algorithms to increase the sample size for statistical process control

Jonathan Simon Greipel^{1,2} · Gina Nottenkämper² · Robert Heinrich Schmitt²

Received: 9 December 2019 / Accepted: 8 April 2020 / Published online: 20 April 2020
© The Author(s) 2020

Abstract

In this study, we present and compare four grouping algorithms to combine samples from low volume production processes. This increases their sample sizes and enables the application of Statistical Process Control (SPC) to low volume production processes. To develop the grouping algorithms, we define different grouping criteria and a general grouping process. To identify which algorithm is optimal, we deduct following requirements on the algorithms from real production datasets: their ability to handle different amount of characteristics and sample sizes within each characteristic as well as being able to separate characteristics possessing distributions with different spreads and locations. To check the fulfillment of these requirements, we define two performance indices and conduct a full-factorial Design of Experiments. We achieve the performance indices for each algorithm by using simulations with artificial data incorporating the aforementioned requirements. One index rates the achieved group sizes and the other one the compactness within groups and the separation between groups. To validate the applicability of grouping algorithms within SPC, we apply real production data to the grouping algorithms and control charts. The result of this analysis shows that the grouping algorithm based on cluster analysis and splitting exceeds the other algorithms. In conclusion, the grouping algorithms enable the application of SPC to small sample sizes. This provides companies, which produce in low volumes, with new means of reducing scrap, generating process knowledge and increasing quality.

Keywords Grouping algorithm · SPC · Low volume production · Pooling of data · Algorithm performance

1 Introduction

Heterogeneous customer requirements and growing competition urge companies to offer more and more product variants [1]. Therefore, they increasingly produce small series, small batches or single pieces [2]. Due to the small quantities, Statistical Process Control (SPC) is no longer applicable as large sample sizes of approximately 125 values are required to achieve reliable estimations and information [3]. Companies can no longer make quality relevant decisions or prove their process capability to their customer. The control limits of control charts

become too large to detect instabilities and are therefore—in practice—no longer useful. Another reason is that the required confidence intervals of the capability indices are dependent on the sample size and become too large to be informative. To overcome the problem of too small sample sizes different approaches exist [4]. Change point models [5], control charts with greater sensitivity [6], self-starting charts [7], pre-control-charts [8], Bayesian approaches [9], control charting process and parameters instead of product features [10] are exclusively for control charts. Normalizing and grouping of similar features (grouping) [11] as well as bootstrapping [12] are methods

✉ Jonathan Simon Greipel, J.Greipel@wzl.rwth-aachen.de | ¹Production Engineering Cluster, Campus-Boulevard 30, 52074 Aachen, Germany. ²Chair of Production Metrology and Quality Management, Laboratory for Machine Tools and Production Engineering WZL, RWTH Aachen University, Aachen, Germany.

for the general application of SPC. This is advantageous, as we cannot only use the enlarged dataset for control charts but also for more informative process capability analyses with more reliable confidence limits due to an enlarged sample size. Whereas grouping is preferable, as according to Ryan, bootstrapping does not work well with SPC applications for example when distributions are skewed [12]. When grouping, the data originates from real production data and we do not artificially enlarge the data, which makes the SPC results more trustworthy and applicable in a production environment. In addition, the ISO 7870-8 *Control charts—Part 8: Charting techniques for short runs and small mixed batches* suggests to group data for enlarging the sample size for the application to SPC, but does not provide a sufficient technique how to do so. For these reasons, this paper scrutinizes grouping approaches. Especially grouping algorithms—automated grouping approaches—are of interest as the time and the knowledge of what to group as well as methods on how to group are often not available within companies. Additionally, grouping algorithms provide more objective results compared to expert knowledge.

The requirements on grouping and respectively on grouping algorithms originate from the application of SPC. The idea of SPC is monitoring the variation of product features to distinguish between natural variation and variation due to assignable causes originating from a production process. By identifying and eliminating an assignable cause, we reduce the overall variation and thus optimize the process [13]. SPC is obliged to a great number of data of the corresponding products to achieve significant statements on which we can base decisions. Therefore, for low volume production, grouping is a solution to achieve the required data [14]. When grouping data, we add a third type of variation: variation due to grouping. The different product properties (e.g. different materials) and the different process set-ups (e.g. different clamping) cause this variation. Grouping can obscure the variation due to assignable causes and lead to either false alarms or no alarms at all in control charts. To minimize variation due to grouping, grouping algorithms are applied which only combine statistically similar groups. We use following definitions regarding grouping algorithms in the further context of this paper:

- *Similar features* are features of the same type produced by processes of the same technology and have quality characteristics with the same distribution, homogeneous spread and location [11].
- *Grouping of similar features* means combining similar features descending from different products [11].
- *Grouping algorithms* automatically identify and combine similar features to increase the database by using

historical measurement data of similar features and statistical methods [11].

Not many grouping algorithms exist and furthermore they have not been compared systematically yet. In this paper, we firstly introduce the state of the art regarding grouping algorithms. Secondly, we set up requirements on grouping algorithms. Thirdly, we introduce statistical methods used within grouping algorithms. Fourthly, we present four different grouping algorithms, which we developed ourselves, in consideration of the requirements and different criteria deducted from the concepts of discretization processes. Furthermore, we also vary two of the four grouping algorithms slightly. Fifthly, we test the algorithms for their reliability and usefulness. We choose or design performance indices based on the requirements. By conducting a full-factorial Design of Experiments, which is based on a simulation with artificial data on the grouping algorithms, performance indices are calculated. Sixthly, with these performance indices we evaluate the requirements for their fulfillment. Seventhly, a summary and conclusion is provided which grouping algorithm and which setting parameters are the best.

2 State of the art on grouping algorithms

In literature, two grouping algorithms for the application of SPC to small quantities exist: by Al-Salti et al. and by Wiederhold. In the following, we provide a short review on these grouping algorithms.

2.1 Grouping algorithm according to Al-Salti et al

The grouping algorithm by Al-Salti et al. is based on Group Technology (GT) [15] and similar to the works of Cheng [16]. GT is a manufacturing technique. GT reduces the costs of manufacturing small batches. When using GT component groups are created based on similarities in design and manufacturing for example by grouping components with a similar geometry. The following procedures to group components according to GT exist:

- Visual search, where an expert defines variants based on part geometry [17].
- A Coding and Classification System, which experts create depending on the components of the company. The experts give each property of the component a number, e.g. the material, clamping, etc. Like this, each variant receives a specific number (Opitz-code) [18]. The similarity of the sequence of numbers indicates the similarity of the variants [17].

- The cladistics originates from biology. In manufacturing, it includes the evolutionary development of a component and its relationship to similar components. An expert creates a tree about the relationship of similar component features [19].

A deficit of GT is that it is a systematic approach, which only classifies components mostly according to their geometry. GT does not take into consideration, whether this classification also corresponds to empirical analyses with historical measurement data of the features.

Al-Salti et al. extend GT by the mentioned missing empirical analysis and assume that the features of those grouped components have similar quality characteristics as well as statistical properties and therefore use GT to increase the database for applying SPC [15]. The idea is that the data of similarly manufactured components (e.g. equal work plans) also possess similar statistical properties.

Al-Salti et al.'s approach consists of two steps. Firstly, an expert creates a *Classification and Coding System*. Hereby each property of a component gets a digit for example each material. The expert encodes all the properties of each component accordingly and creates a database with the codes and the historical measurements of the components.

Secondly, if data for an SPC analysis of a component's feature is needed it is firstly evaluated if sufficient data is available. If not one or several digits of the code freed. An expert makes the decision, which digit is freed, by testing the potentially similar components with statistical methods. Therefore, the expert retrieves historical measurement data of the components' features from the database and creates samples based on the code. Hypothesis tests for homogeneity of spread by using the F-test and for homogeneity of location by using a one-way Analysis of Variance (ANOVA) tests these groups. To compensate varying dimensions of potentially similar features the deviation from nominal (DNOM) of the features is formed. If the test proves homogeneity, the expert groups the components' features. This means these digits/properties of the components are not relevant any longer, meaning more components have the same code and the database is larger. Result are statistically similar features for the application to SPC [15].

2.2 Grouping algorithm according to Wiederhold

Wiederhold groups individual similar features of components according to factor and factor levels in order to apply SPC. The grouping algorithm allows an automated analysis of non-parametric empirical process data [11]. The algorithm consists of three steps. The algorithm repeats

the three steps until it has considered all factors and formed all groups.

In the first step, an expert creates the required database. It contains historical measurement data and possibly relevant factors and factor levels about the process and the product (material, tool, etc.) meant to cause variation due to grouping. Expert knowledge determines these factors and factor levels. The expert answers following question to identify these: "What are differences of similar features that could prevent a grouping? Possible differences identified by experts could be the material, the geometric element, the tool, clamping etc. the feature has been produced in. These differences are translated to factor and factor levels and used to preprocess the data for the application to the algorithms. Additionally from the historical measurement data, the DNOM is calculated. The resulting database serves as input for the next step [11].

The second step involves checking all factors and factor levels for their homogeneity of spread and location by using the Levene- and the Kruskal–Wallis-test. The algorithm sees the factor with the smallest p value as the one causing the highest inhomogeneity and chooses it first. On the factor levels of this factor, a cluster analysis is applied. As input for the cluster analysis, the spread and location of the factor levels are used. On the first run, the algorithm forms two clusters. If the tests show that these clusters are not homogeneous, the algorithms increases the number of clusters by one and repeats the cluster analysis. If both, spread and location are homogeneous, the algorithm forms the groups [11].

The third step consists of recursively calling the entire grouping algorithm. On each recursion, the algorithms examines the most significant—not yet examined—factor. It is possible that the groups split up further. To realize the recursion calls, all so far created groups are selected one after the other and used as database. The grouping algorithm terminates as soon as it has considered all factors [11].

2.3 Conclusion

Al-Salti et al.'s grouping algorithm is not suitable for data with arbitrary distribution, as the incorporated ANOVA requires the residuals of data to be normally distributed. Furthermore, the algorithm is not fully automatic. Both grouping algorithms are costly for a practical application in a production environment, because they require a lot of expert process knowledge to determine possibly relevant factors, for creating a *Coding and Classification System* or pre-processing the required database. This knowledge is often not available or its creation and application very time-consuming for companies. Furthermore, the performance of these algorithms has never been tested or

compared. Within this paper, new automatic grouping algorithms are developed and compared. In the following, we describe the methods used within these algorithms.

3 Commonly used methods within grouping algorithms

In the following, we introduce the building blocks for grouping algorithms. According to these building blocks, we design new easy-to-use grouping algorithms and introduce them in the subsequent chapter. Grouping algorithms need to successfully measure if potential groups possess a homogenous spread and location. We adopt Wiederhold's assumption, that the distributions of features originating from the same technology are always the same; we do not include Goodness-of-fit-tests in the grouping algorithms [11]. To measure the homogeneity of spread and location we use the p value.

The p value is the probability to achieve the same or extreme result by a sample if the null hypothesis is true [13]. The grouping algorithms heuristically use the p value as a measure of homogeneity. The higher the p value the more homogenous the combined groups [11]. Consequently, a group with a p value smaller than the significance level is inhomogeneous. In the following, we introduce the statistical tests applied to check the homogeneity of spread and location within our grouping algorithms and choose the best fitting ones as building blocks.

3.1 Testing the homogeneity of spread

Tests of homogeneity of spread aim to check whether the spreads of groups are the same. The following null hypothesis $H_0 : \sigma_1^2 = \sigma_2^2 = \dots = \sigma_n^2$ and alternative hypothesis $H_1 : \text{At least two variances are different}$ are tested. Possible candidates to do so are the Brown-Forsythe-, Barlett- and Levene-test. To make the grouping algorithm applicable for features with arbitrary distributions and as the Brown-Forsythe-test and the Barlett-test require a normal distribution, we choose the Levene-test. The Levene-test examines the equality of group deviations from a population [20]. It calculates the deviations of each dependent variable from the arithmetic mean of the group for the test. The Levene-test performs an ANOVA on the calculated variances. The ANOVA determines if H_0 is accepted or not [20].

3.2 Testing the homogeneity of location

ANOVA, Median test and Kruskal-Wallis test check whether two or more samples come from a population with the same location parameter. The null

hypothesis $H_0 : \mu_1^2 = \mu_2^2 = \dots = \mu_n^2$ and alternative hypothesis $H_1 : \text{At least two means are different}$ are tested. For the algorithms, we choose the Kruskal-Wallis-test as the ANOVA requires residuals of the distributions to be normally distributed and the Median-test has low power [20]. The Kruskal-Wallis-test uses the mean of the ranking values of groups to check the homogeneity of location. First, the Kruskal-Wallis-test examines all measured values without their group membership, sorts them in order and assigns them a rank. With those ranks and by considering the group membership again it conducts a one-way ANOVA. The Kruskal-Wallis-test requires the distributions of the groups to be of the same type and form. This requirement is met with the assumption that distributions of features originating from the same technology always have the same distribution [11].

4 Developed grouping algorithm alternatives and adaption of existing grouping algorithms for evaluation

This paragraph gives an overview on properties to develop grouping algorithms, the process of grouping, requirements on grouping algorithms and variations to test for homogeneity of location and spread. We then use these to create the new grouping algorithms.

We deduct the properties and process of grouping from discretization algorithms for continuous data. Lui et al. generalized these properties for these discretization algorithms [21]. Due to their similarity to grouping algorithms, we take on the properties and slightly adapt them for grouping. We deduct the following properties from discretization algorithms and transfer them as general properties to construct new grouping algorithms:

- Supervised/unsupervised: Supervised methods consider continuous data as well as a class information. Unsupervised methods only consider the continuous data without a class information.
- Splitting/merging: Splitting methods consider the whole dataset and split it into groups. Merging methods take each group separately and try to merge them into bigger groups.
- Direct/incremental: Direct methods have an information about how many groups they should create. Incremental methods possess a stopping criterion according to which they stop creating further groups.
- Univariate/multivariate: Univariate methods discretize one continuous feature at a time. Multi-variate methods discretize multiple features simultaneously.

We can simplify the process of grouping into three steps. In these steps, we integrate the above-mentioned properties to create new grouping algorithms. The steps are:

- **Evaluation:** For splitting methods, the process seeks the best separation. For merging methods, the process seeks the best combination of groups. Therefore, an evaluation function is used. Typically, the process uses measures like entropy or statistical measures like homogeneity (p value).
- **Splitting/Merging:** The process splits or merges the data to smaller or bigger groups.
- **Stopping:** In this step, the process analyses if it has met a stopping criteria. The stopping criteria is a trade-off of the size of the groups and the required homogeneity.

We deduced the following requirements on grouping algorithms by discussions with industry and by analyzing several datasets from real production processes, which companies, which produce in low volumes, provided. Grouping algorithms need to fulfill following requirements within a production environment:

- Grouping algorithms must be capable of working with variable sample sizes to enable them to process real data from small series production.
- Grouping algorithms must be capable of working with different distributions. Therefore, an expert does not have to make an assumption about the distribution of the measurements.
- Grouping algorithms must be able to keep the groups as homogeneous as possible to avoid alpha errors and beta errors.
- Grouping algorithms should require as little expert knowledge as possible to enable an easy automation of the process of grouping.
- The significance level of grouping algorithms should be variable to make the risk of rejecting a hypothesis adjustable.

We developed four different grouping algorithms according to the process of grouping and the requirements. As evaluation criteria, we use the p value and as stopping criteria we use the significance level of the Levene- and Kruskal–Wallis-test.

- Grouping based on cluster analysis and splitting
- Grouping based on ChiMerge and splitting
- Grouping based on comparison and merging
- Grouping based on splitting

Differently to the grouping algorithms by Al-Salti et al. and Wiederhold our four grouping algorithms do not require expert knowledge [11, 15]. This is an advantage as often within companies, this knowledge is not fully available and integrating the knowledge into the dataset by pre-processing the dataset is often too time-consuming for companies. We achieve this by only using class information for equal features. Therefore, the only expert knowledge and data pre-processing, which our algorithms require is the division of the features in different feature classes like length, diameter, etc. Furthermore, all feature regardless of their feature class have a unique ID. The unique ID contains all information about the feature like material, tolerance etc. The algorithms then perform the division of the features into groups only based on statistical variability.

For creating grouping algorithms, two options to use p values are possible. The Levene-test must always be applied before the Kruskal–Wallis-test as the Kruskal–Wallis-test requires a homogenous spread as a condition to be applicable. According to this condition, two options to create grouping algorithms are possible:

1. *Testing p values of spread and location separately* First conducting the whole algorithms with the p value of the Levene-test and group accordingly. In the second step, using the groups created as input for re-running the algorithm but this time applying the Kruskal–Wallis-test to calculate the p values and split groups further to achieve the homogeneity of location.
2. *Pooling the p values of spread and location* To get the set of p values the algorithm firstly conducts the Levene-test. If the spread proves to be homogeneous according to the Levene-test the algorithm conducts the Kruskal–Wallis-test and its p value is used instead. The p values of the Levene-test, which rejected the null hypothesis and the ones of the Kruskal–Wallis-test are pooled. From that pool, the algorithm chooses the smallest p value to split the group.

In the following, we explain the algorithms based on the defined principles and the simplified grouping process. We do not compare the algorithms with the algorithm Al-Salti et al. as this algorithm is based on an ANOVA and only handles data with normally distributed residuals. We conduct a comparison with a simplified version of Wiederhold (algorithm based on cluster analysis and splitting) as the original algorithm firstly requires data (factor and factor levels), which are then no longer comparable with the other algorithms (chapter 5 and 6) and secondly for the real datasets was not available due to a lack of expert knowledge (chapter 7).

4.1 Algorithm based on cluster analysis and splitting

The algorithm based on cluster analysis and splitting is an adaptation of WIEDERHOLD'S algorithm, which only takes the class information into consideration [11]. The algorithm is depicted in Fig. 1.

First step consists of calculating the DNOM and creating the database. The p values are calculated by using this database and applying the Levene-test and Kruskal–Wallis-test. If the p value is below the significance level, the cluster analysis, which is also used by Wiederhold, is applied on the spread and the location of the features [11].

Wiederhold's grouping algorithm uses the divisive hierarchical DIANA clustering [11]. Differently to any other clustering all data is in one cluster at the beginning and divided into clusters by following approach [22]:

1. In the beginning, all observations are in one cluster;
2. Choose the cluster with the largest distance metric and divide it into two subclusters.
 - (a) Find the object with the largest mean distance to all other objects. This object is the first element of the new splitter group.

- (b) For each object in the larger cluster, calculate the mean distance to the remaining objects and compare it with the mean distance to the objects in the splitter group. The objects closer to the splitter group are assigned to the splitter group.

3. Repeat this process until each cluster contains only a single observation.

As a distance metric, the Euclidean or the Manhattan distance are possible choices. The Euclidean distance is a straight-line distance between two points in Euclidean space. The Manhattan distance is a grid-like distance represented by the sum of absolute differences of the Cartesian coordinates. WIEDERHOLD proposes the Manhattan since the estimated mean and standard deviation are independent of each other [11]. As an output, the algorithm creates a dendrogram, which orders possible clusters according to similarity.

After the group is split for each created subcluster the p value is calculated again with the Levene-test and Kruskal–Wallis-test. If the lowest p value of any subcluster still proves to be below the significance level the subcluster is split further by increasing the amount of

Fig. 1 Flowchart for algorithm based on cluster analysis and splitting. The p value for the whole data is calculated. If the p value is smaller than the significance level, cluster analysis is used to create groups. If these are above the significance level they are homogenous

clusters by one. This is repeated until a homogenous state is achieved and all lowest p value are above the significance level.

4.2 Algorithm based on ChiMerge and splitting

The algorithm based on ChiMerge and splitting in Fig. 2 is explained by dividing it into three steps. In the first step, the database is created. For each measurement, the deviation between the target value and the measured value is

Fig. 2 Flowchart for the algorithm based on ChiMerge and splitting. The ChiMerge-algorithm is used to create intervals. These intervals are used as input for a frequency table from which groups are deducted according to the class frequency

calculated and assigned to the feature (DNOM). The database is sorted according to the size of the deviation. It serves as input for the second step.

In the second step, using the sorted database the algorithm forms intervals. The creation of the intervals is divided into three sub-steps. First, the algorithm splits the sorted database from the first step between changes of different features. This results in intervals. In the next sub-step, the algorithm performs the Chi square-test for all adjacent intervals and saves the Chi square-statistic. In the second sub-step—after all Chi square-statistics have been calculated and saved—the algorithm combines the units with the smallest Chi square-statistic. This results in new intervals in each run. In the third sub-step, the algorithm only joins adjacent intervals if the Chi square-statistic is below the Chi square-threshold determined by a specified significance level. If all values are above the significance level, the intervals generated so far serve as input for this step.

In the third step, the grouping algorithm meets the requirement that it has assigned all measurements of a feature to the same group. During the creation of the intervals this assignment of the measurements to the features is ignored [20]. The algorithm assigns each characteristic value to the interval in which the characteristic is most frequently represented by using a frequency table. Once the algorithm has assigned all characteristic values to an interval, it regards them as a group. The algorithm checks the

groups using the Levene-test and Kruskal–Wallis-test. If the hypothesis tests show that the formed groups are still inhomogeneous, the algorithm splits them up by applying the algorithm again. All steps are recursively repeated until the Levene-test and Kruskal–Wallis-test prove the homogeneity of spread and location for each group.

4.3 Algorithm based on comparison and merging

The algorithm based on comparison and merging in Fig. 3 is a greedy algorithm, it selects the best result without considering future steps. It consists of three steps. In the first step, it generates the database containing the DNOM of the features. The resulting database serves as input for the next step.

In this second step, the algorithm applies the Levene-test. Its aim is to create a pre-grouping for the Kruskal–Wallis-test. The second step consists of two sub-steps. First, the algorithm forms all possible combinations of two features. Second, the algorithm applies the Levene-test to each of these combinations. The p value resulting from the tests are stored. Third, after the algorithm has saved all p values, it selects the combination with the highest p value. If the p value is below the specified significance level, it dissolves the combination into its original parts, because the p value indicates a significant difference between the variances. The algorithm repeats the second step using the new groups until the highest calculated p value is lower

Fig. 3 Flowchart for algorithm based on comparison and merging. Pairwise combinations are created. For each combination a p value is calculated. The combination with the largest p value is merged to a group

than the significance level. The groups created up to this point serve as input for the third step.

In the third step, the homogeneity of the location is tested. For this purpose, the Kruskal–Wallis-test is used. The second and third steps are almost identical. The difference is that the algorithm uses the Kruskal–Wallis-test instead of the Levene-test to check the homogeneity of the location and calculate the p value. Each step resulting from the second step serves as input for the third step.

We created a variation of this grouping algorithm, where the algorithm pools the p values of spread and location. Accordingly the third step is dropped. The variation is referred to as *algorithm based on comparison and merging 2*.

4.4 Algorithm based on comparison and splitting

The algorithm based on comparison splitting in Fig. 4 is a greedy algorithm. The algorithm consists of three steps. In the first step, the database containing all features and DNOM is created. The database from the first step serves as input for the second.

The second step consists of sub-steps. First, the algorithm forms various combinations of features. To do this, the algorithm examines entire database and excludes one feature. n different combinations with $n + 1$ features exist. After the algorithm has created all the combinations, each p value of the Levene-test is determined and stored. That means the algorithm calculates the p value of each combination without considering the excluded group. The Levene-test checks the sample for its homogeneity of spread. After the algorithm has performed all Levene-tests and stored all p values, it selects the combination with the largest p value. It assumes that this sample is the most homogeneous. The algorithm saves the feature excluded in this combination separately with other features. If the p value of the considered combination is above the significance level, the algorithm forms the group with all included features. If this is not the case, the algorithm repeats the process of excluding a feature, whereas the previously excluded feature is no longer considered. If the algorithm achieves and saves a homogeneous group, the excluded features are the new database and the algorithm repeats the second step with those features.

In the third step, the algorithm checks the homogeneity of location of the found groups of the second step. The algorithm repeats the second step, except that it uses the Kruskal–Wallis-test instead of the Levene-test. Output of this test are groups with homogenous spread and location.

A variation of this grouping algorithm exists, where the p values of spread and location are pooled. Accordingly the third step is dropped. The variation is referred to as *algorithm based on comparison and splitting 2*.

5 Performance metrics and evaluation method

To evaluate the different algorithms a full factorial Design of Experiments (DoE) is applied. We define two performance metrics showing the quality of the grouping by using the main requirements the grouping algorithms have to fulfill. We identified influences on the grouping by firstly scrutinizing datasets with real measurements and secondly by conducting expert discussions on what factors (chosen significance level, group size, etc.) might affect the grouping algorithms. According to these effects, a simulation with the programming language *R* was set up.

5.1 Influences on groupings

We conducted a DoE to determine the effects between influencing factors and our target variables *size of created groups, separation and compactness*. In order to be able to recognize the interrelationships of the individual parameters, we selected a higher number of factor levels. We used the factor and factor levels listed in Table 1 to test how well the algorithms perform.

5.2 Target variables: figures of evaluation

For a successful application of SPC to low volume production the following requirements have to be fulfilled:

- The algorithms should provide as large groups as possible to provide a sufficiently large database for further statistical methods like control charts or process performance evaluations.
- The algorithms should provide as homogeneous groups as possible regarding location and spread to realize a good average run length for control charts to prevent false alarms due to groups with differing location and spread.

Deducted from those two requirements to evaluate the grouping algorithms, we chose following performance indices *size of created groups, separation and compactness (SDBW)*:

5.2.1 Size of created groups

The index shows the average sizes of groups in relation to sizes of groups created by the algorithm. The following formula is used:

Fig. 4 Flowchart for algorithm based on comparison and splitting. Combinations are created by taking out each class out of the group once. For each combination a p value is calculated. The combination with the largest p value is grouped

Table 1 Factor, factor levels and the reason for choosing them for testing different algorithms

Factor	Factor levels	Reason
Amount of groups	2, 7, 15	We use this factor to scrutinize how well the algorithms perform with small and large amount of groups
Size of groups	5, 15, 40, variable	The factor exists to analyze the effect of how well the algorithms perform regarding small, large and variable sizes of groups (amount of members of a group)
Distribution	Normal, fold, gamma, uniform	We use different distributions to analyze their effect on grouping. As we assume that the distributions of features originating from the same technology always have the same distribution variable distributions are not tested
Location	0.1, 0.5, 1, 3	We use this factor to analyze the effect of the distances between different distributions. Especially the effect in combination with the spread is of interest
Spread	0.2, 0.5, 1, 5	With this factor we analyze the effect of overlaying spreads and how well the algorithm can distinguish them
Significance level	0.001, 0.05, 0.3, 0.5, 0.9	The significance level is a heuristic for the homogeneity. The higher the p value the more homogenous the combined groups as described in chapter 3. To analyze the effect of how the significance level of the Kruskal–Wallis-test and Levene-test affect the amount of groups and the quality of groups formed, we not only choose common significance levels (e.g. 0.001, 0.05), but also higher ones to see their effect on the group homogeneity

$$t = \frac{\sum_{i=1}^n x_{\text{depositedgroups},i}}{n_{\text{depositedgroups}}} \cdot \frac{\sum_{j=1}^m x_{\text{createdgroupsbyalgorithm},j}}{n_{\text{createdgroupsbyalgorithm}}}$$

x : size of group; n : amount of deposited groups; i : number of deposited groups; m : amount of the created groups; j : number of created groups.

t can be between 0 and 1 whereas a smaller number indicates that group sizes increased by grouping whereas 1 indicates that group sizes stayed the same and no groups were grouped.

5.2.2 SDbw: compactness and separation

SDbw is made up of two measures: compactness and separation. A small value says the measured values within a group are very similar. *SDbw* takes on values between 0 and 2. A *SDbw*-value close to 0 indicates a high separation and compactness. *SDbw* consists of following formula according to Liu et al. [23]:

Compactness The compactness of a group determines how similar the individual measured values within a group are. It shows how close the measured values are. The compactness of a group is determined without placing it in relation to other groups. It shows how well the individual measurements within a group fit. To evaluate the compactness, we chose the formula by Liu et al., as it proved the best according to simulations conducted by Liu et al. [23].

$$Scat(NC) = \frac{1}{NC} \sum_i \frac{\|\sigma(C_i)\|}{\|\sigma(D)\|}$$

Scat: Separation; *NC*: number of clusters; σ : variance; C_i : the i -th cluster; D : dataset.

Separation The separation of groups indicates how clearly the groups differ from each other. It shows, if points are clearly assigned to a group, or if there are overlaps or inconsistencies between groups. The separation is evaluated in relation to other groups. If only one group exists, no separation measure is calculated. To evaluate the separation, we chose the formula by Liu et al., as it proved the best according to simulations conducted by Liu et al. [23]:

$$Dens_bw(NC) = \frac{1}{NC(NC-1)} \sum_i \left[\sum_{j \neq i} \frac{\sum_{x \in C_i \cup C_j} f(x, u_{ij})}{\max \left\{ \sum_{x \in C_i} f(x, c_i), \sum_{x \in C_j} f(x, c_j) \right\}} \right]$$

$$SDbw = Dens_bw(NC) + Scat(NC)$$

SDbw: Compactness and Separation; *Dens_bw*: Compactness; *Scat*: Separation; *NC*: number of clusters.

D : dataset; n : number of objects in D ; c : center of D ; P : attributes number of D ; NC : number of clusters; C_i : the i -th cluster; n_i : number of objects in C_i ; C_i : center of C_i ; (C_i) : variance vector of C_i ; (x, y) : distance between x and y ; $X_i = (X_i^T \cdot X_i)^{\frac{1}{2}}$

6 Evaluation of grouping algorithms

We used the programming language *R* and the development environment *RStudio* to implement the grouping algorithms as well as the simulation according to a full factorial DoE using the factors and their corresponding factor levels recorded in Table 1. We conducted the simulation twelve times and averaged the results. For their implementation, following functions in *R* were used: Levene-test, Kruskal–Wallis-test, ChiMerge of the *R* package *Discretization* and *SDbw* of the *R* package *NbClust*.

In Fig. 5 the main effects of the DoE are depicted. The ordinate axis shows the *SDbw* means of the main effects. The abscissa axis shows the examined factors and factor levels. If the mean of the *SDbw* are low it indicates a good separation and compactness.

From Fig. 5 and the DoE, we deduct the following about the factors and the algorithms:

- The amount of groups has a major effect on the *SDbw*. The more groups exist the better the *SDbw*. With an increasing amount of groups, the *SDbw* decreases, as the probability to form more homogenous groups increases.
- As the size of groups increases, the *SDbw* increases, as groups become less compact and less separable. Variable group sizes do not have a big effect (Variable group sizes were comparable to the group sizes of 5).
- The type of distribution does not have a great effect on the grouping algorithms, except the uniform distribution, as separation and compactness are hard to distinguish for this distribution.
- The location has a big effect on the *SDbw*. The further the distance between the groups the smaller the *SDbw* due to a better separation of the groups.
- The spread has a major effect on the *SDbw*. With increasing spread, *SDbw* increases due to less compactness and decreased separation.
- With increasing significance level, *SDbw* decreases slightly as groups become more compact.
- Pooling *p* values or evaluating them separately does not have a major effect on the *SDbw* as can be seen by the algorithms based on splitting or comparison.
- The algorithm based on cluster analysis and splitting forms groups with the lowest *SDbw*. Therefore the algorithm outperforms the other algorithms in achieving the most compact and separable groups.
- The algorithms based on comparison and splitting achieve the highest *SDbw*, which indicates that they perform worst.
- The algorithm based on the ChiMerge and splitting achieves a high *SDbw*.
- The algorithms based on comparison and merging perform similar to each other and are second after the algorithm based on the cluster analysis and splitting.

Fig. 5 Main effect diagram of *SDbw*. The different grouping algorithms are compared. The lower the *SDbw*-value the more homogenous are the groups created by the grouping algorithm. According to the evaluation the algorithm based on cluster analysis and splitting creates the most homogenous groups

Fig. 6 Main effect diagram of t . The lower the t -value larger the groups created by the grouping algorithm. According to the evaluation the Algorithm based ChiMerge and splitting creates the largest groups

In Fig. 6 the main effects of the simulation for t are depicted. The ordinate axis shows the means of the $SDbw$ regarding the main effects. The abscissa shows the examined factors and factor levels.

From Fig. 6 and the simulation, we deduct the following about the factors and the algorithms:

- As the amount of groups increases, t decreases. This indicates that it is more likely to cluster groups if more groups are available.
- As the group size increases, t increases. It shows the more values within a group are available the less likely it is to cluster groups. A reason could be that the location and spread of a group can be better estimated leading to less clustered groups.
- As the location between groups increases, the less groups are clustered. It shows that the algorithms fulfill the requirement of identifying homogenous groups.
- As the spread of the groups, increases the more groups are clustered. It shows that wide spreads affect the algorithms, which obscure finding homogenous groups.
- With increasing significance level, the less groups are clustered. It shows that the p value fulfills the requirement of being a heuristic for homogeneity.
- Pooling p values or evaluating them separately has an effect on t . Pooling leads to larger group sizes for the

algorithms based on splitting and smaller groups for algorithms based on comparison.

- The algorithms based on comparison and splitting cluster and the algorithm based on cluster analysis and splitting form the least groups and behave similar.
- The algorithms based on comparison and merging behave very similar and cluster the second most groups.
- The algorithm based on the Chi square-statistic clusters the most groups.

In conclusion, we deduct the following from Figs. 5 and 6: The simulation results suggest that the algorithm based on cluster analysis and splitting is the most optimal one to find compact and separable groups. The algorithm based on the Chi square-statistic clusters the most groups, but achieves a bad $SDbw$. To create groups for SPC the homogeneity of the groups is more important than the group sizes therefore we suggest using the algorithm based on cluster analysis and splitting. In the further progress of this paper, the focus is set on the algorithm based on cluster analysis and splitting.

In Figs. 7 and 8, the interaction diagram of the algorithm based on cluster analysis and splitting is depicted which provides information about how factors behave to each other. As the interaction diagrams of the other algorithms are very similar, we use the interaction diagram of the algorithm based on cluster analysis and splitting as example.

Fig. 7 Interaction diagram of SDbw of the algorithm based on cluster analysis and splitting. Size of groups/significance level have an interaction. Location/Spread have an interaction

Fig. 8 Interaction diagram for t of the algorithm based on cluster analysis and splitting. No interactions are found

Within an interaction diagram, two factors have an influence on each other if their lines do not run parallel to each other. Therefore, only the interactions between those factors are considered. From Figs. 7 and 8 and the simulation, following can be deduced:

- **Significance level/Amount of groups:** For small amount of groups a smaller significance level lead to a smaller *SDbw* and less groups are clustered. A larger amount of groups and a larger significance level lead to a smaller *SDbw* and less clustered groups. Reason is that compactness and separation are estimated higher due to the less amount of data points.
- **Size of groups/Significance level:** The larger the significance level and the larger the size of groups the lower the *SDbw* the less groups are clustered. The larger the significance level and the smaller the size of groups the larger the *SDbw* and the more groups are clustered. Nonetheless, the *SDbw* and consequently the homogeneity does not worsen or improve a lot. By changing the significance level from 0.05 to 0.9, an improvement of approximately $\frac{0.6935-0.5550}{2-0} = 7\%$ (figures as highlighted in Fig. 7 divided by the maximum *SDbw* = 2) of the *SDbw* is possible. We propose a clustering with a significance level of 0.05 to achieve homogeneous groups with a sufficient size of groups for the application to SPC.
- **Spread/Location:** Groups with a small spread and with wider distanced locations lead to a smaller *SDbw*. Close distanced locations and wide spreads lead to clustering of groups. Therefore, the algorithms work better if location and spread are separable.

7 Application of grouping algorithms to Statistical Process Control

The usual application of control charts consists of phase I and phase II. Phase I has the objective to provide valid control limits by calculating them with data under statistical control. Therefore, the process owner collects data, which

the process owner analyzes by constructing trial control limits to scrutinize if the process was in control while the data was gathered. If not, the process owner removes the assignable causes and repeats the collection of data until control limits with data under statistical control can be calculated. Phase II follows after achieving the objective of phase I. It uses the control chart to monitor the process and maintain its capability [13]. The usage of the grouping algorithms is during phase I. For every removal of an assignable cause and repeated calculation of control limits, the features must be regrouped to identify if the groups still exist after the assignable cause was removed or when the process is in control. This ensures that the groups and control limits are based on valid data and do not happen to be blurred by instabilities like a change of location or spread.

In order to test the applicability of the grouping algorithms and to investigate, whether we can group homogeneous features in small series production, we apply the grouping algorithms to data of turned parts. The data was generated within the framework of the research project “GriPS: Software-supported grouping of similar characteristics for process control in small quantities” and “APProVe: App-based effort reduction for adaptive inspection in variant production” at a participating company. One of the companies produces turned parts in small series with different dimensions. One of the features of interest is the inner diameter. Table 2 shows an excerpt of the provided anonymized data, which we applied to the grouping algorithms. The data is for a phase I application of SPC.

In Fig. 9, the data of Table 2 is once applied to a “Variable aim, individual and moving range chart” proposed by ISO 7870-8 without grouping. The control chart highlights the different groups by using different colors. The data is instable as it belongs to phase I of SPC application. Nonetheless, it is visible that we cannot group all variants.

In Fig. 10, we exemplarily applied the data to two different algorithms using a *p* value of 0.05 to show the performance differences of the algorithms. We achieve four groups with the algorithm based on cluster analysis and splitting. We achieve three groups when using the

Table 2 Excerpt of the historical real-world data from an application scenario with the relevant factors for grouping being “Variant” and “Deviation”

Variant	Feature	Nominal value	Measurement	Deviation	Upper tolerance	Lower tolerance
Variant1	Diameter	79.91	79.914	0.004	0.03	−0.03
Variant1	Diameter	79.91	79.916	0.006	0.03	−0.03
Variant2	Diameter	88.85	88.852	0.002	0.03	−0.03
Variant2	Diameter	88.85	88.859	0.009	0.03	−0.03
Variant3	Diameter	58.71	58.711	0.001	0.03	−0.03
Variant3	Diameter	58.71	58.713	0.003	0.03	−0.03

Data was provided by a company participating in APProVe and has been anonymized

Fig. 9 Application of the data without grouping to a “Variable aim, individual and moving range chart” proposed by ISO 7870-8. Different colors highlight the variants within the control charts

algorithm based on ChiMerge and splitting. Differences in the mean and variance of the variants’ deviation indicate that the solution with four groups and therefore the algorithm based on cluster analysis and splitting is preferable. This is compliant with the simulation results in chapter 5.

In Fig. 11, we show the effect of how different p values lead to different amount of groups on the example of the algorithm based on cluster analysis and splitting. As visible, the algorithm creates more groups the higher the p value (0.001 vs. 0.05). Therefore, the homogeneity of the groups is controllable by the p value. To achieve sufficiently homogenous groups a p value of 0.05 is proposed as shown by the simulation.

The data of the process depicted in Figs. 9, 10 and 11 is for a phase I application of SPC and not stable. Within this use case, using the grouping algorithm the process owner could assign some of the special causes to an unsuitable clamping of some variants. In the next step, the process owner has to apply the grouping algorithm again and revise the formed groups after changing the clamping. The process owner has to collect data and regroup the variants using the grouping algorithms until the stability of the process is proven.

8 Discussion

In this study, we presented grouping algorithms to combine samples from different similar production processes. We analyzed what statistical properties the grouping algorithms possess and if they enable the application of SPC to low volume production processes.

Mainly two different approaches exist to enlarge the sample size for SPC. The approaches by Wiederhold and Al-Salti et al.—contrary to our approach—require a lot of knowledge and work force [11]. Experts either need to decide upon which data of which variant can be combined or at least, which factors are important to consider when grouping similar features of variants, e.g. if the variants are

made of the same material. This also requires costly data preprocessing. Both, knowledge and work force are often not available in small and medium sized enterprises.

Our approach necessitates little data preprocessing and little expert knowledge, as our grouping algorithms only require historical measurement data of features and a feature category, which shows if features belong to the same entity, e.g. a diameter. The required database and database management is therefore significantly smaller than the one proposed by Zhu et al. Zhu et al. propose a database framework to apply SPC to low volume production based on GT [24]. Conceptual downside of our approach, compared to the other approaches, is, that our approach does not produce process knowledge in the sense that it can be easily comprehended what factors lead to new groups.

Contrary to the work of Al-Salti et al. and Zhu et al., we tested our grouping algorithms using a full-factorial DOE and simulation data as well as real production data. Al-Salti et al. only tested their approach on one real production data example [15]. Wiederhold tested his algorithm using a simulation and real production data as well, but did not use a full-factorial DOE [11]. Therefore, the quality of his algorithm and the groups it produces are not fully comprehensible yet. The results in the previous two sections show, that the grouping algorithms, and especially the algorithm based on cluster analysis and splitting successfully enlarge the sample sizes. Furthermore, we could show, by using real production data, that an application of SPC for low volume production in phase 1 is possible and allows a detection of special causes.

To analyze grouping algorithms further, works of Lin et al. are promising. Lin et al. analyze what ratio of standard deviations are allowable within grouped features. They use the simulations to find out which average run lengths control charts achieve with grouped data [25]. In future, a combination with their approach and grouping algorithms makes sense for further validating grouping results.

Fig. 10 Application of the data with grouping to a “Variable aim, individual and moving range chart” proposed by ISO 7870-8. The algorithm based on cluster analysis splitting and the algorithm based on ChiMerge and splitting created the groups at the pro-

posed p value of 0.05. Their different grouping results (four groups vs. three groups) are depicted in the figure. Different colors highlight the grouped variants within the control charts

Fig. 11 Effect of different p values on the amount of created groups. The higher the p value the more groups the grouping algorithms create. The effect is shown on the example of the algorithm

based on cluster analysis and splitting using the “Variable aim, individual and moving range chart” proposed by ISO 7870-8. Different colors highlight the grouped variants within the control charts

9 Summary and conclusion

From discretization methods, we deducted grouping properties and a grouping process, which can be used to develop further grouping algorithms. We developed four algorithms based on different statistical methods and tested them with a simulation and a DoE. We used the factors *amount of groups*, *size of groups*, *distribution*, *location* and *spread* to test the grouping algorithms for their suitability to find homogenous groups. To assess the performance of the algorithms, we used two indices: the *SDbw*, which assesses the separation and compactness of clustered groups. The index *t* assesses how many groups are clustered to larger groups. The algorithm based on cluster analysis and splitting proved best at successfully detecting homogeneous groups for an application to SPC. For clustering groups, it performs mediocre compared to the other algorithms according to the evaluation with *t*, but performs best in finding homogenous groups according to the evaluation with *SDbw*. We rate the homogeneity of groups as more important than the size of groups for successfully applying SPC. Otherwise alpha and beta errors occur when applying control charts leading to a poor average run length, which would incur high costs due to a needless failure analysis in practice. Furthermore, by scrutinizing interactions between the factors a significance level, we propose a significance level of 0.05 to find homogenous groups with a sufficient size of groups (Figs. 7, 8). Within an application example we showed the applicability of the groups to “Variable aim, individual and moving range chart” proposed by ISO 7870-8.

Acknowledgements Open Access funding provided by Projekt DEAL.

Funding The IGF-promotion plan 18723 N of the Research Community for Quality (FQS), August-Schanz-Straße 21A, 60433 Frankfurt/Main has been funded by the AiF within the program for sponsorship by Industrial Joint Research (IGF) of the German Federal Ministry of Economic Affairs and Energy based on an enactment of the German Parliament. The IGF-promotion plan 20618N of the Research Community for Quality (FQS), August-Schanz-Straße 21A, 60433 Frankfurt/Main has been funded by the AiF within the program for sponsorship by Industrial Joint Research (IGF) of the German Federal Ministry of Economic Affairs and Energy based on an enactment of the German Parliament.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate

if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Brecher C (2012) Integrative production technology for high-wage countries. Springer, Berlin
- Greipel J, Li Y, Schmitt RH (2017) Concept for clustering of similar quality features for optimization of processes in low volume manufacturing. In: Schmitt RH, Schuh G (eds) 7. WGP-Jahreskongress Aachen, 5.-6. Oktober 2017. Apprimus Wissenschaftsverlag, Aachen, pp 363–371
- Dietrich E, Schulze A (2014) Statistische Verfahren zur Maschinen- und Prozessqualifikation, 7., aktualisierte Auflage. Hanser, München
- Marques PA, Cardeira CB, Paranhos P et al (2015) Selection of the most suitable statistical process control approach for short production runs: a decision-model. IJET 5(4):303–310. <https://doi.org/10.7763/IJET.2015.V5.521>
- Hawkins DM, Qiu P, Kang CW (2018) The changepoint model for statistical process control. J Qual Technol 35(4):355–366. <https://doi.org/10.1080/00224065.2003.11980233>
- Elam ME (2008) Control charts for short production runs. In: Ruggeri F, Kenett RS, Faltin FW (eds) Encyclopedia of statistics in quality and reliability. Wiley, Chichester, pp 398–402
- Hawkins DM (1987) Self-starting cusum charts for location and scale. The Statistician 36(4):299–335. <https://doi.org/10.2307/2348827>
- Ledolter J, Swersey A (2018) An evaluation of pre-control. J Qual Technol 29(2):163–171. <https://doi.org/10.1080/00224065.1997.11979747>
- Permin E, Voigtmann C, Schmitt R (2018) A Bayesian approach to process model evaluation in short run SPC. Int J Eng Innov Res 7:92–97
- Foster K (1988) Implementing SPC in low volume manufacturing. ASQC Qual Congr Trans 42:261–267
- Wiederhold M (2017) Clustering of similar features for the application of statistical process control in small batch and job production, 1. Auflage. Ergebnisse aus der Produktionstechnik. Apprimus Verlag, Aachen
- Ryan TP (2011) Statistical methods for quality improvement. Wiley series in probability and statistics, 3rd edn. Wiley, Hoboken
- Montgomery DC (2009) Introduction to statistical quality control, 6th edn. Wiley, Hoboken
- Wiederhold M, Greipel J, Ottone R et al (2016) Clustering of similar processes for the application of statistical process control in small batch and job production. Int J Metrol Qual Eng 7(4):404. <https://doi.org/10.1051/ijmqe/2016018>
- Al-Salti M, Satham A (1994) The application of group technology concept for implementing SPC in small batch manufacture. Int J Qual Reliab Manag 11(4):64–76. <https://doi.org/10.1108/02656719410057962>
- Cheng CS (1989) Group technology and expert systems concepts applied to statistical process control in small-batch manufacturing. Arizona State University, Tempe
- ElMaraghy HA (ed) (2012) Enabling manufacturing competitiveness and economic sustainability: proceedings of the 4th international conference on changeable, agile, reconfigurable and

- virtual production (CARV2011), Montreal, Canada, 2–5 October 2011. Springer, Berlin
18. Opitz H (1970) A classification system to describe workpieces: definitions. Pergamon Press, Oxford
 19. ElMaraghy H, AlGeddawy T, Azab A (2008) Modelling evolution in manufacturing: a biological analogy. *CIRP Ann* 57(1):467–472. <https://doi.org/10.1016/j.cirp.2008.03.136>
 20. Natrella M (2010) NIST/SEMATECH e-handbook of statistical methods
 21. Liu H, Hussain F, Tan CL et al (2002) Discretization: an enabling technique. *Data Min Knowl Discov* 6(4):393–423. <https://doi.org/10.1023/A:1016304305535>
 22. Kaufmann H, Pape H (1996) Clusteranalyse. In: Fahrmeir L, Brachinger W (eds) *Multivariate statistische Verfahren*, 2., überarb. De Gruyter, Berlin
 23. Liu Y, Li Z, Xiong H et al. (2010–2010) Understanding of internal clustering validation measures. In: 2010 IEEE international conference on data mining. IEEE, pp 911–916
 24. Zhu YD, Wong YS, Lee KS (2007) Framework of a computer-aided short-run SPC planning system. *Int J Adv Manuf Technol* 34(3–4):362–377. <https://doi.org/10.1007/s00170-006-0610-7>
 25. Lin S-Y, Lai Y-J, Chang SI (1997) Short-run statistical process control: multicriteria part family formation. *Qual Reliab Eng Int* 13(1):9–24

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.