DEFORMATION CLUSTERING METHODS FOR TOPOLOGICALLY OPTIMIZED STRUCTURES UNDER CRASH LOAD BASED ON DISPLACEMENT TIME SERIES

Yasuyuki Shimizu[1], Nivesh Dommaraju[1], Mariusz Bujny[2], Stefan Menzel[2], Markus Olhofer[2], and Fabian Duddeck[1]

[1] Technical University of Munich, Munich, Germany
{yasuyuki.shimizu, nivesh.dommaraju, duddeck}@tum.de

[2] Honda Research Institute Europe GmbH, Offenbach/Main, Germany
{mariusz.bujny, stefan.menzel, markus.olhofer}@honda-ri.de

Key words: Deformation Behavior, Clustering, Time Series Data, Dynamic Time Warping, Topology Optimization, Crash Load

Abstract. *Multi-objective Topology Optimization has been receiving more and more attention in structural design recently. It attempts to maximize several performance objectives by redistributing the material in a design space for a given set of boundary conditions and constraints, yielding many Pareto-optimal solutions. However, the high number of solutions makes it difficult to identify preferred designs. Therefore, an automatic way of summarizing solutions is needed for selecting interesting designs according to certain criteria, such as crashworthiness, deformation, and stress state. One approach for summarization is to cluster similar designs and obtain design representatives based on a suitable metric. For example, with Euclidean distance of the objective functions as the metric, design groups with similar performance can be identified and only the representative designs from different clusters may be analyzed. However, previous research has not dealt with the deformation-related time-series data of structures with different topologies. Since the non-linear dynamic behavior of designs is important in various fields such as vehicular crashworthiness, a clustering method based on time-dependent behavior of structures is proposed here. To compare the time-series displacement data of selected nodes in the structure and to create similarity matrices of those datasets, euclidean metrics and Dynamic Time Warping (DTW) are introduced. This is combined with clustering techniques such as k-medoids and Ordering Points To Identify the Clustering Structure (OPTICS), and we investigate the use of unsupervised learning methods to identify and group similar designs using the time series of nodal displacement data. In the first part, we create simple time-series datasets using a mass-spring system to validate the proposed methods. Each dataset has predefined clusters of data with distinct behavior such as different periods or modes. Then, we demonstrate that the combination of metrics for comparison of time series (Euclidean and DTW) and the clustering method (k-medoids and OPTICS) can identify the clusters of similar behavior accurately. In the second part, we apply these methods to a more realistic, engineering dataset of nodal displacement time series describing the crash behavior of topologically-optimized designs. We identify similar structures and obtain representative designs from each cluster. This reveals that the suggested method is useful in analyzing dynamic crash behavior and supports the designers in selecting representative structures based on deformation data at the early stages of the design process.*

1   Introduction

Recent advancements in Computer-Aided Engineering (CAE) enable accurate and rapid simulations of dynamic structural deformation. Furthermore, Topology Optimization (TO) [4, 8, 17] can be used to obtain multiple designs of interest for a given engineering task. TO is a mathematical method that maximizes certain performance objectives by redistributing the material in a design space for a given set of boundary conditions and constraints. When multiple solutions are available, data mining methods are increasingly used for analyzing design performance and identifying *interesting* solutions.

TO or other generative design methods [13] yield multitudes of solutions if the constraints are not too restrictive or unknown at the initial stages. Based on experience or the application field, engineers identify certain criteria of interest such as crashworthiness or deformation behavior, and select a few prototypes for the next design iteration. When the number of designs is too high, it is challenging to review all the solutions manually and select a few *interesting* topologies. Machine learning methods can be used for data mining the datasets [12, 9, 10, 14]. For example, clustering methods such as Ordering Points To Identify the Clustering Structure (OPTICS) [3] can group similar designs and obtain representative prototypes in the dataset [15]. Furthermore, manifold learning methods such as UMAP [11] enable 2D visualization of the dataset, which can be used for design exploration [18]. However, for these unsupervised approaches, the metric used for comparing designs is critical. In various fields of industry, analyzing dynamic, nonlinear behavior is important, e.g., for crash simulations in the automobile industry. In the literature, few methods exist for comparing the deformation behavior of structures. Sible et al. [16] use spectral mesh processing to compare deformation time series but the method assumes isometric surface deformations. In this work, we investigate methods for comparing compare volumetric deformation. We monitor the deformation time series of one or more common nodes in the structures using Dynamic Time Warping (DTW) [5], a state-of-the-art method for comparing time series data. In this paper, we evaluate the usefulness of the metric proposed by us with the help of test datasets and clustering. The clustering quality is evaluated using methods such as Adjusted Mutual Information (AMI) [6] and silhouette score [2]. The first test case is simple time-series datasets of a mass-spring system with distinct behavior such as different periods or modes. The combination of DTW and OPTICS can identify the clusters of similar behavior accurately. The second test case is a dataset which contains the crash behavior of topologically-optimized designs. We identify similar structures and obtain representative designs from each cluster. In the remainder of the paper, we will discuss methods for comparing the deformation behavior of structures in Section 2. Then, Section 3 demonstrates the use of our proposed metric of deformation using test datasets. Section 4 discusses the conclusions of this work as well as the future research directions.

2   Comparing Deformation Behavior

Sible et al. [16] used spectral mesh processing to represent geometric shapes using a set of eigen coefficients. So, the deformation behavior of a structure can be observed by monitoring the evolution of the coefficient values over time. However, the Laplace-Beltrami operator used for spectral processing assumes isometric deformation and can only analyze the surface deformation.

Given these drawbacks, we propose a general method that can analyze the volumetric deformation while using the state-of-the-art metric called DTW of time series data. First, we identify *common nodes* between structures and construct the time series data for the nodal displacement. Then, we use metrics that compare the time series data to construct the dissimilarity measure of each deformation behavior.

Using this, we can cluster a set of design solutions and pick designs with typical deformation from each cluster. In the next section, different components used by the suggested method will be introduced.

2.1 Outline of the proposed method

Figure 1 shows the workflow of the proposed method. TO yields a set of input solutions, which are required to be analyzed based on deformation behavior. In the proposed method, we extract some representative nodes based on their displacement or the application. Later in this paper, we describe a method for choosing nodes based on displacement using clustering. When the TO results do not share all the nodes, only the *common nodes* are considered. After extracting the nodes, the displacement time history of the nodes is compared using Euclidean metrics or DTW. The proposed metric gives a so-called *distance matrix*, which contains the pair-wise distances between the designs based on the deformation behavior. Then, the clustering method uses this *distance matrix* as an input to cluster designs with similar deformation patterns. Finally, a design engineer may analyze the representatives from each cluster to select the preferred cluster. This process can be repeated until the user finds the desired design(s).
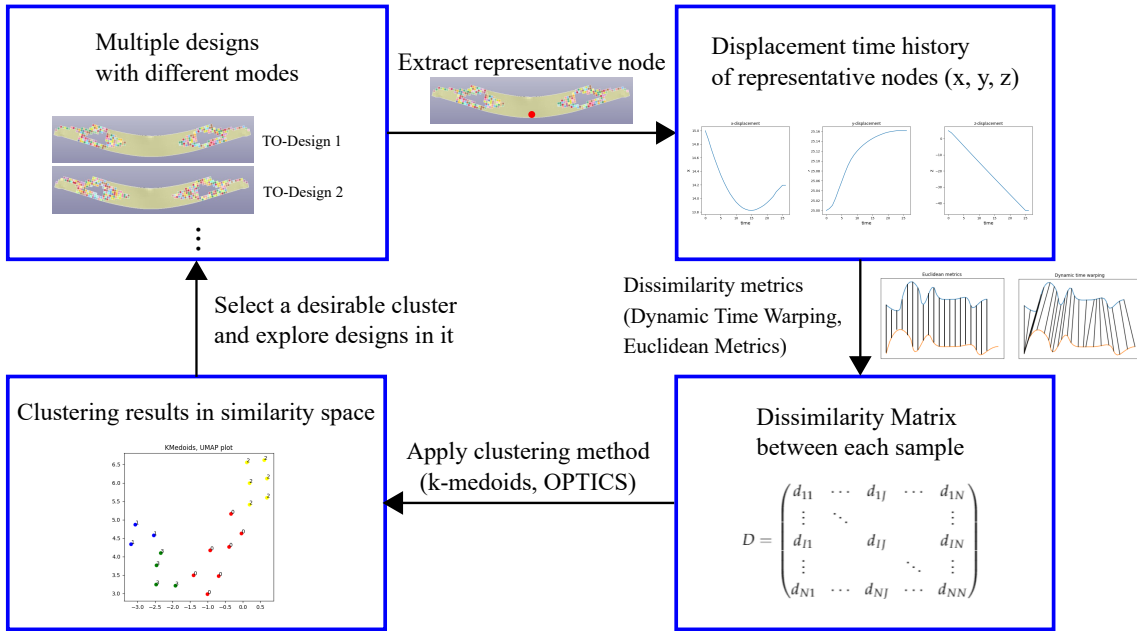

Figure 1: General workflow of the proposed method.

In the following, the different components of the workflow are explained.

2.2 Metrics for Comparing Time Series Data

As mentioned previously, we compare the deformation behavior by comparing the time series data of certain *common nodes* of the designs. Given N number of designs, to compare the designs $I, J \in \{1, ..., N\}$, we pick up the *common nodes* $n_1, n_2, ..., n_{end}$ of the given designs. For each design, displacement (**disp**) of the *common nodes* at time $t = t_0$ is expressed as a single *displacement vector*. For the

designs I and J, the corresponding *displacement vectors* are defined as follows:

$$\mathbf{disp}^I(t=t_0) = \begin{pmatrix} d_{n_1}^I(t=t_0) \\ d_{n_2}^I(t=t_0) \\ \vdots \\ d_{n_{end}}^I(t=t_0) \end{pmatrix}, \quad \mathbf{disp}^J(t=t_0) = \begin{pmatrix} d_{n_1}^J(t=t_0) \\ d_{n_2}^J(t=t_0) \\ \vdots \\ d_{n_{end}}^J(t=t_0) \end{pmatrix}, \tag{1}$$

where d_{n_i} is the displacement of the node n_i. The time series of the *displacement vector* of design I and J at $t = t_0, t_1, ..., T$ yields the following displacement matrix:

$$\mathbf{disp}^I = \begin{bmatrix} \begin{pmatrix} d_{n_1}^I(t=t_0) \\ d_{n_2}^I(t=t_0) \\ \vdots \\ d_{n_{end}}^I(t=t_0) \end{pmatrix}, \begin{pmatrix} d_{n_1}^I(t=t_1) \\ d_{n_2}^I(t=t_1) \\ \vdots \\ d_{n_{end}}^I(t=t_1) \end{pmatrix}, ..., \begin{pmatrix} d_{n_1}^I(t=T) \\ d_{n_2}^I(t=T) \\ \vdots \\ d_{n_{end}}^I(t=T) \end{pmatrix} \end{bmatrix}, \tag{2}$$

$$\mathbf{disp}^J = \begin{bmatrix} \begin{pmatrix} d_{n_1}^J(t=t_0) \\ d_{n_2}^J(t=t_0) \\ \vdots \\ d_{n_{end}}^J(t=t_0) \end{pmatrix}, \begin{pmatrix} d_{n_1}^J(t=t_1) \\ d_{n_2}^J(t=t_1) \\ \vdots \\ d_{n_{end}}^J(t=t_1) \end{pmatrix}, ..., \begin{pmatrix} d_{n_1}^J(t=T) \\ d_{n_2}^J(t=T) \\ \vdots \\ d_{n_{end}}^J(t=T) \end{pmatrix} \end{bmatrix}, \tag{3}$$

We compute the dissimilarity between these two time-series vectors as the *distance s_{IJ}*, which compares the deformation behavior of the two designs. For N number of designs, the pair-wise distances between the designs are given by the *distance matrix S*.

$$S = \begin{pmatrix} s_{11} & \cdots & s_{1J} & \cdots & s_{1N} \\ \vdots & \ddots & & & \vdots \\ s_{I1} & & s_{IJ} & & s_{IN} \\ \vdots & & & \ddots & \vdots \\ s_{N1} & \cdots & s_{NJ} & \cdots & s_{NN} \end{pmatrix} \tag{4}$$

where s_{IJ} is the distance between the displacement matrix of design I and J. To compute *distance matrix S*, we introduce two approaches: Euclidean metric and DTW which will be explained next.

2.2.1 Euclidean metric

The Euclidean metric between two data between $\mathbf{disp}^I$ and $\mathbf{disp}^J$ is given by

$$s_{IJ} = \sqrt{\frac{1}{n} \sum_{t=t_0}^{T} \{\mathbf{disp}^I(t) - \mathbf{disp}^J(t)\}^2} \tag{5}$$

where both the time series have n number of steps. While this is a simple metric, it cannot be used if the temporal data have a different number of steps. If the temporal data have a similar shape but are shifted in phase, the Euclidean metric identifies them as different.

2.2.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is designed to measure dissimilarity between two temporal sequences [5]. Unlike the Euclidean metric, DTW aligns the two given time-dependent sequences by warping them in a non-linear fashion to match each other. For 3D crash simulations, the temporal data is a sequence of 3D points in time and multi-dimensional DTW is used to measure the dissimilarity.

Figure 2 presents the difference between the Euclidean distance and DTW using a 1D temporal sequence. Although DTW is computationally more expensive than the Euclidean metric, it is often used in the field of pattern recognition in temporal data because it allows time series to have different lengths. Furthermore, DTW ignores the phase shift between time series with the same shape.
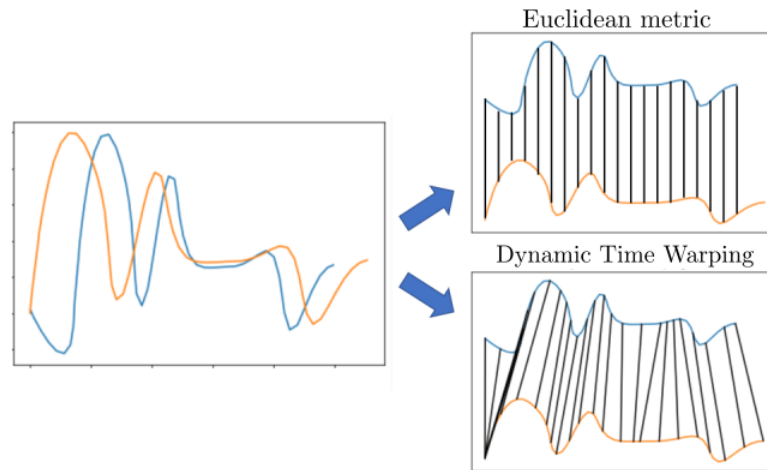

Figure 2: Comparison between Euclidean metric and DTW; the latter matches the points non-linearly.

2.3 Clustering method

After generating a *distance matrix*, clustering methods are used to identify designs with similar deformation behavior. In this paper, we use two clustering methods that can utilize the *distance matrix*: k-medoids and OPTICS. Since the k-means method [1] can only use the Euclidean metric, we do not use it in this paper.

2.3.1 k-medoids clustering

k-medoids method [7] is very similar to a popular clustering algorithm called the k-means method. k-means clustering tries to partition the given data into k number of clusters. Initially, k-means randomly identifies k data samples as the cluster centers. It then assigns the nearest samples for each center as a cluster, followed by an evaluation of the new cluster means; this process is repeated until the clusters do not change. In k-medoids, the medoids of the cluster samples are used instead of the mean of the cluster samples. The number of clusters k will be optimized using the mean silhouette score of the data samples for a given clustering [2]. Silhouette score of a sample i in cluster C is a measure of how similar the sample i is to other samples in C compared to other clusters.

2.3.2   Ordering Points To Identify the Clustering Structure (OPTICS)

OPTICS is a density-based algorithm, based on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [3]. However, unlike DBSCAN, OPTICS can identify clusters with different densities and can be used without much parameter tuning. Density-based clustering tries to connect samples in a high-density region into a cluster. OPTICS require the parameters ε, which describes the maximum radius to consider, and *MinPts*, being the number of minimal samples required to form a cluster. In our experiments, setting *MinPts* = 6 results in meaningful clusters.

2.4   Evaluation of Clusters

In real-world datasets, the expected clusters in the dataset are not known; otherwise, clustering is not needed. In this case, the silhouette score can evaluate how well separated the clusters are. A higher silhouette score indicates higher quality in clustering. The silhouette score ranges from -1 to 1. If the clusters are well-separated, the score is closer to 1. If the clusters are very close, the score is near 0. If they overlap strongly, the score is closer to -1.

For certain test datasets discussed later in this paper, the expected classes are known. In this case, Adjusted Mutual Information (AMI) [6] can be used to compare the similarity between the cluster labels and the expected classes/ground-truth labels. AMI score ranges from 0 to 1. If the clustering label matches correctly with the ground-truth labels, the score is 1. For random assignment of cluster labels, the AMI score would be 0.

Manifold learning techniques such as UMAP [11] can be used to visualize high-dimensional datasets, such as the time-series data considered in this paper, in 2D. Visualization of the dataset enables qualitative inspection of the clusters, especially if the classes/clusters in the dataset are known beforehand. Furthermore, visualization enables an intuitive understanding of the clusters present in the dataset.

3   Experiments on Test Datasets

In this section, we apply our proposed method to the datasets of two different experiments. In the first experiment, we cluster the motion of a spring-mass system with two masses, each with a degree of freedom along the springs. In the second experiment, we generate TO results under static and dynamic load conditions and classify them using the displacement pattern of selected nodes with our proposed metric. The clusterings obtained using Euclidean metric and OPTICS are compared using the silhouette score, in general, and additionally with AMI, when the expected classes are also known.

3.1   Clustering for 2 spring-mass system

Two DOF spring-mass system is designed to yield test datasets with well-defined clusters, i.e., the ground-truth labels/classes are known beforehand. Figure 3 shows four different modes of the system. By adding noise to these modes, four different classes of motion are generated. Mass values are $m_1 = m_2 = 1$ and the spring constants are $k_1 = k_2 = k_3 = 1$. Each mode has three samples, and each sample's starting positions are initialized with noise. Therefore, there are a total of 12 samples with 4 classes. For clustering, the *distance matrix* with the time series of two mass points (m_1 and m_2) is calculated as discussed in Section 2.2.

Figure 4 shows the displacement curves of the two masses; the curves are colored according to the cluster labels. Here, OPTICS clustering is used with DTW as the metric, which accurately identifies
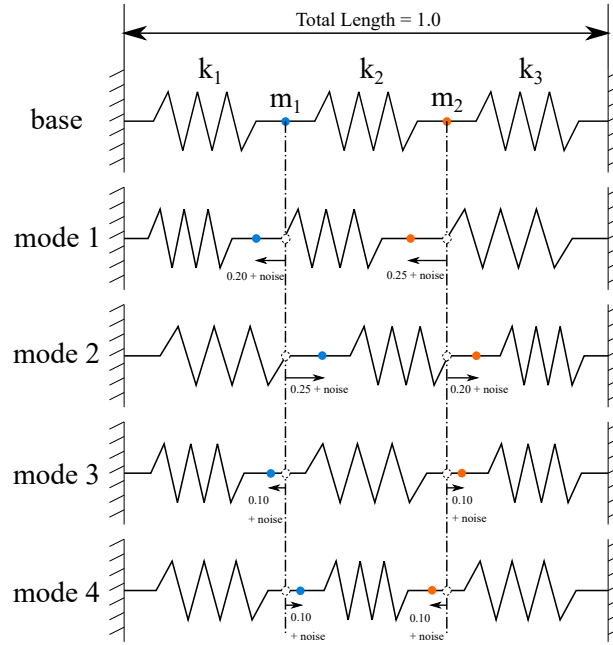
Figure 3: 4 different modes of spring behavior at t=0.

the four different classes. Table 1 shows the AMI scores and silhouette scores of each combination of dissimilarity metric and clustering method. Since the *k*-medoids method only identifies local optima, it is repeated a few times and only the best score is reported. OPTICS always accurately identifies the classes irrespective of the metric used. When *k*-medoids clustering is used, the AMI score is always less than 1. Similarly, silhouette scores with the OPTICS method are always nearly 1 and are higher compared to those of the *k*-medoids method.
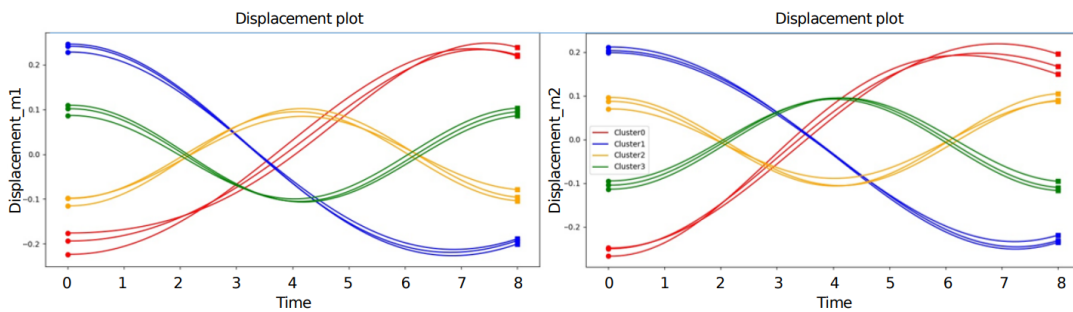

Figure 4: Time-displacement plot of mode analysis. Separated DTW + OPTICS is used as a clustering method. The color is according to the cluster label.

Next, we evaluate the metrics to detect the delay in time series data. The following delays in the initial position for mass 1 are used: 0 s (no delay), 4 s, 8 s, and 12 s (see Figure 5), which result in four different classes; for each class, three samples are generated using noise. For this experiment, the

Table 1: AMI and silhouette score of each combination (distance metric + clustering method) for detecting different modes.

Detection of different mode	AMI score	Silhouette score
DTW+OPTICS	1.0	0.943
Euclidean+OPTICS	1.0	0.951
DTW+k-medoids (best score)	1.0	0.943
Euclidean+k-medoids (best score)	1.0	0.951

Euclidean metric yields the best AMI score (1.0) and silhouette score (0.952). DTW yields a lower AMI score ($\in [0.40, 0.42]$) irrespective of the clustering method used. This is expected since DTW ignores the phase shift which may or may not be desired. If the application desires sensitivity to phase shift, use the Euclidean metric; otherwise we recommend using DTW.
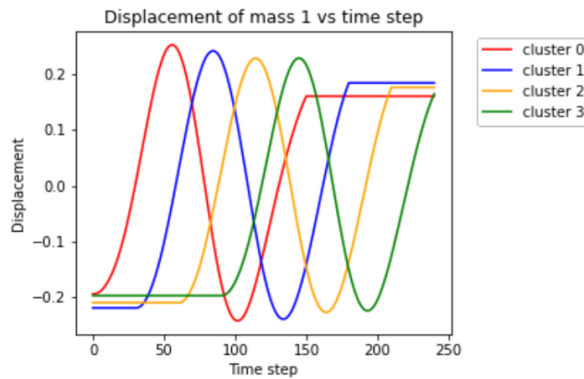

Figure 5: Different types of delayed time series. Only displacement of mass 1 is shown.

3.2 Topologically-optimized beam structures with crash load

In this section, a simply supported beam is optimized for a crash load and a static load, as described in [15]. Figure 6 shows the design space of dimensions 600 mm × 50 mm × 50 mm, which is split into 12000 solid elements. The structure is optimized to minimize the mechanical compliance for lateral static loads, 1000 N magnitude, and to maximize the energy absorption for the crash load from the top, which is defined by a prescribed velocity of 1000 mm/s. Bilinear elasto-plastic model (MAT_024 in LS-DYNA) of an aluminum material is used with the following properties: maximum mass density $\rho_{max} = 2.7 \times 10^3$ kg/m^3, Young's modulus $E = 70$ GPa, Poisson's ratio $\nu = 0.33$, yield strength $\sigma_Y = 117$ MPa, and hardening modulus $E_{tan} = 49$ GPa. For a given set of weights, we use a multi-objective TO method called SEW-HCA [8] with a maximum of 25 iterations, an allowed volume fraction of 0.4, a move limit of 0.1, and a penalization factor of 3.

Since the objectives are conflicting, we obtain non-dominated solutions (Pareto-optimal solutions) as shown in Figure 7. The figure also shows a sample from each cluster obtained using k-medoids and DTW metric. To obtain this clustering, we first select sample nodes in the *common nodes* in TO results on the Pareto front. The *common nodes* are partitioned into three groups using k-means clustering with
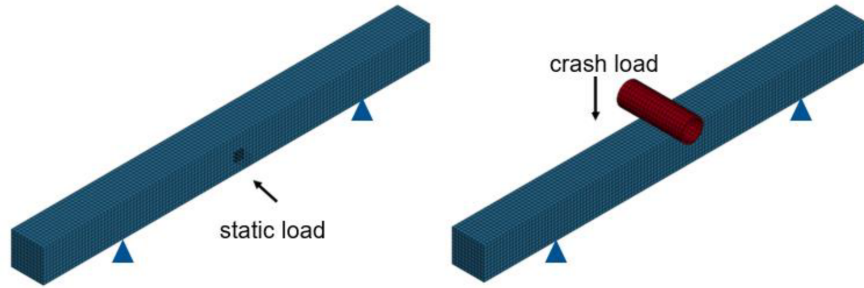
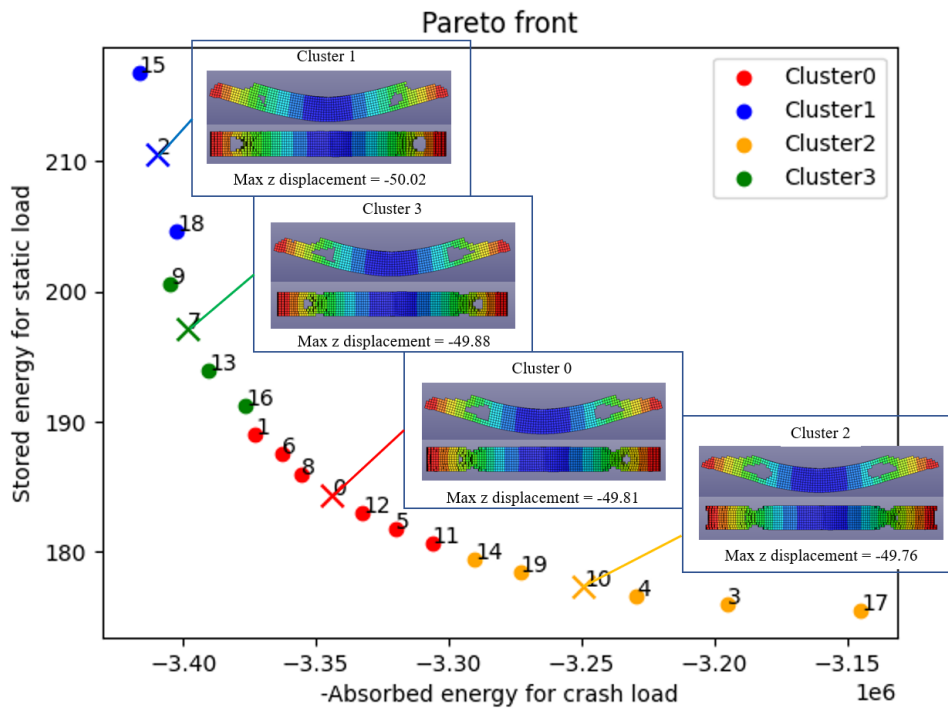Figure 6: Boundary conditions for optimization of a simply-supported beam for two separate loads [15].

Figure 7: Pareto front of optimized design set clustered using DTW + *k*-medoids. Cluster number for *k*-medoids is set to 3. The color is according to the cluster label. Medoids (representative designs) are shown.

displacement magnitude of nodes as input (see Figure 8). The medoids of the clusters are chosen as the selected nodes, whose displacement matrices are compared using DTW or Euclidean metrics.

Figure 7 shows the Pareto front of designs colored according to the clusters in Figure 9a. Note that the clusters based on deformation are not split in the objective space, which indicates that the structure with similar performance has similar deformation behavior for this dataset.

Figure 9a, 9b shows the result of clustering using the *k*-medoid method and the OPTICS method separately. In both cases, DTW is used as the metric. UMAP uses the corresponding pair-wise *distance*
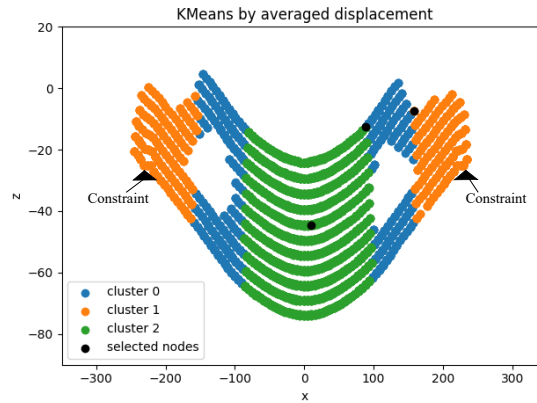
9

Figure 8: Common nodes among the TO results and the selected nodes based on clustering with displacement magnitude as input. Nodes are colored according to their cluster labels and a node is selected from each cluster (black point). Green nodes have the largest average displacement, and orange nodes have the least average displacement.

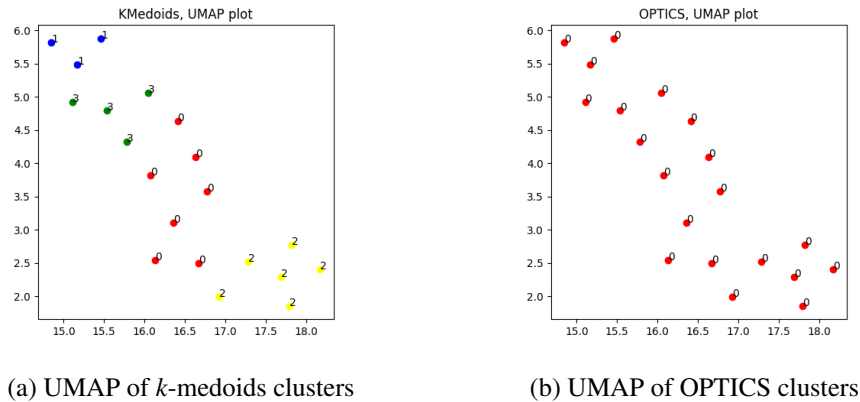(a) UMAP of k-medoids clusters

(b) UMAP of OPTICS clusters

Figure 9: Clustered design points in the similarity space

matrix to yield a 2D similarity space. It shows that clusters are kept intact in the UMAP space. These datasets do not form distinct clusters in terms of their density. In the k-medoids result, the clusters are contiguous, but they are well partitioned. On the other hand, if OPTICS is used as the clustering method, all designs are classified as belonging to the same cluster, because there is no obvious density deviation.

Since this dataset has no predefined labels, AMI cannot be used, but the silhouette score can be used to evaluate the clustering. Since OPTICS yields a single cluster, the silhouette score cannot be used. So, we can only compare the clustering obtained using k-medoids and the two metrics: Euclidean metric and DTW. DTW yields a slightly higher silhouette score of 0.55 compared to that of the Euclidean metric (the score is 0.54). The choice between the two metrics depends on whether phase shift and time warping

is needed, which depends on the application.

4 Discussion

In this paper, we proposed a method for clustering TO results based on the structural behavior. The proposed metric uses the time series data of nodal displacement for selected *common nodes* in the TO results. Furthermore, we used state-of-the-art metrics, namely DTW and Euclidean metrics, to compare the temporal data. The proposed metric of deformation behavior was analyzed using test datasets which include a mass-spring system, and a simply-supported beam optimized using a multi-objective TO called SEW-HCA [8, 15].

The nodes of interest are selected using a novel approach of clustering based on displacement magnitude. Based on our experiments, we conclude that the choice between the Euclidean metric and DTW depends on whether the phase difference between temporal data needs to be considered or not, which depends on the application. In the first experiment with a spring-mass system, both Euclidean and DTW metrics result in the best AMI and silhouette score, which validates their use to identify simple deformation modes. However, when there is a phase shift in the data, Euclidean distance is more sensitive than DTW; whether this is desirable or not depends on the application. Furthermore, we compared *k*-medoids and OPTICS clustering methods. From our experiments, we recommend the use of OPTICS to identify *natural* clusters, as in the spring-mass dataset. This is validated by the high AMI and silhouette score obtained with OPTICS clustering for the first dataset. When OPTICS yields too many outliers or a single cluster, we prefer using *k*-medoids, a partitioning algorithm, as in the case of the TO dataset.

The method used to identify representative nodes needs to be evaluated with more engineering examples and be improved if needed. Unlike the methods such as spectral mesh processing [16], the proposed method can handle volumetric deformation but may be affected by the high dimensionality if the number of selected nodes is high. So, dimensionality reduction techniques such as Principal Component Analysis or Artificial Neural Networks may be used along with the proposed metric. Additionally, in this paper, only nodes which existed after Topology Optimization were considered. If nodes that are eliminated in some designs remain in other designs and have apparent characteristic behavior, we need to have methods that consider those nodes' effect. Interpolation or extrapolation of eliminated nodes may be used to keep those nodes and their behavior.

In this work, we identified structures with similar deformation behavior and obtained representative designs from each cluster. Extensive experiments using simple deformation modes as well as a crashworthiness optimization example validate the proposed metric of deformation behavior. This reveals that the suggested method has the potential to analyze dynamic crash behavior and supports the designers in selecting representative structures based on deformation data at the early stages of the design process.

References

[1] James MacQueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[2] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.

[3] Mihael Ankerst et al. "OPTICS: Ordering points to identify the clustering structure". In: *Proc. ACM SIGMOD'99 Int. Conf. on Management of Data, Philadelphia PA, 1999*. Vol. 28. 2. 1999, pp. 49–60. DOI: 10.1145/304181.304187.

[4] Martin P. Bendsøe and Ole Sigmund. *Topology optimization*. Springer Berlin Heidelberg, 2004. DOI: 10.1007/978-3-662-05086-6.

[5] Meinard Müller. *Dynamic Time Warping*. Springer Berlin Heidelberg, 2007, pp. 69–84. DOI: 10.1007/978-3-540-74048-3_4.

[6] Nguyen Xuan Vinh, Julien Epps, and James Bailey. "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance". In: *J. Mach. Learn. Res.* 11 (Dec. 2010), pp. 2837–2854.

[7] Christian Bauckhage. *NumPy / SciPy Recipes for Data Science: k-Medoids Clustering*. Feb. 2015. DOI: 10.13140/2.1.4453.2009. URL: https://www.researchgate.net/publication/272351873_NumPy_SciPy_Recipes_for_Data_Science_k-Medoids_Clustering (visited on 05/01/2020).

[8] Nikola Aulig et al. "Preference-based topology optimization for vehicle concept design with concurrent static and crash load cases". In: *Structural and Multidisciplinary Optimization* 57.1 (2018), pp. 251–266. DOI: 10.1007/s00158-017-1751-z.

[9] C. Diez et al. "Big-Data Based Rule-Finding for Analysis of Crash Simulations". In: *Advances in Structural and Multidisciplinary Optimization*. June. 2018, pp. 396–410. DOI: 10.1007/978-3-319-67988-4_31.

[10] Justin Matejka et al. "Dream Lens". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, pp. 1–12. DOI: 10.1145/3173574.3173943.

[11] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861.

[12] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly UK Ltd., Oct. 2019. 819 pp. ISBN: 9781492032649.

[13] Sangeun Oh et al. "Deep Generative Design: Integration of Topology Optimization and Generative Models". In: *Journal of Mechanical Design* 141.11 (2019). DOI: 10.1115/1.4044229.

[14] Yuki Sato et al. "Data mining based on clustering and association rule analysis for knowledge discovery in multiobjective topology optimization". In: *Expert Systems with Applications* 119 (2019), pp. 247–261. DOI: 10.1016/j.eswa.2018.10.047.

[15] Nivesh Dommaraju et al. "Simultaneous Exploration of Geometric Features and Performance in Design Optimization". In: *16th International LS-DYNA Users Conference*. 2020.

[16] Sible Skylar et al. "A Compact Spectral Descriptor for Shape Deformations". In: *European Conference on Artificial Intelligence*. Vol. 325. IOS Press, 2020, pp. 1930–1937. DOI: 10.3233/FAIA200311.

[17] Mariusz Bujny et al. "Topology Optimization of 3D-printed joints under crash loads using Evolutionary Algorithms". In: *Structural and Multidisciplinary Optimization* 64.6 (2021), pp. 4181–4206. DOI: 10.1007/s00158-021-03053-4.

[18] Nivesh Dommaraju et al. "Evaluation of geometric similarity metrics for structural clusters generated using topology optimization". In: *Applied Intelligence* (2022). DOI: 10.1007/s10489-022-03301-0.