Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Simultaneous Exploration of Geometric Features and Performance in Design Optimization

Nivesh Dommaraju, Mariusz Bujny, Stefan Menzel, Markus Olhofer, Fabian Duddeck

2020

Preprint:

This is an accepted article published in 16th International LS-DYNA Conference 2020. The final authenticated version is available online at: https://doi.org/[DOI not available]

Simultaneous Exploration of Geometric Features and Performance in Design Optimization

Nivesh Dommaraju[1], Mariusz Bujny[2], Stefan Menzel[2], Markus Olhofer[2], Fabian Duddeck[1, 3]

[1]*Technical University of Munich, Munich, Germany*
[2]*Honda Research Institute Europe GmbH, Offenbach/Main, Germany*
[3]*Queen Mary University of London, London, UK*

Abstract

Topology optimization (TO) algorithms generate novel concepts to inspire and propel the design iteration process. LS-TaSC® is an industrial tool that implements TO algorithms and generates designs optimized for performance objectives such as maximum stiffness or energy absorption, under specified constraints e.g. allowed mass fraction of material in the design space. A multi-objective design exploration framework based on LS-Opt® and LS-TaSC® to generate designs is already available. The framework yields a Pareto set of designs by varying a parameter representing the relative preference of the user among the different objectives. A challenge persists as to how potentially large datasets of designs, generated using such an approach, can be reviewed efficiently by a designer. In this paper, we propose a method to identify a few representative design prototypes, which can be more easily reviewed by a designer. More concretely, the approach identifies classes of designs that look significantly different from a geometric point of view. For this purpose, we encode the information about the geometry using a voxel representation of the design. Subsequently, we use Principal Component Analysis (PCA), to reduce the high dimensionality of the representation, and extract features that encapsulate the geometric variation in the set of designs. Design prototypes are derived based on clustering algorithms using weights of principal components as features. To evaluate the proposed approach, we consider a solid beam model that is optimized for high

The authors gratefully acknowledge the financial support from Honda Research Institute Europe (HRI-EU) GmbH, Germany. We also thank Michael Gienger (HRI-EU) for his valuable comments.

stiffness under a static load case and high-energy absorption in a crash load case. Similar design problems are especially common in the car body design. We generate a Pareto set of designs for this test case and identify design prototypes. An interesting application of this method is to find designs with similar geometric appearance but very different performances. This can help us to estimate the robustness of a design. By helping in design exploration and selection, the proposed approach shows promise in large-scale industrial applications.

1 Introduction

Design optimization problems with multiple, possibly conflicting objectives, can result in a large set of feasible designs, called a Pareto-optimal set. The set consists of non-dominated solutions. This means that no other solution is better in all objectives when compared to such a solution. In structural optimization, Topology Optimization (TO) [1]–[6] is a promising tool to optimize the material layout. For structural problems with multiple load cases, methods exist to include user preferences regarding their relative importance [7], [8]. From the Pareto set, a few solutions can be selected for further review [9], as one cannot review all the designs due to limited time and computational resources. An exemplary selection method is any of the clustering techniques known in the field of machine learning.

In this work, we intend to use *geometric similarity* to select a few *representative* designs from the Pareto set. Quantifying the design similarity is a challenging problem that has been, e.g., the subject of research in the field of 3D object classification [10]. The challenges associated with this task are identifying a representation for the design, and an appropriate metric to quantify geometrical differences. We discuss here a data-driven method to construct so-called *similarity coordinates* for designs so that close distances in this coordinate space represent *geometric similarity*. Using the coordinates as features, a *representative* design is then picked from a group of design clusters in the Pareto front using *LS-Opt® and LS-TaSC®* frameworks.

As discussed, to generate *similarity coordinates*, we need to decide on a representation of the design and an appropriate metric to quantify geometrical differences. For this purpose, we use a simple voxel representation of designs. However, the high dimensionality of the representation makes it difficult to identify similar designs using the Euclidean metric. This is because of the so-called “curse of dimensionality”, a

phenomenon where for data with a large number of features, pairs of samples tend to have little differences if the Euclidean metric is used. This makes it difficult to find meaningful clusters in the data. To alleviate this problem, we can use a dimensionality reduction method such as Principal Component Analysis (PCA), a common technique used in data mining [11], to reduce the voxel representation to a much lower-dimensional representation. PCA linearly maps designs to new coordinates that reflect the variance in data.

In this paper, to illustrate our method, we use PCA to capture geometric differences. Other dimensionality reduction methods such as t-SNE [12], UMAP [13], Autoencoders [14] can be used instead of PCA, but the interpretation of the coordinates will change accordingly. The new coordinates, referred to as *similarity coordinates* in this paper, are used to find clusters of designs that are geometrically similar. Furthermore, design prototypes are found for each cluster.

In Section 2, we describe briefly a method to generate a Pareto front of designs by sampling load case preferences. Given a set of designs, we propose a method to identify prototypes based on geometric similarity in Section 4. The proposed method is demonstrated using a Pareto set of designs in Section 4. We conclude the discussion in Section 5.

2 Pareto front generation using load case preferences

Given boundary conditions and constraints for a finite element model, TO optimizes material layout for objectives such as structural compliance for static load cases, energy absorption for crash load cases [15]–[17]. Scaled Energy Weighting Hybrid Cellular Automata (SEW-HCA) [7] is a multi-disciplinary TO method that can concurrently optimize for multiple objectives. The method allows the specification of the relative user preference for each objective, with each choice leading to an optimized design. In this paper, SEW-HCA-inspired *SEW-LS-TaSC* is used to generate a Pareto front with two competing objectives: minimization of structural compliance for a static load case, and maximization of energy absorption for a crash load case.

In *LS-Opt*® 6.0 workflow (Figure 1), a stage is created to use *LS-TaSC*® 4.0, and generate designs using *SEW-LS-TaSC* as in [8]. Using a sampling method, different preferences for load cases are generated, with each preference leading to a design in

the Pareto set. The setting "D-Optimal" with a linear polynomial model is used for sampling user preferences each of which are used to generate a design in the stage "LSTASC". After this stage, objectives for the two load cases are calculated for each optimized design using *LS-Dyna solver* in stages "opt_LC1" and "opt_LC2" which completes the design generation and performance evaluation step. An additional stage "Prototyping", which includes our proposed method to identify design prototypes, is explained in the next section.

3 Prototype identification in Pareto set

The "Prototyping" stage, as shown in Figure 1, quantifies geometrical differences in Pareto set using a *similarity coordinate*, which is subsequently used for clustering designs to identify prototypes. This stage process each finite element model using a Python script which can be defined in the setting "User-Defined Postprocessor". In this setting, the command specifies the path to a Python binary followed by the script used to process each design. In this stage, for each finite element model, two user-defined responses, *similarity coordinate "s", and prototype flag "p"* are calculated. The *prototype flag* is calculated such that $p = 1$ for prototypes, and $p = 0$ otherwise. This calculation includes three major steps: (i) conversion of optimized finite element models to voxel representation; (ii) mapping voxel representation to *similarity coordinates*, a lower-dimensional representation; and (iii) clustering of designs based on *similarity coordinates* followed by prototype identification. The steps are described in detail as follows.

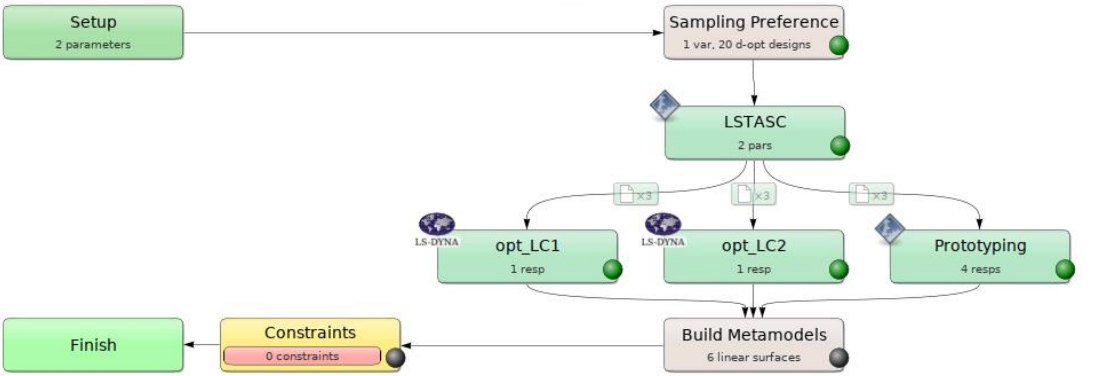

Figure 1: LS-Opt workflow with optimization in the stage "LSTASC", performance evaluation for crash and static load cases in stages "opt_LC1" and "opt_LC2" respectively, and

prototype identification in stage "Prototyping". The stages "Build Metamodels" and "Constraints" are not used in this work.

3.1 Voxelization of finite element models

Conversion of optimized finite element models to a *voxel representation* is an important pre-processing step in our method. Although voxels are a space-consuming representation, we use it here owing to its simplicity. For converting into voxels, we use a Python script to extract mesh information from *LS-Dyna*® keyword files. The voxel representation is a binary vector (a vector of zeros and ones), where each component, called *voxel* hereafter, is associated with an element in the finite element model. Elements with a minimum threshold density (relative density $> 1\%$ in this paper) are assigned a *voxel* value 1, otherwise with a value 0. The order of elements is not important as long as it is consistent across all finite element models. Since we know the location of each element, and hence the *voxel*, we can construct the geometry in 3D space from the voxel representation.

3.2 Similarity coordinates

As discussed previously, we convert the voxel representation to *similarity coordinates* using a dimensionality reduction technique. In this paper, we propose to use a simple dimensionality reduction method called Principal Component Analysis (PCA) [18], [19]. It linearly maps the voxel representation into a new coordinate system, with basis vectors called principal components, such that data has the highest variance in the first principal component. The variance decreases as one moves down the order of components. Since we use PCA on the voxel representation, each principal component can be interpreted as a design. One can represent any of the designs in the dataset as a linear combination of the principal components. The *similarity coordinates* are the weights of the corresponding principal components in the linear combination. For the design set considered in this paper, a single component is found to be sufficient to capture the geometric variation. Furthermore, it allows visualization of the Pareto front along with other objectives. For a more complex data set, more sophisticated data reduction techniques [12]–[14] could be explored.

3.3 Clustering and identifying prototypes

In machine learning, a subset of designs can be selected using clustering methods on one or more of the design features. For example, designs in the Pareto set can be

clustered using the similarity coordinate as a feature, leading to geometrically dissimilar prototypes. For clustering, we use K-Means, a simple and computationally cheap algorithm, but the expected number of clusters needs to be provided. In a Pareto front, where the clusters may not be clearly defined, K-Means is a convenient method to identify sub-regions. The medoid, a point with the least mean distance to other points within the cluster, serves as a prototype [14], [20] for each cluster.

4 Results

4.1 Test case generation

As a test case, we consider a simply supported beam of dimensions (600 mm x 50 mm x 50 mm) modeled using 12,000 solid elements. Using SEW-LS-TaSC, the model is optimized concurrently for static and crash load cases (Figure 2). The crash load is defined by a prescribed velocity of 1000 mm/sec, and the static load by a force of 1000 N. The goal is to minimize structural compliance for the static load and maximize energy absorption for the crash load. The optimization parameters used are a volume fraction of 0.3 and a move limit of 0.1. The method for sampling user preferences for each objective and generating the Pareto front is as discussed in Section 3.

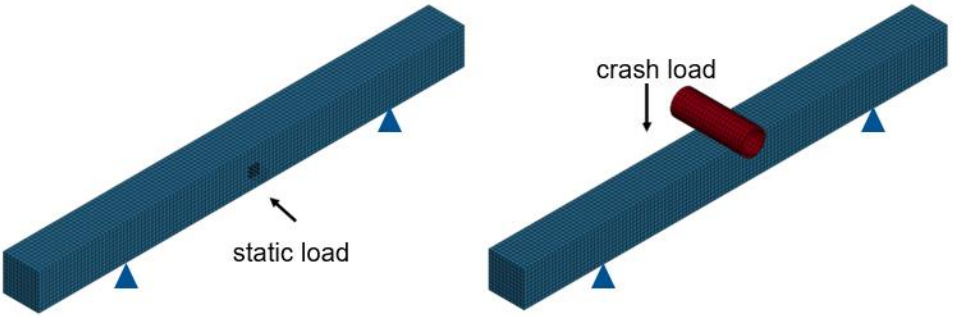

Figure 2: Static and crash load-case acting on a simply supported beam.

4.2 Pareto front and similarity coordinates

By sampling the relative preference for each of the two load cases, different designs are obtained. Using PCA, the voxel representation of each design is converted to a similarity coordinate, which is associated with the first principal component as described in Section 3.2. Figure *3* shows the Pareto front with absorbed energy for

crash load case (Ic1_IE), stored energy for static load case (Ic2_IE), and the similarity coordinate "s". Note that "s" changes monotonically as the user preferences change along the Pareto front. This indicates that geometrical features change gradually with the user preference. As expected, with improvement in energy absorption for the crash load case, there is an increase of structural compliance for the static load case.

4.3 Prototypes based on geometric similarity

Using K-Means as the clustering method, three clusters (Figure 4A) are identified based on "s", with prototypes found for each cluster. Prototype 3 is the thinnest structure capable of absorbing maximum energy (Ic1_IE) under crash load compared to other prototypes. As expected, Prototype 1 has the highest stiffness among prototypes. From a geometrical point of view, the edges become progressively thinner through prototypes 1-3 (see right-view in Figure 4A). The size of the hollow region, as seen in top-view, also increases. However, in the cross-sectional view, the size of the hollow part decreases. In all these geometrical aspects, prototype 2 falls in between prototypes 1 and 3. The voxel differences between prototypes are shown more clearly in Figure 4B/C. Comparing prototypes 1 and 2 (Figure 4B) indicate that prototype 2 has more material in the center and less material at the edges than prototype 1. The same is true when prototype 2 is compared with prototype 3 (Figure 4C).

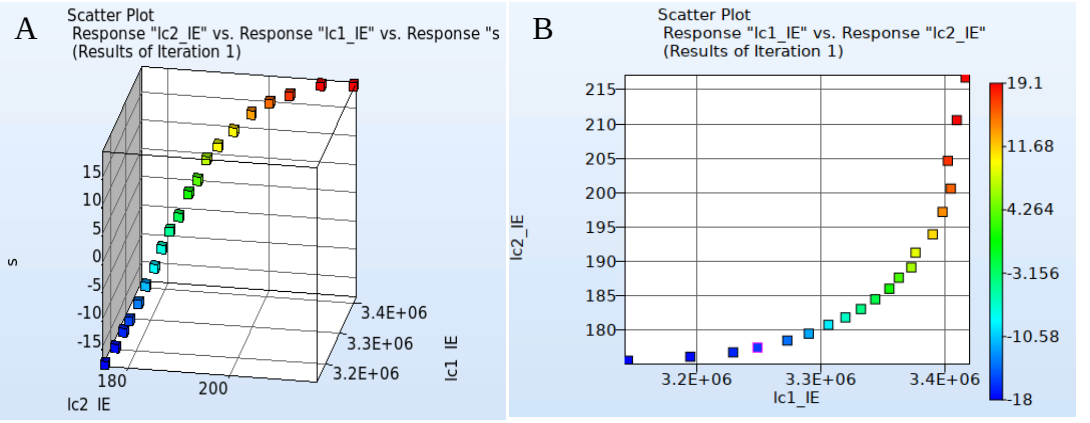

Figure 3: Objectives for the crash load (Ic1 IE) and the static load (Ic2 IE) vs. similarity measure. To aid visualization, data hue is based on "s"- similarity coordinate.

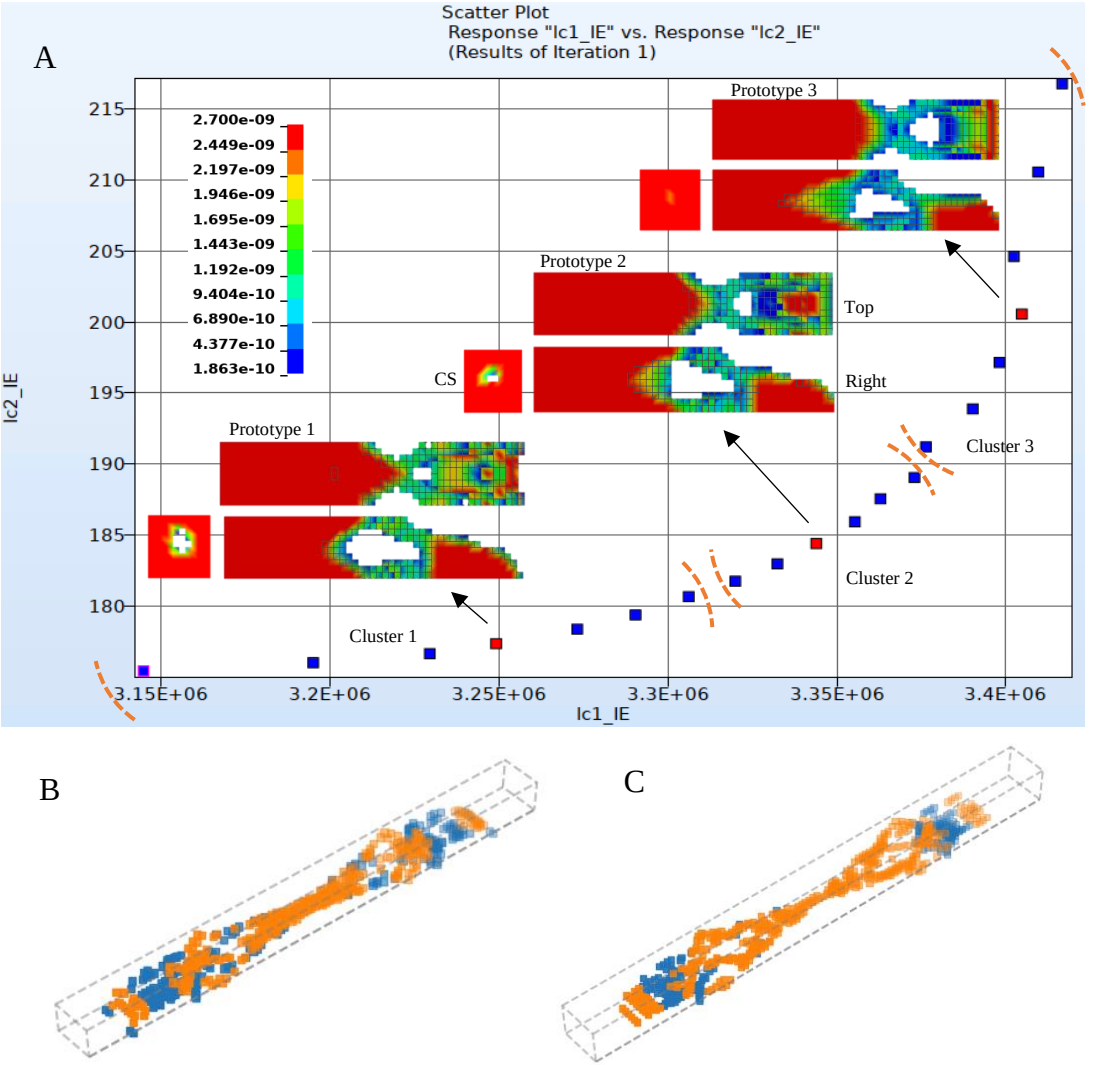

Figure 4: Prototypes (marked red on Pareto front) based on the similarity coordinate. A: Geometrical features of prototype 2 fall in between prototypes 1 and 3. Different views of the prototypes are shown with the density as hue: cross-section (CS), cut model-view (at mid-span) from top and right. The three design clusters are enclosed in dashed brackets. B: Difference between prototype 1 and 2. Blue voxels are present only in 1, not in 2; orange voxels are only in 2. C: Difference between prototype 2 and 3. Blue voxels are present only in 2; orange voxels are only in 3.

4.4 Interpretation of principal components

Each principal component, being a basis vector, has a unit magnitude. Unlike the voxel representation whose elements are either 0 or 1, elements in a principal component can have negative or fractional values. A possible way to interpret principal components is to find elements with large absolute values. These elements indicate regions of high variation across the design set. Using this method, the first two principal components of the Pareto set of designs are shown in Figure 5. The first component explains a variance of **51** % in the data; the second component explains **19** % of the variance. The similarity coordinate of a design is the magnitude of projection onto the first principal component. An alternative method to PCA decomposition is non-negative matrix factorization [21] which ensures that the components have non-negative values using a constraint. This can lead to an easier and possibly more sensible interpretation of components.

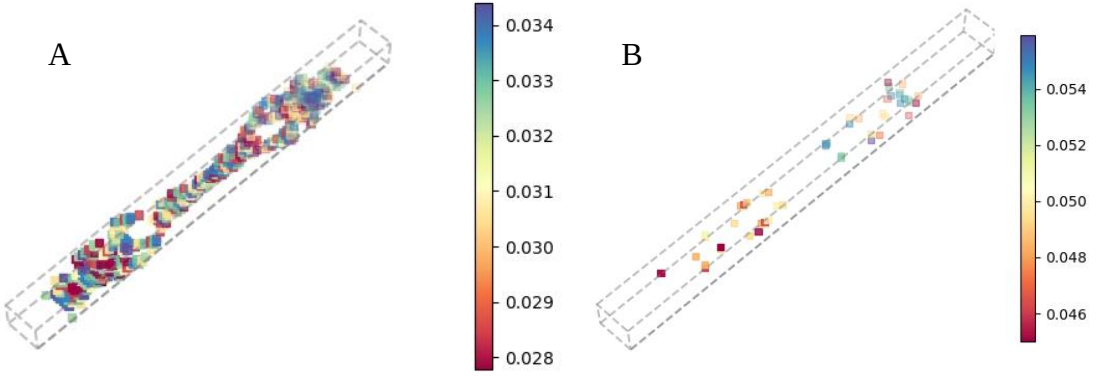

Figure 5: Voxel interpretation of principal components. Each element of the principal component is a voxel that is colored based on its magnitude. A: First principal component associated with the similarity coordinate ("s"). B: Second principal component.

4.5 Robustness analysis

Figure 6 shows the robustness of geometrical change with respect to the relative preference of load cases. The figure indicates that initially, "s" changes slowly with an increase in pref_1 (for values < 0.25); changes relatively quicker midway (when pref_1 is around 0.5); changes slowly again at the end (pref_1 > 0.75). This indicates that the geometry changes more prominently for changes in preferences, and hence for changes in objectives, at the central region of the Pareto front.

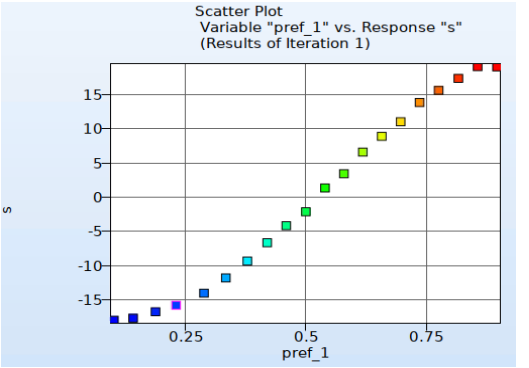

Figure 6: Sensitivity of similarity coordinate "s" on "pref_1" (user-preference of the crash case over the static case)

5 Conclusion and outlook

We presented a novel approach to automatically process the large number of designs generated by topology optimization algorithms. The method identifies the most important geometric variations in the designs and finds prototypes in clusters of similar designs. This gives additional insight into the designs and allows geometrical features to be treated as any other performance-based feature in design exploration, simplifying considerably the design process. In this paper, we use an *LS-Opt®* workflow to accomplish this task. Designs on a Pareto front are clustered based on *geometrical similarity*, with a prototype identified for each cluster. To encode *geometrical similarity*, we demonstrate the use of PCA on the voxel representation of finite element models. PCA can automatically identify important geometrical variances in a design data-set and encode it into a single similarity coordinate. In the future, it would be interesting to explore more complex design datasets, considering user preferences for geometrical features concurrently with other performance measures.

The voxelization process and similarity coordinate calculation could be computationally expensive. For the latter, to calculate similarity coordinate for a single design, all the other designs need to be considered as well. Furthermore, one obtains *similarity coordinates* for the other designs as well. It is inefficient to run this method for other designs again, which is what a stage in *LS-TaSC®* does by default. Therefore, we suggest *LS-TaSC®* to provide a step that only runs once per stage, instead of for

each test case. This could be relevant for other post-processing tasks that only need to run once for a set of designs.

6 References

[1] M. P. Bendsøe, "Optimal shape design as a material distribution problem," *Structural Optimization*, vol. 1, no. 4, pp. 193–202, 1989.

[2] J. Liu and Y. Ma, "A survey of manufacturing oriented topology optimization methods," *Advances in Engineering Software*, vol. 100, pp. 161–175, 2016.

[3] M. Bujny, N. Aulig, M. Olhofer, and F. Duddeck, "Evolutionary Level Set Method for Crashworthiness Topology Optimization," in *Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016)*, 2016, pp. 309–322.

[4] E. Raponi, M. Bujny, M. Olhofer, N. Aulig, S. Boria, and F. Duddeck, "Kriging-assisted topology optimization of crash structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 348, pp. 730–752, 2019.

[5] D. Zeng and F. Duddeck, "Improved hybrid cellular automata for crashworthiness optimization of thin-walled structures," *Structural and Multidisciplinary Optimization*, vol. 56, no. 1, pp. 101–115, 2017.

[6] F. Duddeck, S. Hunkeler, P. Lozano, E. Wehrle, and D. Zeng, "Topology optimization for crashworthiness of thin-walled structures under axial impact using hybrid cellular automata," *Structural and Multidisciplinary Optimization*, vol. 54, no. 3, pp. 415–428, 2016.

[7] N. Aulig, E. Nutwell, S. Menzel, and D. Detwiler, "Preference-based topology optimization for vehicle concept design with concurrent static and crash load cases," *Structural and Multidisciplinary Optimization*, vol. 57, no. 1, pp. 251–266, 2018.

[8] S. Ramnath, N. Aulig, M. Bujny, S. Menzel, I. Gandikota, and K. Horner, "Load Case Preference Patterns based on Parameterized Pareto-Optimal Vehicle Design Concept Optimization," in *European LS-DYNA Conference 2019*, 2019.

[9] J. C. Ferreira, C. M. Fonseca, and A. Gaspar-Cunha, "Methodology to select solutions from the pareto-optimal set: A comparative study," in *Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference*, 2007.

[10] S. Biasotti, D. Giorgi, S. Marini, M. Spagnuolo, and B. Falcidieno, "A comparison framework for 3D object classification methods," in *B. Gunsel, A.*

K. Jain, A. M. Tekalp, B. Sankur (eds) Multimedia Content Representation, Classification and Security. MRCS 2006. Lecture Notes in Computer Science, 2006, vol. 4105, pp. 314–321.

[11] D. Feldman, M. Schmidt, and C. Sohler, "Turning Big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2013, pp. 1434–1453.

[12] Laurens van der Maaten and H. Geoffrey E., "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[13] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[14] N. Dommaraju, M. Bujny, S. Menzel, M. Olhofer, and F. Duddeck, "Identifying topological prototypes using deep point cloud autoencoder networks," in *IEEE International Conference on Data Mining Workshops, ICDMW*, 2019, pp. 761–768.

[15] M. P. Bendsøe and O. Sigmund, *Topology Optimization*. 2004.

[16] A. Karrman and G. Allaire, "Structural optimization using sensitivity analysis and a level-set method, in Scilab and Matlab," *Techn. Rep. Ecole Polytechnique*, pp. 1–13, 2009.

[17] M. Bujny, N. Aulig, M. Olhofer, and F. Duddeck, "Identification of optimal topologies for crashworthiness with the evolutionary level set method," *International Journal of Crashworthiness*, vol. 23, no. 4, pp. 395–416, 2018.

[18] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[19] E. Ulu, R. Zhang, and L. B. Kara, "A data-driven investigation and estimation of optimal topologies under variable loading configurations," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, vol. 4, no. 2, pp. 61–72, 2016.

[20] A. Hagg, A. Asteroth, and T. Bäck, "Prototype Discovery using Quality-Diversity," *Computing Research Repository (CoRR)*, vol. abs/1807.0, 2018.

[21] Z. Bozakov, L. Graening, S. Hasler, H. Wersing, and S. Menzel, "Unsupervised extraction of design components for a 3D parts-based representation," in *Proceedings of the International Joint Conference on Neural Networks*, 2008, pp. 2009–2016.