Design Science

Complexity-driven layout exploration for aircraft structures

Jean-François Gamache[1], Aurelian Vadean[1], Mario Capo[1], Thomas Rochefort-Beaudoin[1], Nicolas Dodane[2] and Sofiane Achiche[1]

[1] *Department of Mechanical Engineering, Polytechnique Montreal, Montreal, QC, Canada*
[2] *Research & Technology Stelia Aéronautique Canada, Research & Technology, Mirabel, QC, Canada*

Abstract

Topology optimization has been identified as a powerful tool to improve aircraft structures for many years. Yet, innovative layouts have not been successfully implemented in commercial aircraft for several reasons. One reason identified by our research group is the lack of design constraints during topology optimization, such as buckling stability, which yields complex solutions that are not easily manufacturable. Second, the complexity of the resulting layouts makes integration with other systems highly challenging. With respect to these challenges, we propose a new heuristic layout optimization process: complexity-driven layout exploration for aircraft structures (CD-LEAS). The new process addresses the challenges of complexity and nonlinear constraints, such as buckling, in aircraft structure layout optimization. The novelty of CD-LEAS comes from the integration of a relative complexity metric as a driver to navigate the design space efficiently. Two case studies of commonly used stiffened panels are carried out to showcase the performance of the process. The results show that using complexity to navigate an explicit design space allows our process to quickly output a family of simple, light, stiff and buckling-resistant layouts.

Keywords: Aircraft Design, Layout Optimization, Generative Design, Complexity Metric, Axiomatic Design

Received 18 June 2021
Revised 30 April 2023
Accepted 02 May 2023

Corresponding author
Jean-François Gamache
jean-francois.gamache@polymtl.ca

Des. Sci., vol. 9, e13
journals.cambridge.org/dsj
DOI: 10.1017/dsj.2023.12

1. Introduction

Reducing the weight of structures is an ongoing challenge of the aerospace industry, as structure weight is linked with environmental impact (Kaufmann, Zenkert, & Wennhage 2010).

Aircraft structures are traditionally built from stiffened panel assemblies (Bruhn 1973). The elementary structures have specific functions, such as the ribs, spars and skins which are then assembled to create sections, such as the fuselage, the wings and the cockpit (Bruhn 1973). A significant advantage of stiffened panels is their easy integration and assembly with other systems of the aircraft.

Stiffened panels are built from two sets of components, the skin and the stiffeners, see Figure 1. Functionally, the skin is sized for structural stiffness and the stiffeners are used to maintain the structure's stability and increase local buckling resistance (Bruhn 1973). The design of stiffened panels is a complex multidisciplinary activity. Its design can impact the aerodynamic, vibration, controls, systems and so forth. For structural engineers, the main challenge is the study

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

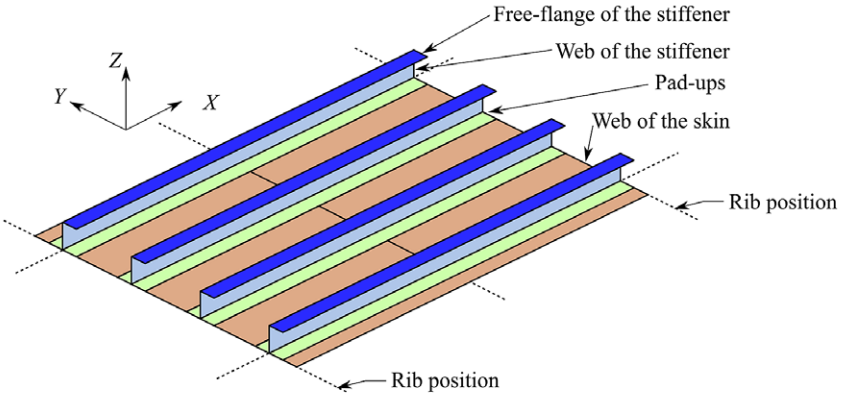

Figure 1. Subcomponents of the stiffened panel, orthogrid layout illustrated.

of mechanical failures, thus ensuring the panels can withstand heavy loads throughout the lifecycle of the aircraft. Some of the mechanical failure causes are local, such as fracture, rupture, collapse, local buckling, and post-buckling instability (Megson 2017). Other failure causes are global, for example, flutter or global buckling (Megson 2017). State-of-the-art design methods for stiffened panels are based on handbook relations to address most mechanical failure causes (Bruhn 1973). The solutions proposed in this paper are in regard to the challenge of creating panel layouts outside the traditional ones, in the hope of reducing the weight of aircraft structures (Gamache *et al.* 2021).

Stiffened panels are mainly built from orthogonally placed stiffeners in the orthogrid layout (Bedair 2009), which is the illustrated layout of Figure 1. There are very few new layouts used in aircraft design, as they are linked to an increase in analysis, validation and certification complexity. Additionally, there is only a limited amount of empirical data available for such layouts, meaning semi-empirical calculations are not usable for their analysis. One of the reasons for the success of the orthogrid layout is its ability to maintain global stability even when local skin segments are buckling; this phenomenon is referred to as post-buckling stability (Bruhn 1973; Gamache *et al.* 2019). Controlling it through sizing optimization allows the orthogrid stiffened panel weight to be reduced significantly.

Buckling containment features (BCF) and curvilinear stiffeners are some good examples of new layouts. BCFs were found to increase the strength-to-weight ratio of stiffened panels in certain regions of a commercial aircraft wing (Houston *et al.* 2016). BCFs and curvilinear stiffeners have been proposed using the experience and intuition of engineers and researchers to improve the design of stiffened panels (Kapania, Li, & Kapoor 2005; Houston *et al.* 2017). These layouts have generated traction due to advances in manufacturing and assembly techniques such as electron beam free-forming, additive manufacturing and integrally machined stiffeners (Kapania, Li, & Kapoor 2005; Caseiro 2013; Mulani, Slemp, & Kapania 2013). The topological design space opened up by these new manufacturing methods is large and complex to navigate. Consequently, traditional design and optimization tools are having trouble finding innovative ways to leverage the new layouts and manufacturing techniques.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

Topology optimization is an example of a numerical tool that attempts to find new lighter and innovative layouts (Zhu, Zhang, & Xia 2016; Aage *et al.* 2017; Tyflopoulos *et al.* 2018).

Topology optimization has been used in multiple academic and industrial research projects (Sigmund & Maute 2013; Deaton & Grandhi 2014). In current industrial practices, the solid isotropic material with penalization (SIMP) method is the most commonly used and is implemented in many commercial software (*Ansys*© Workbench 2020; Optistruct 2018). Older techniques include the ground-structure method (GSM) and the homogenization method (Bendsøe & Sigmund 2003). More recent topology optimization techniques are the level-set method (LSM) and the moving morphable component (MMC) (van Dijk *et al.* 2013; Guo, Zhang, & Zhong 2014).

In the aviation industry, SIMP has been used for the design of many aircraft components such as wings, fuselage and subsystems (Krog *et al.* 2004; Zhu *et al.* 2016; Aage *et al.* 2017). Still, no major projects have shown any improvement in the main structural component (Aage *et al.* 2017). In our research group, we have identified buckling load factor constraints as one of the main factors that impact the viability of stiffened panel optimization for both SIMP and GSM approaches (Gamache *et al.* 2020, 2021). So far, our research has only been able to show that topology optimization can reach viable results, when buckling is considered, if coupled with a random restart approach that yields an improved exploration of the design space (Gamache *et al.* 2020, 2021). Still, random restart approaches have shown to be highly inefficient as they tend to find only the minima with the largest "valleys," which are, at best, as efficient as the orthogrid layout (Gamache *et al.* 2020, 2021).

Another aspect we observed with the use of topology optimization in the industry is that the results tend to be highly complex geometries that cannot be manufactured without design simplification. However, these simplifications made by engineers tend to cut down any apparent weight gains made through topology optimization. It explains partially why it is so challenging to propagate optimized layouts throughout all the design phases. The effect of design complexity is shown well in the large-scale project carried out by the Technical University of Denmark (DTU) where the optimized material distribution sacrifices its apparent stiffness and weight advantages when interpreted to a manufacturable design (Aage *et al.* 2017). This effect has also been observed in previous work of our research group which was carried out on ribs, pressure panels and wing components with different topology optimization algorithms (Dugré, Vadean, & Chaussée 2016; Gamache *et al.* 2018, 2020, 2021). Making SIMP and GSM work for stiffened panels optimization required several weeks of fine-tuning simulation and optimization parameters towards results that can be ultimately transferred to detailed design and manufacturing (Gamache *et al.* 2020, 2021). This tuning is necessary, as not all load cases and mechanical failure causes can be included as topology optimization constraints, due to generation capabilities and limited computing power. This leads to a classic optimization issue, where an optimization solver will optimize out any considerations that cannot be included in the optimization formulation. This dichotomy between design and optimization makes optimization difficult to use for the design of structures, and is especially true for the more complex layout optimization problem.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

In summary, our previous work has not shown topology optimization to be ready for larger use cases, especially due to difficult tuning requirements, high computing costs and solutions' complexity (Gamache *et al.* 2020, 2021).

Current topology optimization algorithms and processes, when applied for layout optimization, have yet to efficiently find layouts lighter than conventional layouts. Considering the limitations discussed above, we have developed a new heuristic two-level optimization process specifically for stiffened panel layout optimization. There have been other works on two-level optimization using topology optimization, but none on stiffened panel layout or using complexity (Chirehdast & Papalambros 1992; Walbrun, Witzgall, & Wartzack 2019).

Topology optimization is an important computational tool that reassembles traditional design searches (Hay *et al.* 2017) and as such our new tool will be based on a design-centric approach. Our development approach is based on the generative design framework discussed in the first author's thesis (Gamache 2021). This framework distinguishes three properties of generative systems along three axes: generation, representation and navigation. These properties are defined as follows:

> **Generation** refers to the way the system infers the architecture of the solution from the description of the topological variables. Generative systems can range from **implicit** to **explicit** generative variables, where implicit variables describe a physical property of a specific volume while an explicit variable would describe the relationship between discrete components, see Figure 2.

> **Representation** refers to the degree of fidelity used for the performance evaluation of the generative system. A gradient exists between **low-fidelity** to **high-fidelity** simulation, as low-fidelity is less computationally expensive, with the compromise of less complicated models. Note that low-fidelity simulations can still be highly accurate, but require well-understood simulation hypotheses.

> **Navigation** refers to the axis between **exploiting** a small but promising region of the search space, and **exploring** through a larger region for possible better optimum (Gamache 2021). In optimization vocabulary, it refers to whether the process is looking for local or global optima.

The advantages of using the generative design framework are its effectiveness to identify trends and its common usage in the topology optimization community. First, we identified that the new research topics are moving from implicit (e.g.,

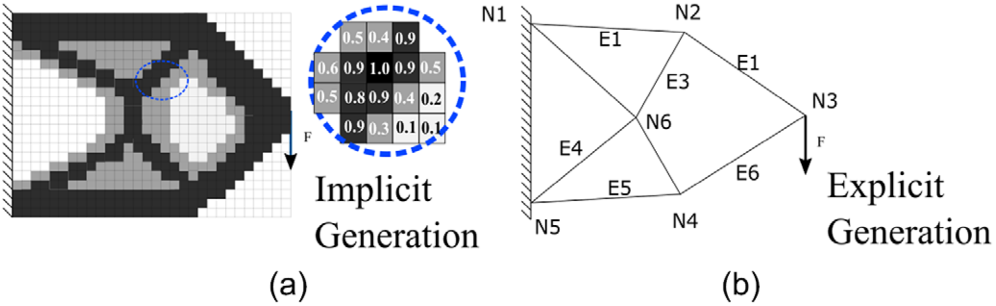

Figure 2. Illustration of implicit versus explicit generation. (a) The topology is defined by the density values of each element (as in SIMP). (b) The topology is defined from a graph, containing nodes (NX) and edges (EX), as used in the process presented in this work.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

SIMP) (Bendsøe & Sigmund 2003) towards a more explicit generation (e.g., moving morphable components) (Guo *et al.* 2014). Second, adding physics considerations such as stress and buckling, has been an ongoing issue since it translates into nonlinear constraints (Ferrari & Sigmund 2019). In turn, the nonlinear constraints induce feasible spaces that are more difficult to navigate, requiring even more computing power. Finally, we observe a major difference between academic work and industrial usage of topology optimization, where in academia topology optimization is presented as a tool to find an optimum in a convex design space, thus focusing on exploitation, while in the industry topology optimization is rather used to diversify solutions, thus focusing on exploration.

These observations led to the conclusion that the way topology optimization was carried out differently in academia and in industry can explain, in part, why its use for aircraft structure innovation has not been successful yet. The challenges of topology and layout optimization for aircraft structure are explored further in the next section.

2. Using complexity to drive the topological design space navigation

Following the trends identified in the generative design framework in stiffened panel topology optimization, we created a new process that addresses the layout optimization challenge. This process integrates design considerations during optimization and focuses on exploring simple, viable solutions while limiting the computing time. The key to efficiency is the new complexity objective that drives navigation and allows adding complexity to the design (layout) only if it is necessary. We named the new process after this key feature: complexity-driven layout exploration for aircraft structure (CD-LEAS).

Integrating a nonlinear constraint, such as buckling, requires a simulation model with more elements than ones aiming only for compliance but does not need an increase in design variables. In SIMP, the simulation model and optimization models use the same elements and as such, increasing representation fidelity also increases optimization cost with no added benefits. Consequently, to reduce computation cost, it is crucial to decouple the simulation and the optimization models. Decoupling a process depends on the type of generation used, and is made easier with more explicit approaches such as MMC (Guo *et al.* 2014).

Explicit generation creates a navigation challenge as the design space becomes disconnected. Therefore a disconnected space cannot be easily navigated by gradient-based optimization. To search in this design space, a heuristic optimization approach is better suited. However, it is known to be less efficient than gradient-based algorithms. This low efficiency is alleviated in CD-LEAS by using a novel complexity measure as an optimization objective, as it limits the search to low-complexity solutions. Intuitively, using complexity as an objective and starting with a low-complexity layout emulates the thinking process of an experienced engineer, which usually starts with simple solutions, and increases complexity only if it brings any benefits towards the design objective. In other words, the complexity objective modifies the search pattern to easily explore first the smaller low-complexity area before moving on to the larger high-complexity area of the design space. Moreover, using an explicit design space has the advantage of making significant topological changes between each optimization iterations, whereas

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

gradient-based solutions require a simulation of multiple solutions with the same layout, but with slightly different design parameters. These search patterns reduce significantly the number of inner-loop optimization runs. Using the complexity measure to infer direction to the navigation is the key element of the CD-LEAS process.

The CD-LEAS process combines know-how from many disciplines to achieve an effective design exploration, see Figure 3 for an illustration of the process. CD-LEAS can be described as a two-level optimization process, where layout/topology is optimized in the outer loop and sizing is optimized in the inner loop. The layout optimization formulation is detailed in the navigation section and the sizing optimization is in the representation section of this paper. Another advantage of the CD-LEAS process is that the explicit generation allows for an easy integration of the two optimization loops, rather than an approximation between the two loops as done in previous research work (Bremicker *et al.* 1991; Chirehdast & Papalambros 1992; Gamache *et al.* 2018).

For the generation, CD-LEAS uses a graph-grammar that can create many reinforcement layouts from actions and rules. Graph-grammars allow for a quick explicit generation of layouts that also ensures significant topological changes for each iteration. In this work, the generation is implemented in Matlab (MATLAB 2020).

As the generation is carried out using an explicit method, the choice of the fidelity of the representation remains flexible. The limitation here is the software and hardware available to the user. As such, any fidelity level can be used, from linear static to nonlinear geometric analysis. For the implementation discussed in this paper and the use case described, we opted for linear models for both compliance and buckling to keep computation time reasonable but simulation

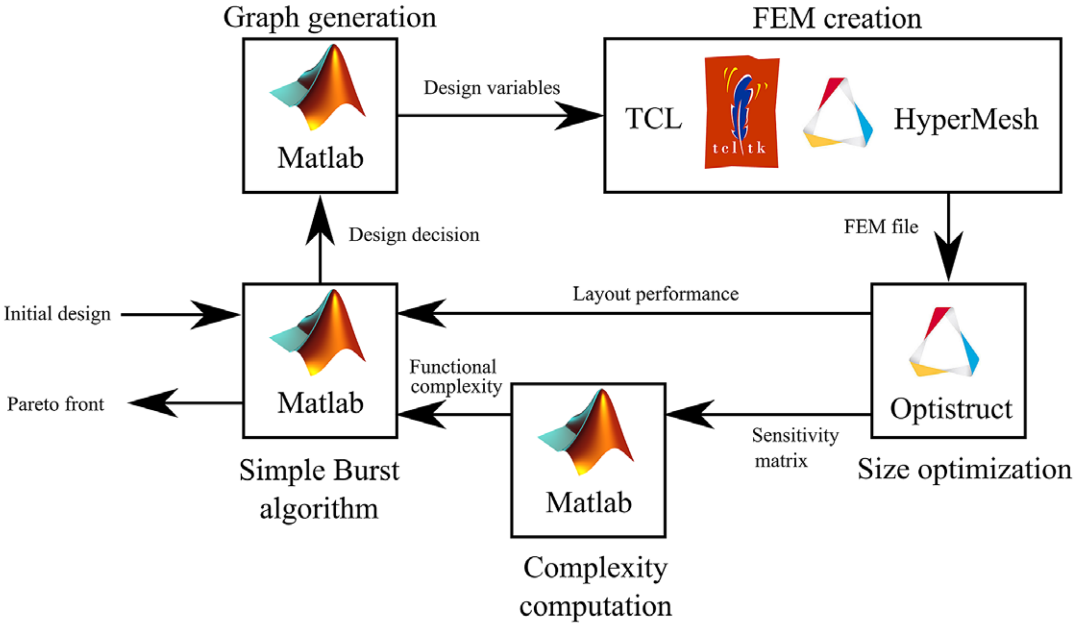

Figure 3. Components of the CD-LEAS process.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

accurate. We developed an automation script in HyperMesh (HyperWorks 2018) to create a finite-element model (FEM) for each proposed topology. Furthermore, each layout is sized with an inner optimization loop and will be discussed in more detail in the representation section. HyperMesh and Optistruct from Altair (HyperWorks 2018) are used to create, evaluate and size each model. The automation is done via the TCL scripting capabilities of HyperMesh (HyperWorks 2018).

Finally, for navigation, CD-LEAS uses a relative functional complexity measure as one of the objectives to drive the design space exploration. This measure integration has two advantages: first, it ensures that layouts can be more easily transferred into detailed design and second, that optimization seeks complex solutions only if it benefits the objective function. To navigate the disconnected topology design space, we decided for a stochastic graph search algorithm, namely the basic Burst algorithm (Königseder & Shea 2014). The navigation is also implemented in Matlab (MATLAB 2020).

As for the structure of this paper, we first discuss the source and usage of the proposed functional complexity measure. We then develop how the generic complexity measure is applied for the specific case of layout optimization of stiffened panels. The measure is presented in parallel with the description of the inner sizing optimization loop. From there, the discussion focuses on the graph-grammar generation process. Our implementation of the basic Burst algorithm is explained with the adaptation necessary for complexity-driven navigation. Finally, two case studies of the common usage of stiffened panels are proposed to showcase the efficiency of CD-LEAS. The first focuses on a rectangular pressure bulkhead and the second on a highly compressed, buckling-sensitive panel.

3. Complexity measures for optimization

In the topology optimization community, some published research has implemented complexity constraints to topology optimization. In density-based topology optimization, size filters are considered complexity limits (Sigmund 2007). In GSM, the number of nodes, edges and connections are used to measure and constrain geometric complexity (Torii, Lopez, & Miguel 2016). In LSM, the complexity is calculated depending on the number of basis functions (Zhang *et al.* 2017). Finally, for MMC, complexity is measured by the number of effective components (Zhang *et al.* 2017). In summary, in the topology optimization community, complexity is only measured with regard to the geometry of the solution and not its overall design challenge. In this work, design complexity refers to the difficulty of subsequent design phases. Using design complexity to drive an automated design and optimization process has not been done before, to the best knowledge of the authors. As discussed earlier, the nature of the complexity measure for CD-LEAS is not as important as the fact it exists to shape the design space towards something that is manageable by optimization in a reasonable time.

From the existing topology complexity measures, we found none that answered our need to translate design complexity as an optimization objective. There is already existing knowledge on this aspect in the axiomatic design community, for which the authors were already familiar with (Suh 2005). As such, we leveraged existing know-how and saw a clear path from the qualitative approach towards a quantitative measure that we could implement with stiffened panels. Measuring

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

design complexity is an ongoing area of research, with perspectives from many different disciplines (Min, Suh, & Hölttä-Otto 2015; Mohebbi, Achiche, & Baron 2018; Sinha, Suh, & de Weck 2018; Chouinard, Achiche, & Baron 2019; Bjarklev *et al.* 2020). In this work, we focus on the definitions used in axiomatic design for conceptual design. As such, complexity is not only related to geometry, but it also becomes dependent on the specific design problem. Qualitative aspects such as a measure of interaction between discrete components are often used in complexity measurement literature. This concept is strongly linked to axiomatic design through the first axiom of Independence. Measuring for design parameters interaction is a good evaluation of complexity, as unpredictable interactions will push the design process towards an inefficient round of trial-and-error to identify a successful solution. As such, complexity is not only related to geometry, but it is also dependent on the specific design problem. The measure we propose is relative and allows for an easier synthesis of multiple solutions with respect to a specific problem.

4. Complexity measure with axiomatic design

Axiomatic design is born out of a need to create a systematic approach for the design process (Suh 2005). Axiomatic design provides a framework to understand the effect of design on the quality of the product. In other words, it makes sure that *you do the right things and that you do things right* (Farid & Suh 2016). One of the main aspects of axiomatic design is the separation of the design activity into four domains, customer attributes (CA), functional requirements (FR), design parameters (DP) and process variables (PV) (Suh 1990).

For CD-LEAS' complexity measure, we focus on the relationship between the FRs and DPs, which is characterized and given by the design matrix (**A**). The design matrix can help identify the types of information that are present in the current layout (see Eq. (1)). The design matrix may be built in two different ways, either as a set of linear equations or using the differential relationship of each FRs with respect to each DPs (see Eq. (2)). In this work, we use the differential design matrix in order to leverage information from optimization:

$$[\mathbf{FR}]_m = \mathbf{A}[\mathbf{DP}]_n, \tag{1}$$

$$A_{ij} = \frac{\delta FR_i}{\delta DP_j}. \tag{2}$$

In addition to the domains, there are two identified axioms that stand as the basis of axiomatic design: the Independence and the Information axioms (Suh 2005). The first axiom stipulates that a good design maintains independence between the FRs. The second axiom puts emphasis on the need to minimize the Information content of the design. The Information content is defined as the probabilistic relation related to the number of combinations that leads to uncontrolled FRs with respect to the DPs (Suh 2005). Said otherwise, the Information content of a design reflects the probability of DPs not satisfying every FRs.

Using these axioms as starting points, Suh has defined complexity as the difficulty to fulfill these two axioms in a single design (Suh 2005). Further work by Puik & Ceglarek (2016) has extended this idea, by developing a design

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

complexity theory that uses Information content as its focus point. Their theory separates design Information into different categories, providing a framework to understand complexity:

Unrecognized information	It is the unrecognized coupling between FRs and DPs that leads to unexpected complexity. This Information content becomes recognized when erratic behavior happens during product testing.
Recognized information	It is the recognized coupling relations between FRs displayed in the design matrix.
Axiomatic information	It is the uncoupled or decoupled information that is due to a difference in design and system ranges. It deals with the probability of DPs to satisfy the FRs. Is usually addressed by robust design and optimization techniques.
Superfluous information	It has no effect on the relations between FRs and DPs.
Useful information	The total information that affects the relations between FRs and DPs (Puik & Ceglarek 2016).

Traditionally, the visualization of the design matrix allows engineers to categorize solutions between coupled (challenging), decoupled (acceptable) and uncoupled (ideal) solutions (Suh 2005). See Eqs. (3), (4) and (5) for the possible coupling levels observable in the design matrix. Uncoupled solutions are the easiest solutions to implement, as no coupling between DPs can perturb the FRs of the design thus making design easy independently of the design decisions order.

Uncoupled matrix:

$$\begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix} \tag{3}$$

Decoupled matrix:

$$\begin{bmatrix} X & X & X \\ 0 & X & x \\ 0 & 0 & X \end{bmatrix} \tag{4}$$

Coupled matrix:

$$\begin{bmatrix} X & X & X \\ x & X & x \\ X & x & X \end{bmatrix} \tag{5}$$

In axiomatic design literature, there are two measures based on the design matrix for assessing the coupling of the design matrix: re-angularity and semi-angularity (Suh 1990). Re-angularity measures the coupling of each FRs, with respect to all DPs (Suh 1990). In a similar fashion, semi-angularity can detect if a system is decoupled or uncoupled (Suh 1990). Both of these measures are not meant to measure the relative complexity of a given system, it only classifies the design matrix in one of the three coupling categories. Still, we keep in mind what they are trying to measure: the coupling of FRs, that is, the recognized Information content in Puik's framework.

5. Complexity measure for complexity-driven exploration

The present work uses the information complexity framework proposed by Puik and Ceglarek, which was discussed in the last section (Puik & Ceglarek 2016). Note that our goal is not to directly measure the information content of each design, but to develop indirect approximations that correlate with the Information content. Unrecognized information cannot be identified using only analytical data, but the three other types (recognized, superfluous and axiomatic) can be approximated to reduce unnecessary complexity.

In this section, the three approximations are introduced, with the symbol Ψ representing a complexity approximation. These approximations will be aggregated using a simple Euclidean norm to approximate the useful (or total) Information content, see Eq. (6).

$$\Psi_U = \sqrt{\Psi_A^2 + \Psi_R^2 + \Psi_S^2}, \tag{6}$$

where Ψ_U is the useful Information content, Ψ_A is the axiomatic Information content, Ψ_R is the recognized Information content and Ψ_S is the superfluous Information content.

Let us start by approximating the recognized Information content, which is the coupling of the FRs with respect to DPs. Said in mathematical terms, we seek to measure the collinearity of the FR vectors in the design matrix. A good tool to measure the collinearity of vectors is the matrix conditioning number. In this work, we start from a generic definition (see Eq. (7); Trefethen & Bau 1997). This conditioning number uses the singular value decomposition (SVD) to measure if any collinearity in the design matrix. For our implementation, we can use the function *cond()* natively provided in Matlab (MATLAB 2020).

$$cond(\mathbf{A}) = \frac{max(SVD(\mathbf{A}))}{min(SVD(\mathbf{A}))}. \tag{7}$$

However, this definition only uses the extreme (min and max) measures of the matrix and as such can only identify if there is any collinearity or not at all. It is a very useful measure of the complexity of computing the inverse of an arbitrary matrix, but not a good measure of recognized Information. Consequently, as a recognized Information measure, we propose a normalized sum of all SVD components:

$$\Psi_R = \frac{\sum_{i=1}^{n}\left[1 - \left(\frac{SVD(\mathbf{A})_i}{max(SVD(\mathbf{A}))}\right)^2\right]}{rank(\mathbf{A})}, \tag{8}$$

where Ψ_R is the recognized complexity, the *SVD* operation returns the singular value decomposition vector and the *rank* operation is the size of the design matrix. This measure of recognized Information content is relative. When Ψ_R goes to zero,

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

there is no collinearity, as all the SVD elements are unity. If there are multiple collinearities detected, then the measure goes up to one.

Another aspect that is useful to measure is the axiomatic Information, which measures how difficult it is to realize the functions with a given set of DPs. It is an indirect measure of the robustness of the DPs with respect to the FRs. To approximate axiomatic Information, we use the condition number of a nonlinear function (Trefethen & Bau 1997), as defined in Eq. (9). The condition number of a nonlinear function reflects the sensitivity of the FRs with respect to the DPs (Trefethen & Bau 1997).

$$\Psi_A = \frac{\|J(\mathbf{x})\|}{\|f(\mathbf{x})\|/\|\mathbf{x}\|}, \tag{9}$$

where Ψ_A is the axiomatic complexity, $J(x)$ is the Jacobian matrix of the sizing problem at the local optimum, $f(x)$ is the performance at the local optimum and x is the DPs at the local optimum. The Euclidean norm was used for this equation, but any matrix and vector norm can be used. For our application, as we use the differential notation of the design matrix $\mathbf{A}$, we can simply replace the design variable as the Jacobian $J(x)$ in the axiomatic complexity equation. From our experiences with stiffened panels, Ψ_A usually varies between 0 and 5. When Ψ_A goes closer to zero, it means that big variations of DPs have a low impact on FRs. As it gets higher, the impact of DPs increases on FRs, making the design more complicated to manage.

Finally, superfluous Information can arise when some DPs have no impact on the FRs, they just make optimization more difficult. As such, we define the amount of superfluous information using an arbitrary threshold (we use 10%) on the norm of the vector of the influence of each DPs, as shown in Eq. (10).

$$\Psi_S = 1 - \frac{rank(\mathbf{A}_{reduced})}{rank(\mathbf{A})}, \tag{10}$$

where $\mathbf{A}_{reduced}$ is the reduced design matrix, from which DP vectors with a norm below 10% of the maximum vector norm of A are removed.

Unrecognized Information is not captured in the design matrix, and as such we are not able to measure it. Still, solutions with low complexity, with a low recognized, axiomatic and superfluous information, will have an easier time managing the added Information that can arise during testing, production and maintenance.

In summary, each of these Information content approximations can be used to approximate a relative complexity measure. This measure of complexity could potentially be used for any design problem where the design matrix can be computed in its differential form. In the next section, we introduce how the complexity is measured specifically for stiffened panels as well as a description of the inner optimization loop.

6. Representation: Complexity, objective and constraints evaluation for stiffened panels

For CD-LEAS, we need to be able to measure performance and complexity automatically for any layouts created. In order to do so, we present here how we

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

deal with the FRs, constraints and performance of stiffened panels using a FEM-based inner loop sizing optimization. Stiffened panels are composed of multiple subcomponents; the main component is the skin to which stiffeners are added in order to increase strength (Megson 2017). The stiffeners can be built in different ways, but generally have a pad-up section, a web that is perpendicular to the skin and a free flange to increase the buckling resistance of the stiffener (see Figure 1). The panels are periodically attached to bigger assembly components such as the spars and ribs (Megson 2017).

As the stiffened panels are built using thin sheet metal or laminated composite components, buckling stability, imperfections and out-of-plane forces are the main concerns for their design (Bruhn 1973). The most used design method for stiffened panels is based on handbooks (Bruhn 1973; Megson 2017) that contain important relations and equations for the orthogrid layout. For example, analytical and semi-empirical equations exist to predict skin and Euler buckling, as well as Euler-Johnson's post-buckling collapse (Bruhn 1973, Megson 2017).

In this work, a more flexible analysis approach is necessary to model any layouts. FEM is a flexible and accurate approach, at the trade-off of more computing power than handbook methods. The buckling analysis is done via eigenvalue analysis. Post-buckling is out of the scope of the current implementation of CD-LEAS, but it could be implemented in a future revision.

To help with the synthesis and comparison of the different layouts, an inner loop sizing optimization is included in the performance evaluation. For the inner loop, we use Optistruct, which includes a family of efficient gradient-based solvers. The sequential quadratic programming (SQP) algorithm is used for the sizing problem.

From experience, we have observed a good convergence towards the same minima from different initial values, indicating a relative stability of the optimization. Evaluating the design space convexity is out of the scope of this work. Of course, more work and computing power could be leveraged in future studies to ensure global optimality of the inner-loop optimization problem.

The objective of the sizing optimization can vary depending on the use case. It could be a weight minimization with regard to nonlinear constraints, or a simple compliance minimization given a weight constraint. The exact formulation of the sizing optimization depends on the case study and is discussed in their respective sections.

At the optimum obtained from the sizing optimization, it is possible to measure the layout's complexity. At this point, the distinction between the optimization and design problem is central, as the optimization problem may be formulated differently, as to approximate the design problem. In CD-LEAS, thickness variables and the compliance objective from the optimization are selected as the simplified DPs and FRs of the design problem, as described in Eq. (11). To note, this optimization formation is different than the inner-loop optimization problem, and is only used to compute sensitivity, which is used to approximate the design matrix $\mathbf{A}_{\mathbf{SP}}$.

$$\min_{\mathbf{t}} \sum_{i=1}^{m} C_i, \tag{11}$$

where $\mathbf{t}$ is the thickness of each segment of a given layout, C_i is the compliance of each segment, i is the segment id and m is the number of segments.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

Compliance, computed as the deformation energy, is most often used in topology optimization as the main optimization objective (Bendsøe & Sigmund 2003). It allows for a fast convergence towards not only stiff, but coherent material distribution (Bendsøe & Sigmund 2003). In topology optimization, a coherent material distribution ensures structural stability (e.g., no unexpected possible displacement) and connectivity of boundary conditions. As such, compliance is the best quantifiable mean to represent the FR of structures.

While topology optimization commonly measures the compliance of the whole structure by measuring the total strain energy, in CD-LEAS we are instead interested in the local compliance of discrete skin and stiffener segments. As such, in this work we define the FRs of the stiffened panel as the stiffness of each component, that is each skin and stiffener sections (measured as their local compliance). The DPs of a given layout are defined as the thickness of each subcomponents.

Any other responses, such as weight, stress, buckling or fatigue, can be considered as design constraints and consequently, they are not required to adhere to the design axioms (Suh 2005). Therefore, it is important to point out the difference between design constraints and optimization constraints, as well as the nuance between FRs and optimization objectives. The optimization objectives and constraints are to be defined from the FRs, DPs, and design constraints while the optimization is built to answer a specific design question. For example, an optimization problem could answer the question: “What is the minimal weight this layout could achieve, with respect to a stress constraint?”. For this question, the FRs and DPs remain stiffness and thicknesses, but the optimization objectives and constraints are respectively weight and stress. As such, a well-defined optimization problem can use design constraints as optimization objectives, while the design FRs are used as optimization constraints. They are related, but not causal. More discussion on this in the case study section.

Using local compliance measures as the FRs allows the process to build the design matrix from the sensitivity analysis of the sizing optimization, with regard to the thicknesses and compliance of each subcomponent, see Eq. (12). Emphasis here, the optimization sensitivity is an approximation of the design problem and does not need to be the same as the inner-loop optimization problem.

$$\mathbf{A_{SP}} = \begin{bmatrix} \frac{\delta C_1}{\delta t_1} & \cdots & \frac{\delta C_1}{\delta t_m} \\ \vdots & \ddots & \vdots \\ \frac{\delta C_m}{\delta t_1} & \cdots & \frac{\delta C_m}{\delta t_m} \end{bmatrix}, \tag{12}$$

where $\mathbf{A_{SP}}$ is the design matrix of a given stiffened panel, C_i denotes the compliance of each subcomponent and t_i is their thickness.

With this design matrix for stiffened panels, it is possible to implement the complexity measure of discussed in the previous section. The updated Eqs. (13), (15), (14) update the design matrices from $\mathbf{A}$ to $\mathbf{A_{SP}}$. We also updated the FR and DPs from generic to applied; $f(\mathbf{x})$ and $\mathbf{x}$ to $\mathbf{C}(\mathbf{t})$ and $\mathbf{t}$.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

$$\Psi_{R,SP} = \frac{n\sum_{i=1}^{n}\left[1-\left(\frac{SVD(\mathbf{A_{SP}})_i}{max(SVD(\mathbf{A_{SP}}))}\right)^2\right]}{rank(\mathbf{A_{SP}})}, \tag{13}$$

$$\Psi_{A,SP} = \frac{\|\mathbf{A_{SP}}\|_2}{\|\mathbf{C}(\mathbf{t})\|_2/\|\mathbf{t}\|_2}, \tag{14}$$

$$\Psi_{S,SP} = 1 - \frac{rank(\mathbf{A}_{SP,reduced})}{rank(\mathbf{A_{SP}})}, \tag{15}$$

$$\Psi_{U,SP} = \sqrt{\Psi_{A,SP}^2 + \Psi_{R,SP}^2 + \Psi_{S,SP}^2}. \tag{16}$$

Until now, we have described a general functional complexity measure and then how it can be used for a sizing optimization problem of a stiffened panel with any layout. In the next section, we describe a graph-grammar implementation that can create any layout with straight stiffeners from simple design rules.

7. Generation: graph-grammar for stiffened panels

In generative design, graph-grammars are used to generate design concepts from explicit design actions and rules that affect each component and their relationships. There are multiple types of graph-grammars and in this work, we have chosen to work on a set grid with a simple graph-grammar composed of only one possible type of action and a few rules. We use a fixed grid for two reasons. First, it allows for an easy implementation of a "bar code," which is a unique ordered string that describes the topology of the layout. The bar code ensures that the navigation algorithm does not investigate the same solution twice, via different decisions path. Secondly, fixed-grid solutions are much easier to work with for a reinforcement learning algorithm, which will be studied in the future in our research group.

Using this grid, the only available action for our grammar is named "CreateStiffener," which has to respect certain rules before being applied. This action creates an edge and connects two nodes on an existing grid, activating each node it passes through. See Figure 4 for an example of actions with a 5 × 5 grid. To define the rules of our implementation, each node of the fixed grid has a specific status. They are either "active," "offset," "side" or "inactive." "Active" and "Inactive" refer to nodes with or without a connected edge respectively "Side" nodes are a special case of "active" nodes, that is, they are always considered active but will only allow the action "CreateStiffener" between nodes of different sides. "Offset" nodes are semi-active nodes, in the sense that if a stiffener passes in their area of influence, the "offset" node becomes available for connection and their position is modified to be at the crossing of the new stiffener. Implement the "offset" nodes allows the identification of a graph independently of the sequence of rules required to get there.

With this single action and rules, the graph-grammar can create a graph representation of any piecewise linear stiffening layout. From this graph representation, we can convert the position of nodes and edges connectivity into a parametric file that is passed to a TCL script that builds a FEM model of the given

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

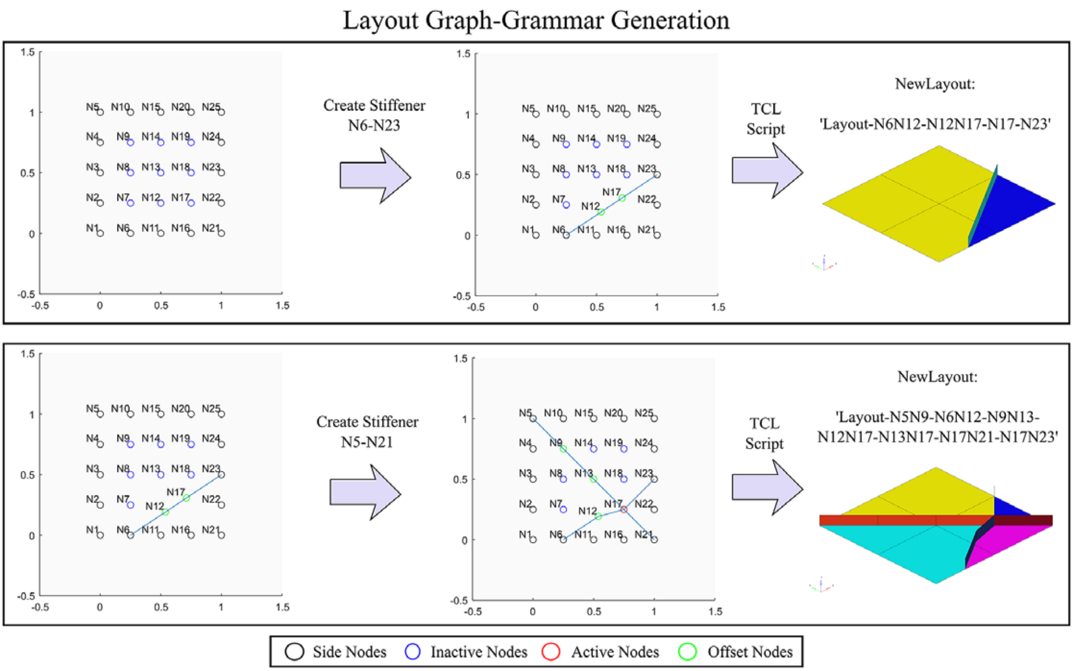
Layout Graph-Grammar Generation
N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 N15 N16 N17 N18 N19 N20 N21 N22 N23 N24 N25
Create Stiffener
N6-N23
TCL
Script
NewLayout:
'Layout-N6N12-N12N17-N17-N23'
Create Stiffener
N5-N21
'Layout-N5N9-N6N12-N9N13-
N12N17-N13N17-N17N21-N17N23'
-0.5
0
0.5
1
1.5
Side Nodes
Inactive Nodes
Active Nodes
Offset Nodes

Figure 4. Generation of graph representation and conversion to FEM from the application of the action "CreateStiffener". The generated FEM has the same aspect ratio of 1:1, but it is possible to change the conversion to other surfaces.

layout using HyperMesh automation capabilities (HyperWorks 2018). The TCL script also calls Optistruct for the inner-loop optimization problem. Also, as the node positions are described on a grid from 0 to 1 in 2D, the TCL script can easily extrude the stiffeners with respect to the normal of any planar or curved plane using a bilinear projection.

This graph-grammar is more explicit than GSM topology optimization for a simple reason: the decisions are not made for each single edge. Said otherwise, the layout is not fixed, only the grid. The process is able to create stiffeners, by connecting edges without preexisting connections. This approach ensures that significant topological changes are made at each decision. Another advantage of the graph-grammar is that the description of the layout is independent of the sequence to get to the solution. In the future, more complex actions could be built, such as "create grid", "copy pattern", "mirror pattern" or even introduce curvilinear stiffeners.

Still, one limitation is to be noted. As the grid is fixed, where more than one stiffener passes near an "offset" node, the offset is removed and all the edges connect at the fixed node. This tweak ensures that a given layout, with a given bar code, reflects the same structure independently of the order of previous decisions. So, for a more "precise" description, a finer grid of nodes can be used, although it significantly increases the design space. For this current study, a limit to 10 × 10 grids is set, which is still more than enough to find interesting solutions for simple stiffened panels. We also implemented a symmetry option, which helps reduce the size of the problem.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

The graph-grammar is implemented as a class in Matlab. This class is capable of returning a list of possible actions that respond to the design rules, and in the next section, we describe the algorithm used to choose which actions to take in this list.

8. Navigation: complexity-driven basic burst-algorithm

The list of possible actions for any layouts is large. Therefore, the stochastic search process is about searching effectively through a decision tree with controlled random decisions that reduce the total amount of required performance evaluations. Commonly used stochastic search algorithms are the greedy search, random search, random walk, evolutionary algorithms, etc. (Hoos & Stützle 2004). The choice of a search algorithm is driven by the need to balance randomized and goal-directed search; decide which is more important in terms of exploration versus exploitation (Hoos & Stützle 2004). For our case of multiobjective graph-grammar navigation, we found that the Burst algorithm provided simple parameters that can be leveraged to push navigation towards exploration rather than exploitation (Königseder & Shea 2014). A basic and advanced version of the Burst algorithm are proposed in Königseder & Shea (2014). For the sake of simplicity, this work uses our own implementation of the basic Burst algorithm. A possible future improvement for CD-LEAS could be to implement the advanced Burst algorithm or any more efficient search algorithm.

Our implementation of the basic Burst algorithm is illustrated in Figure 5. The algorithm is initialized with any number of initial layouts that are placed in the archive. Throughout the search, the archive will contain all layouts and their properties. At each iteration, a Pareto Front is built from the layouts in the archive.

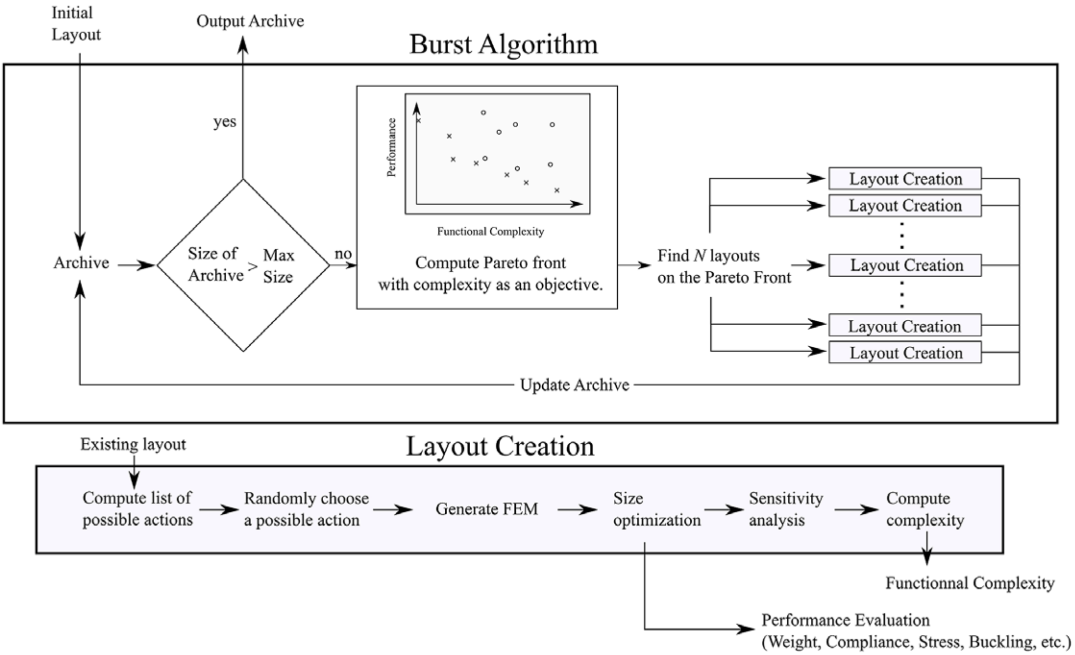

Figure 5. Our implementation of the basic Burst algorithm.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

In our implementation, we used the script available from the MathWorks file exchange (Tom 2021). From the Pareto front, we randomly choose N layouts that are created, updated and evaluated in parallel. N can easily be scaled to the number of cores available in the user's computer or server. We added a condition to ensure that each layout of the Pareto front is chosen at least once for improvements at each iteration. As we found the Pareto front to be too small at the beginning of the search, a relaxation mechanism was used; if the number of layouts on the Pareto front is smaller than $N{*}r$, where r is the relaxation parameter, we redo the Pareto evaluation on the remaining nondominant layouts. This step is repeated until there are at least $N{*}r$ layouts in the active search set. In our test cases, it was found that using $r = 3$ yields a good balance between exploration and exploitation. In short, N and r are search parameters that can be used to tune exploration versus exploitation. If we increase the values of N, there are going to be more decisions made from a single set of layouts without looking at performance. Navigating randomly in short bursts can increase exploration and diversity, at the cost of more evaluations that might lead to uninteresting layouts. At the opposite, small values of N will make the algorithm behave more like a greedy search, thus improving the exploitation at the cost of exploration. The only other search parameter is the stopping criteria $Archive_{max}$ which is simply the maximum number of created layouts.

To summarize this method section, CD-LEAS introduces a balance of generation, representation, and navigation compared to current topology optimization methods. This balance is closer to what is used in industry, where exploration and simplicity are key to finding new sound and manufacturable solutions. The relative complexity measure is used to focus on exploration as well as to limit the search within simple layouts. The graph-grammar allows for the generation of solutions that is decoupled from the performance evaluation. And finally, our implementation of the basic Burst algorithm is used to explore the design space, with a good balance of exploration versus exploitation.

The next section will demonstrate the capabilities of CD-LEAS on some challenging industrial topology optimization problems that our research team encountered previously (Dugré *et al.* 2016; Gamache *et al.* 2020, 2021).

9. Case studies of CD-LEAS

In this section, we propose two different case studies of common loading conditions of stiffened panels: a pressurized bulkhead and a highly compressed wing section panel of a wing section.

The only elements that change between the examples are the boundary and load conditions. Even then, the case study has very different design spaces for which CD-LEAS can find adequate results. Furthermore, a comparison with the SIMP method for the pressurized bulkhead and handbook optimization for the compressed wing section are presented.

10. Pressure bulkhead with out-of-plane loading

The pressure bulkhead is a known difficult challenge for topology optimization (Dugré *et al.* 2016; Warwick, Mechefske, & Kim 2019). In our previous work, several shortcomings of SIMP were found in this case study, finding that the

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

interpretation required an extensive study of the layouts proposed by topology optimization (Dugré *et al.* 2016).

11. Description of the pressure bulkhead case study

In aircraft, bulkheads are at the front and back of fuselages to keep a stable pressure in the cabin during flight operations. It is usually round, but was kept as a square panel in this study for the sake of simplicity.

The plate is flat on the X–Y plane, modeled using Shell (PSHELL in Optistruct) elements with quadratic interpolation (CQUAD8 in Optistruct). These elements are used to model accurately the rotation and displacement, as they have six degrees of freedom, but with reduced computation cost relative to 3D elements. This modeling approach is accurate as the modeled structure is made of thin components. The boundary conditions are simply supported ($U_Z = 0$) on all sides and pressure is applied evenly on all the plate as a distributed normal force P_z, see Figure 6. Aluminum 7075 is used and its properties are found in Table 2. The graph-grammar actions and rules are the same as presented in the previous sections.

CD-LEAS can be discussed as a two-level optimization scheme, with the outer loop focusing on the layout/topology and the inner loop with sizing the

Table 1. Properties of the model created from an interpretation of SIMP material distribution

Property	Value
Weight (w) (lb.)	5.0
Compliance (C) (lbf.in)	57
Functional complexity (Ψ_U) ($-$)	1.8

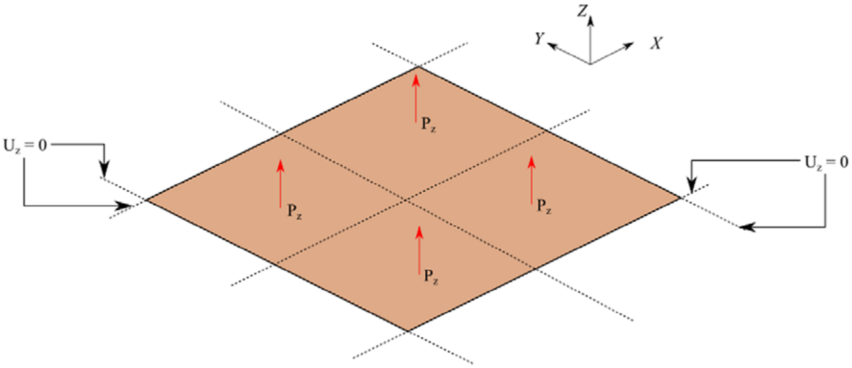

Figure 6. Test case of a stiffened panel with a uniform pressure, with simply supported boundary conditions. U_z: Imposed Displacement (all other degrees of freedom are not restricted). P_z: Uniform Pressure Value. The panel is 20 × 20 inches. The boundary conditions simulate the stiffness of the ribs around the panel and keep the skin free to rotate. Elements are of size 0.3 in, PSHELL with quadratic interpolation.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

Table 2. Aluminum 7075 properties (Rice *et al.* 2003)

Property	Value
Young's modulus (E_0) (ksi)	10,700
Poisson's ratio (ν) ($-$)	0.33
Yield limit (F_{cy}) (ksi)	68
Material density $\left[\frac{lb.}{in^3}\right]$	0.10

components. For the pressure bulkhead case study, we are seeking an optimal placement of stiffeners with respect to a weight constraint. This formulation is selected to be able to compare easily with topology optimization. The weight constraint is only applied in the inner loop, and layouts that cannot achieve the weight constraint are eliminated from the outer loop. This makes the implementation of the layout optimization easier, as the single objective gradient descent of the inner-loop can deal with optimization constraints more easily than the heuristic multiobjective outer loop. The topology multiobjective outer-loop optimization formulation is given in Eq. (17).

$$\min_{\mathbf{D}} \quad \sum_{i=1}^{n} C_i(\mathbf{D}), \Psi_U(\mathbf{D}), \tag{17}$$

where $\mathbf{D}$ is the layout graph (topological variables), C_i is the compliance of each subcomponent, Ψ_U is the useful complexity and n is the number of subcomponents of a given $\mathbf{D}$ layout.

For the inner loop sizing optimization of this test case, the compliance is minimized with respect to a weight constraint, see Eq. (18).

$$\begin{array}{ll} \min\limits_{\mathbf{t}} & \sum\limits_{i=1}^{n} C_i(\mathbf{t}) \\ s.t. & \sum\limits_{i=1}^{n} w_i(\mathbf{t}) \leq 5.0 \end{array}, \tag{18}$$

where $\mathbf{t}$ is the thickness of each subcomponent, C_i is the compliance of each subcomponent, $w_i(\mathbf{t})$ is the weight of each subcomponent and n is the number of subcomponents of a given $\mathbf{D}$ layout.

The weight constraint is set to 5.0 lb. and the value of the pressure is arbitrarily set to 10 psi. An arbitrary pressure value does not have any impact on the layout in the case of compliance minimization, as compliance is directly proportional to the applied loads. What is important for the compliance value is the direction and location of the load.

As a layout reference, we ran the SIMP method with Optistruct (Optistruct 2018) and converted the results to an explicit stiffened panel, see Figure 7. SIMP was run with standard parameters, penalty factor is set to 2, volume constraints to 20%, and shell thickness to 1 inch. The results for the SIMP results can be found in Table 1. Finally, the Burst and graph-grammar parameters are defined as follows for the pressure bulkhead case.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

Evaluation per iterations (N)	8
Max. evaluation ($Archive_{max}$)	500
Relaxation (r)	3
Graph grid size	10×10
Symmetry	From the center line in both X and Y axis.

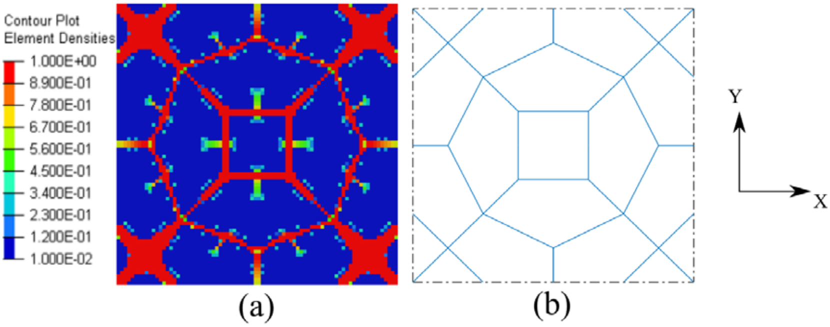

Figure 7. (a) Material Distribution using SIMP. (b) our interpretation into the graph of the panel.

These parameters offer a good balance between exploration and exploitation and computation time is 25 minutes with an AMD Ryzen 7 3700X @4.0GHz, running the eight layout evaluations in parallel on the eight cores.

12. CD-LEAS results for the pressure bulkhead

The results of CD-LEAS for the pressure bulkhead are presented in different ways. First, the layouts of the final Pareto set are presented in Figure 8. Then, the performance of the entire archive is presented in Figure 9a. Finally, the convergence plot of the compliance is shown in Figure 9b. It is possible to see that convergence is obtained quickly, in fewer than 20 iterations.

Of course, using a stochastic search approach is highly dependent on the random choices the algorithm makes during exploration. It is possible to see in Figure 8 that the obtained layouts share some similar patterns. We have rerun the search with the exact same parameters and initial layout, for which the solutions of the Pareto front are illustrated in Figure 10.

In both cases, CD-LEAS is able to find solutions that are both stiffer and less complex than the results obtained with SIMP. The best CD-LEAS result has a compliance of 37, a reduction of 35% in comparison to the layout found with SIMP. However, it comes at the cost of a more complex solution. If we chose a layout similar in complexity at $\Psi_U \approx 1.7$, the layout of Figure 8 with $\Psi_U \approx 1.62$ (third row, second column) has a compliance of 41.2, still a reduction of 28% in compliance with a similar relative complexity. For this interpretation, it is important to consider that useful complexity is approximated by the Euclidean norm of three factors: recognized Information (defined by the coupling of stiffeners sensitivity),

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

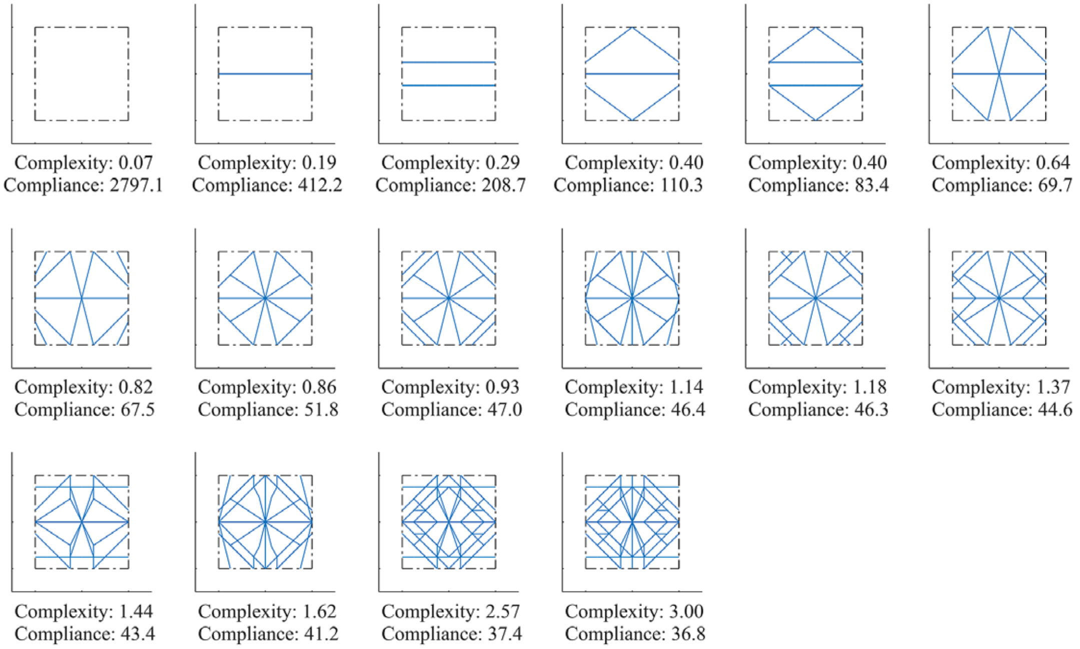

Figure 8. Solutions on the Pareto front for the pressurized stiffened panel exploration.

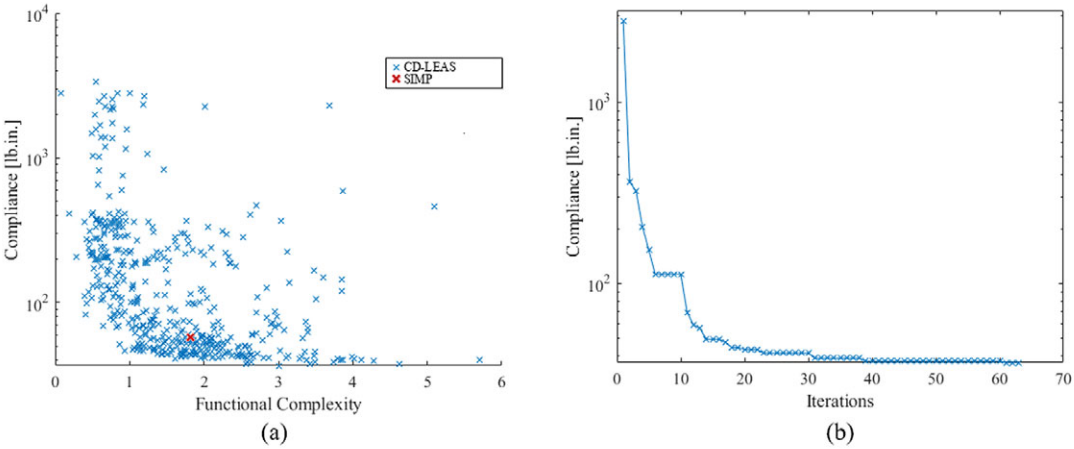

Figure 9. Results of the compliance-based run for the pressure case study. (a) Scatter of the archive of all layouts created in this compliance-based run. (b) Evolution of minimum compliance of layouts on the Pareto front for the compliance-based pressure case.

axiomatic Information (which relates to the sensitivity of the solution to the thickness of stiffeners), and superfluous Information (which concerns whether all stiffeners are meaningful or necessary). In the context of this specific comparison, it means that even though the two layouts may look very different, they present a similar design challenge.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

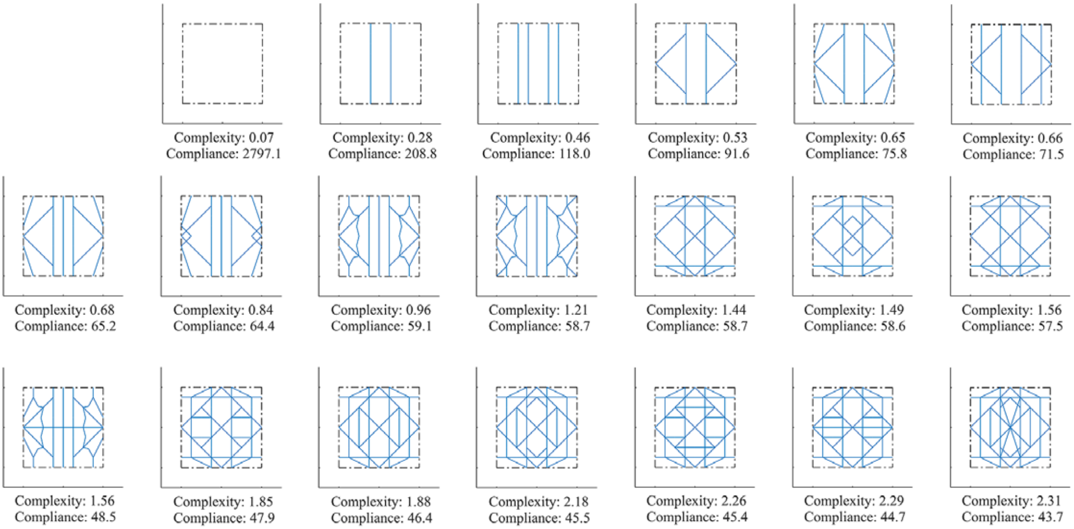

Figure 10. Second set of solutions of the Pareto front for the pressurized stiffened panel exploration. This second set has been generated using the same parameters as for the results of Figure 8 to illustrate the heuristic nature of CD-LEAS. This time, the process seems to have focused on improving a repeating pattern rather than a radial pattern.

Furthermore, the Pareto set of the pressure bulkhead problem shows a clear compromise between complexity and compliance. For the layout design of stiffened panels, obtaining components with high stiffness is regularly not sufficient for design, as there are generally other design constraints, which can be structural. Often, constraints come from other engineering disciplines involved in the design process such as manufacturing, systems or aerodynamics. Selecting only stiffness constraints will make for difficult post-processing of the results. As an example, an optimized material distribution for a wing box would not leave any place for fuel tanks. The Pareto set can empower designers to easily choose an adequate level of compliance and complexity that should allow for a much easier subsequent detailed design phase. Our industrial partners have also found it particularly useful in a multidisciplinary approach, as it can reduce the amount of interaction with other disciplines. This trade-off can be translated as a clear hierarchy of the primary and secondary stiffening components, with components of simple solutions offering the most effective compromise of complexity stiffness. The results also show a clear advantage with respect to exploration when compared to gradient-based topology optimization. Furthermore, results obtained with SIMP are often organic and complex, which creates design challenges down the road, especially with respect to unrecognized information.

In summary, CD-LEAS can rapidly generate simple and efficient solutions, even offering better results than SIMP for the case of the pressure bulkhead. A major advantage of CD-LEAS against topology optimization, such as SIMP and LSM, is the fact that the stiffeners are explicitly generated, ensuring a proper representation of the stiffened panel which does not require any further interpretation. Besides, using the Pareto front as the starting point of each iteration

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

allows the Burst algorithm not to get stuck in the design space of complex solutions, as the design space is smaller in the simple space than in the complex space.

The next section discusses the compressed panel case study as an in-plane loading.

13. Compressed stiffened panel, with in-plane loading

The design of compressed stiffened panel is also an active challenge of topology optimization, mainly due to convergence difficulty created by buckling load factor constraints (Layachi, Xu, & Bennaceur 2017; Chu *et al.* 2020; Gamache *et al.* 2021). These panels are found on the outer skins of the wing, mainly on the upper skin. The bending action of the lift and gust forces on the wing creates high compressive loads on the panels.

In the industry, these panels are currently designed and optimized using a handbook method (Bruhn 1973; Megson 2017). Relations and equations found in these handbooks are only for the orthogrid layout and are based on analytical models and empirical data to model the collapse of the panels in terms of buckling, crippling and post-buckling. The handbook design method is very efficient for large-scale problems and can easily be automated once the layout is fixed. As such, in this section, we will compare the results of CD-LEAS with the results of the design with the handbook method.

14. Description of the compressed stiffened panel case study

We propose in this section two different optimization problems, for the same design use case. The two optimization problem is used to showcase the effect of nonlinear optimization constraints on the solutions. In the first problem, we use the same optimization formulation as for the pressure panel, that is compliance and complexity minimization as given in Eq. (17) for the outer loop and compliance minimization with weight constraints for the inner loop from Eq. (18). Secondly, we introduce the buckling resistance optimization where we minimize weight and complexity, see Eq. (19), and with the inner-loop working with a buckling resistance constraint, shown in Eq. (20). This second formulation is selected as answering the question of the minimal weight of panels is the key. Here again, the buckling load factor constraint is only applied during the inner-loop optimization to ease the implementation of the outer-loop multiobjective problem. To note, in the buckling optimization case, compliance is only used for the complexity measure, which is acceptable as the design problem can be different from the optimization problem.

$$\min_{\mathbf{D}} \sum_{i=1}^{n} w_i(\mathbf{D}), \Psi_U(\mathbf{D}), \tag{19}$$

where $\mathbf{D}$ is the layout graph (topological variables), w_i is the weight of each subcomponent and Ψ_U is the useful complexity.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

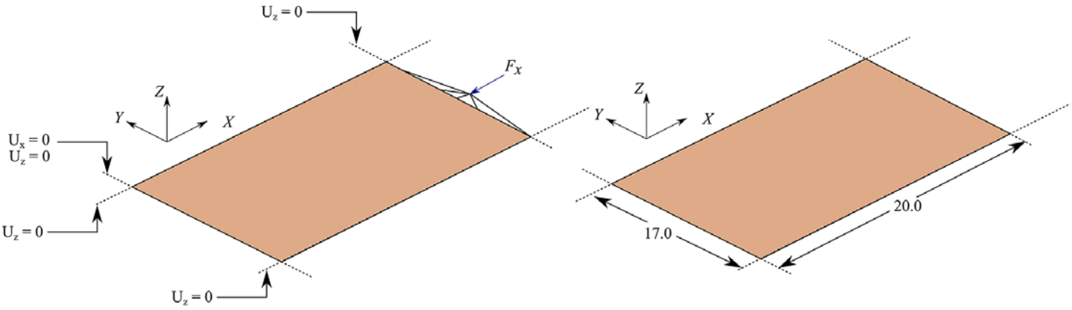

Figure 11. Test case of an axially compressed stiffened panel, with simply supported boundary conditions. Panel boundary geometry is defined from an existing section of aircraft structure between two spars and two ribs. Elements are of size 0.3 in, PSHELL with quadratic interpolation. U_z, U_x: imposed displacement, F_x: Applied Force.

$$\begin{aligned} &\min_{\mathbf{t}} \ \sum_{i=1}^{n} w_i(\mathbf{t}), \\ &s.t. \ \ \lambda_1(\mathbf{t}) \geq 1.0 \end{aligned} \tag{20}$$

where $\mathbf{t}$ is the thickness of each subcomponent, w_i is the weight of each subcomponent and λ_1 the first buckling load of the structure.

The same graph-grammar actions and rules that were presented previously are used, with the only possible action is "CreateStiffener." Here also, the plate is flat on the X–Y plane and the boundary conditions are simply supported ($U_z = 0$), see Figure 11. Elements are modeled using PShell elements with quadratic interpolation and the material is still aluminum 7075. This time, however, the compressive load is applied on one side via a rigid (RBE2) element, in the X direction. The load value is set at 120,000 lbf. The RBE2 is only active in the X direction. On the opposite side, the displacement ($U_x = 0$) is set to resist the compressive load. This loading boundary condition method is used as it reproduces the behavior of stiffened panels integrated in larger assemblies (Gamache *et al.* 2021).

For the sizing optimization, we propose two formulations for the compressed panel. The first is the same as for the pressure bulkhead, with a compliance minimization and a weight constraint. For the second one, we use a weight minimization and a buckling load factor constraint. As for the Burst algorithm, we use the same parameters as for the pressure bulkhead case.

Evaluation per iterations	8
Graph grid size	10 × 10
Maximum evaluation	500
Symmetry	From the center line of the X and Y axis.

For the bulkhead pressure case, we have used SIMP to create an optimized baseline. However, for compressed stiffened panels, no results different from the orthogrid layout could be obtained with SIMP (Gamache *et al.* 2021). Consequently, we define the baseline as the layout currently used in commercial aircraft,

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

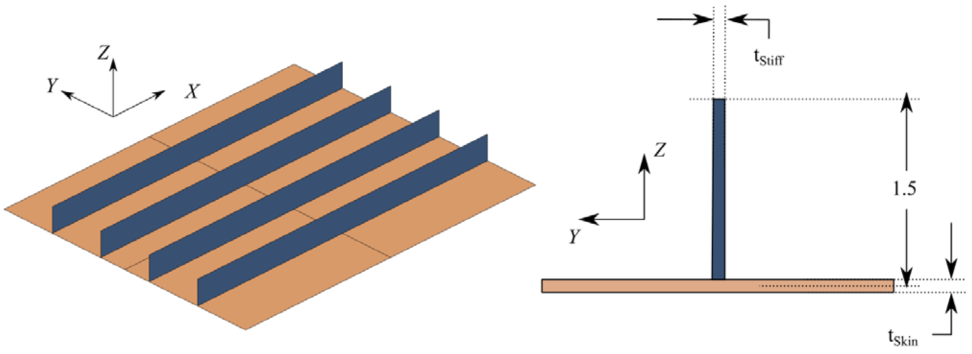

Figure 12. Compression baseline layout, from a business aircraft, sized with handbook methods. t_{Stiff} and t_{Skin} are, respectively, the thickness variables for stiffeners and skin sections.

Table 3. Baseline properties

Property	FEM sizing
T_{Skin} (in.)	0.11
T_{Stiff} (in.)	0.15
Weight (lb.)	5.3
Compliance (lbf.in.)	5000
Maximum Stress (ksi)	45
Complexity (ksi)	0.27
Linear Buckling (λ_1)	1.0

as illustrated in Figure 12. In Table 3, the result of a FEM-based sizing optimization is presented. The optimization considers the eigenvalues of the linear buckling as a constraint and minimizes the weight of the panel.

15. Compliance-based optimization of the compressed stiffened panels

In this run, we define the sizing optimization as a compliance minimization with a weight constraint, set to 5.0 lb. It took 22 minutes to create, size and evaluate the 500 layouts. The layouts found on the Pareto front of this run are illustrated in Figure 13, the full archive is presented as a scatter plot in Figure 14a and the convergence of compliance in Figure 14b.

The results of this run are interesting, as they reflect a particularly difficult case for topology optimization. As shown in the scatter of the results in Figure 14a, there is a plateau in the solution space. Any other solutions that do not provide stiffeners parallel to the load will both increase complexity and compliance. As such, for topology optimization methods using implicit generation (SIMP, Level-Set, MMC), it is very difficult to make any decisions that will not have a negative

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

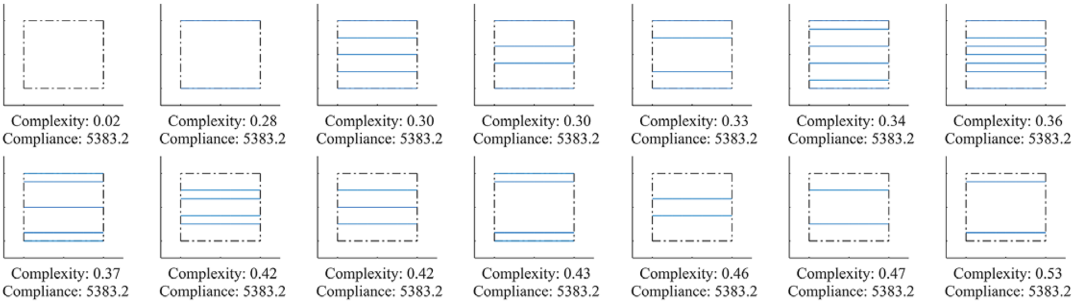

Figure 13. Layouts on the Pareto front for the compliance-based compression case.

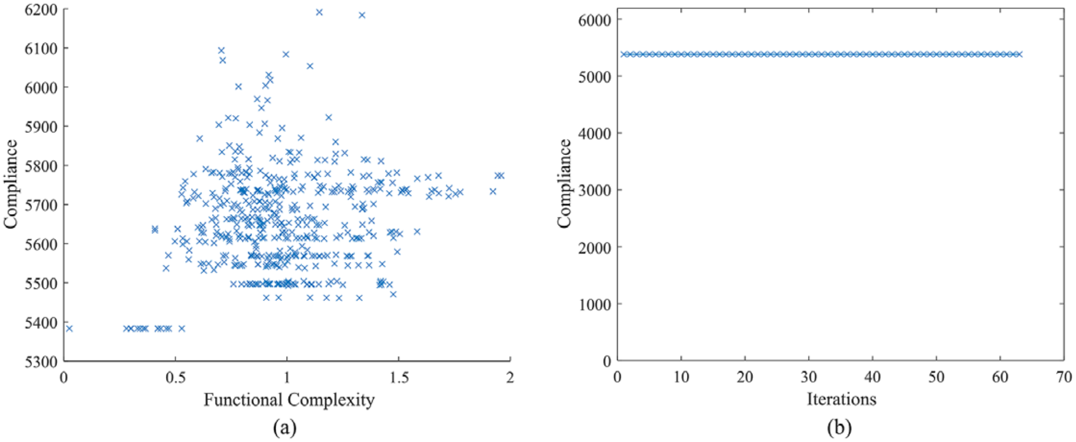

Figure 14. Results of the compliance-based run of the compression case study. (a) Scatter of the archive of all layouts created in this compliance-based run. (b) Evolution of minimum compliance of layouts on the Pareto front for the compliance-based compression case.

impact on compliance. The only topology optimization method that is capable of easily finding a solution for this case is GSM, but it is computationally very expensive due to the evaluation of buckling and the use of finite difference evaluations (Gamache *et al.* 2020).

CD-LEAS finds a range of efficient solutions at different complexity levels. Still, as noted earlier, buckling is often the main design driver for compressed stiffened panels. As such, to check if the results can beat the baseline, we use a different inner loop sizing to include buckling. As the panels are expected to be at least able to withstand the buckling, the linear eigenvalue constraint is set to 1.0. The objective is to reduce the weight of the panel, given the buckling load factor constraint. See Figure 15 for the results of each layout.

As these panels are sized, it appears that when properly sized for buckling, the proposed layout is unable to surpass the baseline. The best layout proposed by CD-LEAS, with only a compliance objective, has a weight of 5.5 lb., whereas the FEM-sized baseline weight 5.3 lb. This comparison is to be expected as the distribution of the stiffeners on the panel does not affect the compliance, but does

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

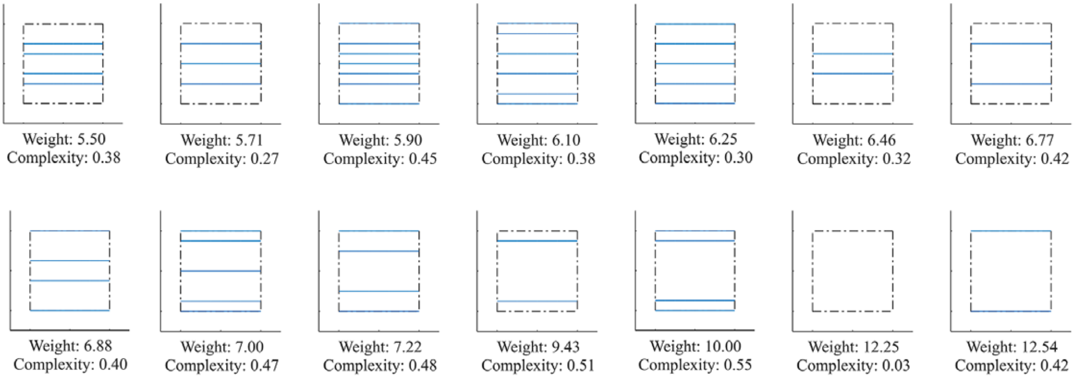

Figure 15. Sizing optimization with weight minimization and a buckling load factor constraint ($\lambda_1 \geq 1.0$) for the layouts proposed by CD-LEAS for the search with compliance only.

significantly impact the buckling. Still, by using CD-LEAS, any unnecessary components that add complexity are ignored.

In the next section, we use an inner loop sizing optimization that considers buckling.

16. Buckling-based optimization of the compressed stiffened panels

The addition of buckling in the inner loop sizing greatly increased the computation time of CD-LEAS. On the same processor, for 500 layouts creation, sizing and evaluation took 5.8 hours. Still, a reasonable amount of time, as it can easily be run overnight, which is quite usual for structural optimization (Venkataraman & Haftka 2004). Furthermore, compared to SIMP or GSM this computing time is a significant improvement. As discussed in (Gamache *et al.* 2021), SIMP is relatively difficult to use, exploration can hardly be controlled and it requires multiple days of computation to obtain only some efficient layouts. As for GSM, a single gradient descent requires more than 5 hours of computation, with the same hardware (Gamache *et al.* 2020) for the same problem. Parallelism could reduce slightly computation time, but it would still be longer than CD-LEAS. Furthermore, exploration is only controlled by changing the initial optimization variables.

For the CD-LEAS run, the layouts found on the Pareto front are illustrated in Figure 16, the full archive is presented as a scatter in Figure 17a and the convergence of weight in Figure 17b.

Compared to the compliance-based search, using buckling shows a cleaner convergence. Also, the scatter plot shows that there are some great solutions with low complexity. Furthermore, this time the best results have a weight of 5.03, lower than the baseline with a trade-off of increased complexity. The next best solution at 5.06 still beats the baseline while having a similar complexity value to the baseline.

Choosing whether to use buckling or not during CD-LEAS search is up to the available time and the problem at hand. Still, both formulations are capable of generating feasible results with a performance similar to the ones created by experienced engineers. The compromise proposed on the Pareto front is clear

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

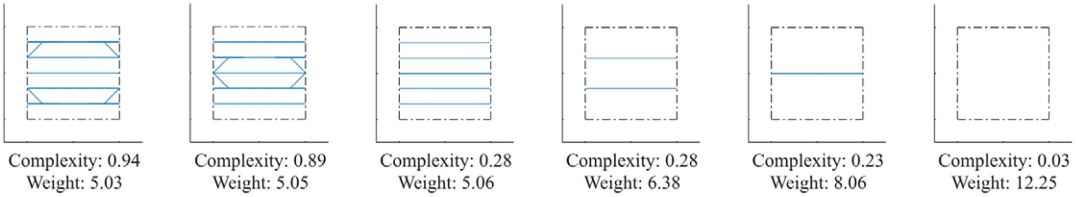

Figure 16. Layouts on the Pareto front for the buckling-based compression case.

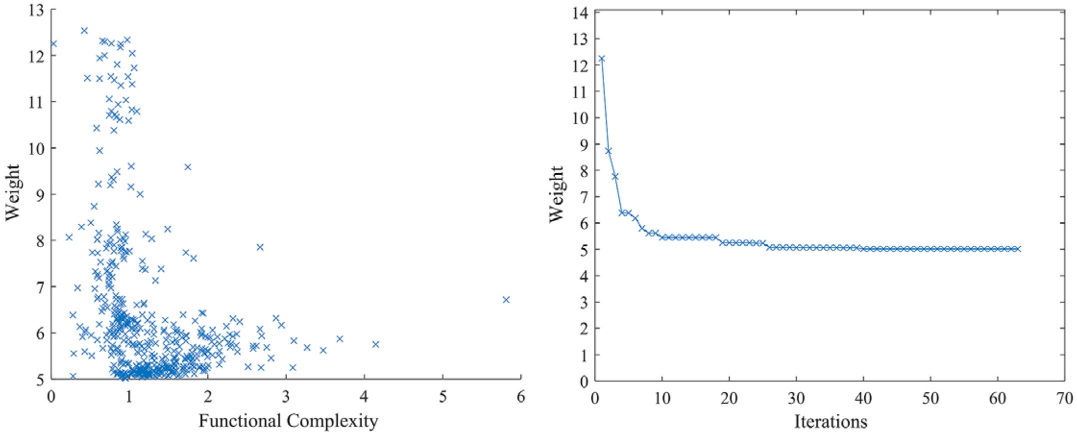

Figure 17. Results of the buckling-based run of the compression case study. The formulation is weight reduction with $\lambda_1 = 1.0$ (a) Scatter of the archive of all layouts created in the buckling-based run. (b) Evolution of minimum weight of layouts on the Pareto front for the buckling-based compression case.

and offers interesting information to the users with regards to the impact of layouts. In summary, the results obtained show a better or equal performance to the ones proposed by topology optimization, with SIMP or GSM, while requiring less computation time. More importantly, the generation used by CD-LEAS is explicit, making the representation of buckling more accurate and the exploration-focused search shows an efficient and diversified search.

17. Discussion and limitations

Both case studies have shown consistent improvement with respect to current design tools. The compliance-based solutions offered by CD-LEAS were stiff yet simple, making interpretation much easier. With our implemented complexity measure, we ensure that each stiffener behaves as independently of each other as possible (recognized Information content), that one stiffener does not dominate the performance of the panel (axiomatic Information content) and that there are no stiffeners that do not add any value (superfluous Information content). From a design point of view, our measure of complexity reflects the sensitivity of the design to parameter changes. A good physical analogy would be with a serial robotic manipulator, where you want to avoid being near a singularity (in short, a robotic

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

singularity is a pose where joint speed becomes unexpectedly high, despite an end-effector controlled at low speed) (Angeles 2003). As with the robot, there exist multiple “design” singularity that needs to be avoided to reduce avoidable surprises, each in very different configurations.

For the designers, this simplicity carries into subsequent design phases where new design constraints and objectives can be integrated more easily with the new layouts. As such, the results presented for the compressed panels should not be surprising as the orthogrid layout has been proven time again that it is a very robust and efficient layout. However, this result is only obvious as the panels are usually optimized independently, separated by orthogonally placed spars and ribs. This scenario has been replicated by the boundary conditions used in this use case. As future work will increase the scale of the design space, innovative shapes are expected to arise from the use of CD-LEAS.

The case studies show that complexity measure effectively helps navigation for an effective design space exploration. By selecting a simple, albeit nonoptimal, solution as the starting point, the graph-grammar only has a few options for moves in the beginning. This smaller pool of options acts as a relaxation of the outer-loop optimization, which allows the Burst algorithm to keep looking around existing simple layouts, even if their performance is not as promising. Eventually, the process can find solutions that have both high performance and low relative complexity. Furthermore, as shown by the case studies, CD-LEAS is more effective than current topology optimization methods for the diversification of solutions, yielding multiple interesting solutions in the same run. The diversification of solutions is a significant industrial advantage, because as more design constraints arise, it can adapt and still find sound alternatives in the solution pool.

There are two advantages in the use of a relative complexity measures to limit the complexity of new layouts. First, the functional complexity is context-dependent. This effect is shown in the compressed stiffened panel case, where using only compliance still yields results that are relatively efficient in terms of buckling. Secondly, as the complexity measure is relative, there is no arbitrary threshold to set. As the process explores and builds the Pareto front at each iteration from the whole archive of solutions, it will organically set a threshold on complexity by ignoring layouts that are both complex and with low performance, as shown in both case studies.

As discussed in the introduction, the generative design framework has three main properties: generation, representation and navigation. In this work, generation is carried out using the graph-grammar, ensuring an explicit description of components and their relations through a graph. In addition, the use of the archive and the bar code ensures that the exploration does not evaluate the same layout twice. Moreover, the explicit generation is capable of creating evaluation models with an accurate representation. In this work, the representation of each layout is provided by an automatically created inner-loop optimization problem. This inner loop provides everything needed for performance and complexity assessments. As such, in terms of representation, CD-LEAS uses automation to link both topology and size optimization, which in itself is of interest for the topology optimization community. Finally, the navigation in CD-LEAS is done using the basic Burst algorithm, using both a complexity measure and any performance assessment as the objectives. In summary, by adjusting the balance of topology optimization towards exploration rather than exploitation, CD-LEAS offers an approach to

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

layout optimization that is closer to the need of the industry. It is ready to find sound layouts and integrate new constraints that are clear compromises of simplicity and performance.

One aspect to consider, though, is that CD-LEAS trades off a reduction of complexity of the layout for an increased process complexity compared to topology optimization. SIMP topology optimization is a rather straightforward and elegant algorithm that has proven to be highly flexible when working with a simple stiffness objective. Its formulation is problem independent and physics independent. In comparison, CD-LEAS requires a more ad hoc implementation for the representation process. For the implementation discussed in this paper, we had to work with two different scripting languages and software suites that were not made specifically to work together.

In the future, there are many ways in which CD-LEAS could be improved. With regard to generation, the current implementation has only one action and few design rules. It is possible to imagine actions that add more stiffeners in a single action, such as the creation of a grid, actions that modify a grid by moving connections or still actions that combine different graphs. For example, the action "createGrid" could take as input a "spacing", "length" and "width" parameters and create many stiffeners at once, using the same design rules as we already defined. Moreover, curvilinear stiffeners have been studied in multiple works recently, and the possibility to add them could be interesting (Mulani *et al.* 2013). Another possible improvement is the implementation of primary/secondary stiffeners with different heights, such as the BCF proposed in Houston *et al.* (2017). It could also be possible to use multiple different cross-sections. Finally, the complexity measure is based on axiomatic design, which is still debated for its practical validity (Nordlund, Lee, & Kim 2015). More work could be done to improve upon the accuracy and validity of the complexity measure of CD-LEAS.

18. Conclusion

The main novelty aspect of CD-LEAS is the use of a complexity measure to drive an effective layout navigation, allowing for a more flexible explicit generation and the effective use of critical nonlinear constraints. With CD-LEAS we propose a new process for the generation of stiffened panel layouts. Leveraging knowledge acquired from generative design, axiomatic design and topology optimization, the CD-LEAS process is a two-level topology optimization process that is closer to the need of industrial structural designers and regulation aspects. This process leads to the generation of trade-offs with respect to performance and simplicity. This trade-off is important and was not evident with algorithms such as SIMP. Often times, the layouts generated required extensive design and performance analysis (Dugré *et al.* 2016; Gamache *et al.* 2019). As an example, in the buckling-based case study, a trade-off of less than 1% of weight can reduce complexity from 0.94 to 0.28. The results of this work are limited to a scope of smaller components as we have yet to test it on larger assemblies. Still, we finally have a process for which we can be confident, at least for small problems, that the results will allow the designers to learn something about the design space. In comparison, there are multiple SIMP projects that have simply sent the designer towards a challenging post-processing phase with no profits (Aage *et al.* 2017). On the whole, CD-LEAS

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

offers an improvement in both performance and process reliability and its potential for larger-scale problems is promising.

The effectiveness of CD-LEAS comes from the introduction of the complexity measure which imitates the reasoning process of an experienced designer with an open mind for the navigation of the design space. This new objective helps navigation to stay focused on simple solutions and compromises only when necessary. This approach is different from current topology optimization methods that only focus on a given objective function and will make any sacrifice necessary to complexity to get 1 or 2 final percentage of performance. As CD-LEAS considers complexity, it becomes possible to select slightly underperforming solutions, in exchange for reduced complexity which opens up the integration of possible new design constraints. Topology optimization searches only the local space of the large topological design space.

Furthermore, CD-LEAS has shown an improvement in both computation time and performance with respect to widely used topology optimization methods for the case of buckling optimization. Here again, by considering a quantifiable metric of complexity, we force the process to start with no stiffeners and thus low complexity. Increasing complexity only when necessary, also yields simpler inner loop optimization problems, consequently reducing processing time.

Improvements from CD-LEAS are presented with two case studies that are compared with previous work of the authors regarding topology optimization for stiffened panels. In the bulkhead case study, we have shown an improved exploration, a stiffer layout and multiple good trade-off solutions. For the compressed panel case study, we have shown that using CD-LEAS yields similar results from compliance and buckling-based objectives. This observation suggests a hypothesis to be validated in future work: reducing complexity yields more versatile solutions, as suggested by axiomatic design. In this particular case, previous work had identified a very difficult use with SIMP, whereas CD-LEAS has shown a consistent convergence towards an efficient minimum. Future work to explore larger and more challenging case studies will be required to assess the full potential of CD-LEAS, and more broadly the use of complexity measures in a generative design process.

Financial support

The authors wish to thank Stelia Amérique du Nord and Bombardier as industrial collaborators in this project. We would also like to thank the CARIC and MITACS for their funding of our project: MDO-1601: MUltidisciplinary Framework for Optimization of wingboX – 1a [IT07469].

References

Aage, N., **Andreassen, E.**, **Lazarov, B. S.** & **Sigmund, O.** 2017 Giga-voxel computational morphogenesis for structural design. *Nature* **550**, 84; doi:10.1038/nature23911.

Angeles, J. 2003 *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*. Springer.

Ansys© Academic Research Mechanical, Release 2020. R2, Help System, Topology Optimization, ANSYS, Inc.

Bedair, O. 2009 Analysis and limit state design of stiffened plates and shells: A world view. *Applied Mechanics Reviews* **62** (2), 020801.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

Bendsøe, M. P. & **Sigmund, O.** 2003 *Topology Optimization: Theory, Methods, and Applications*. Springer.

Bjarklev, K., **Eifler, T.**, **Mortensen, N. H.**, **Linnebjerg, S.** & **Ebro, M.** 2020 Product behavior complexity metric for early prioritization of tolerance analysis tasks. *Design Science* **6**, e1.

Bremicker, M., **Chirehdast, M.**, **Kikuchi, N.** & **Papalambros, P.** 1991 Integrated topology and shape optimization in structural design. *Journal of Structural Mechanics* **19** (4), 551–587.

Bruhn, E. F. 1973 *Analysis and Design of Flight Vehicle Structures*. Jacobs Publishing House.

Caseiro, J. F. 2013 On the elasto-plastic buckling of Integrally Stiffened Panels (ISP) joined by Friction Stir Welding (FSW): Numerical simulation and optimization algorithms. *International Journal of Mechanical Sciences* **76**, 49–59.

Chirehdast, M. & **Papalambros, P.** 1992 A note on automated detection of mobility of skeletal structures. *Computers & Structures* **45** (1), 197–207. https://linkinghub.elsevier.com/retrieve/pii/0045794992903576.

Chouinard, U., **Achiche, S.** & **Baron, L.** 2019 Integrating negative dependencies assessment during mechatronics conceptual design using fuzzy logic and quantitative graph theory. *Mechatronics* **59**, 140–153. https://www.sciencedirect.com/science/article/pii/S0957415819300376.

Chu, S., **Townsend, S.**, **Featherston, C.** & **Kim, H. A.** 2020 Simultaneous layout and topology optimization of curved stiffened panels. In *AIAA AVIATION 2020 FORUM*, p. 3144. AIAA.

Deaton, J. D. & **Grandhi, R. V.** 2014 A survey of structural and multidisciplinary continuum topology optimization: Post 2000. *Structural and Multidisciplinary Optimization* **49** (1), 1–38; doi:10.1007/s00158-013-0956-z.

Dugré, A., **Vadean, A.** & **Chaussée, J.** 2016 Challenges of using topology optimization for the design of pressurized stiffened panels. *Structural and Multidisciplinary Optimization* **53** (2), 303–320; doi:10.1007/s00158-015-1321-1.

Farid, A. M. & **Suh, N. P.**, eds. 2016 *Axiomatic Design in Large Systems*. Springer; doi: 10.1007/978-3-319-32388-6.

Ferrari, F. & **Sigmund, O.** 2019 Revisiting topology optimization with buckling constraints. *Structural and Multidisciplinary Optimization* **59** (5), 1401–1415; doi:10.1007/s00158-019-02253-3.

Gamache, J.-F. 2021 On topology optimization in the generative design framework: Use on aircraft structure design. PhD Thesis, Polytechnique Montréal. https://publications.polymtl.ca/9133/.

Gamache, J.-F., **Vadean, A.**, **Dodane, N.** & **Achiche, S.** 2019 Validating novel stiffened panel configuration generated with topology optimization using non-linear analysis. In *Proceedings of Canadian Aeronautics and Space Institute AERO '19*. Canadian Aeronautics and Space Institute.

Gamache, J.-F., **Vadean, A.**, **Dodane, N.** & **Achiche, S.** 2020 Topology optimization for stiffened panels: A ground structure method. In *IDETC-CIE2020, Volume 11A: 46th Design Automation Conference (DAC)*, p. V11AT11A049. ASME; doi:10.1115/DETC2020-22103.

Gamache, J.-F., **Vadean, A.**, **Dodane, N.** & **Achiche, S.** 2021 On generating stiffening layouts with density-based topology optimization considering buckling. *CEAS Aeronautical Journal* **12** (4), 863–877.

Gamache, J.-F., **Vadean, A.**, **Noirot-Nerin, E.**, **Beaini, D.** & **Achiche, S.** 2018 Image-based truss recognition for density-based topology optimization approach. *Structural and Multidisciplinary Optimization* **58** (6), 2697–2709; doi:10.1007/s00158-018-2028-x.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

Guo, X., **Zhang, W.** & **Zhong, W.** 2014 Doing topology optimization explicitly and geometrically—A new moving morphable components based framework. *Journal of Applied Mechanics* **81** (8), 081009.

Hay, L., **Duffy, A. H. B.**, **McTeague, C.**, **Pidgeon, L. M.**, **Vuletic, T.**, & **Grealy, M.** 2017 A systematic review of protocol studies on conceptual design cognition: Design as search and exploration. *Design Science* **3**, e10. doi:10.1017/dsj.2017.11.

Hoos, H. H. & **Stützle, T.** 2004 *Stochastic Local Search: Foundations and Applications*. Elsevier.

Houston, G., **Quinn, D.**, **Murphy, A.** & **Bron, F.** 2016 Wing panel design with novel skin-buckling containment features. *Journal of Aircraft* **53** (2), 416–426; doi:10.2514/1.C033540.

Houston, G., **Quinn, D.**, **Murphy, A.** & **Bron, F.** 2017 Design rules for stiffened panel buckling containment features. *Thin-Walled Structures* **116**, 69–81. https://linkinghub.elsevier.com/retrieve/pii/S0263823117302586.

HyperWorks 2018 *Detroit*. Altair Engineering.

Kapania, R., **Li, J.** & **Kapoor, H.** 2005 Optimal design of unitized panels with curvilinear stiffeners. In *AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*. American Institute of Aeronautics and Astronautics; doi:10.2514/6.2005-7482.

Kaufmann, M., **Zenkert, D.** & **Wennhage, P.** 2010 Integrated cost/weight optimization of aircraft structures. *Structural and Multidisciplinary Optimization* **41** (2), 325–334; doi:10.1007/s00158-009-0413-1.

Königseder, C. & **Shea, K.** 2014 Strategies for topologic and parametric rule application in automated design synthesis using graph grammars. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 46315, p. V02AT03A007. American Society of Mechanical Engineers.

Krog, L., **Tucker, A.**, **Kemp, M.** & **Boyd, R.** 2004 Topology optimisation of aircraft wing box ribs. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2004-4481.

Layachi, H., **Xu, Y.-m.** & **Bennaceur, M. A.** 2017 Topology optimization and design guidelines of sub-stiffened panels in aerospace applications. *MATEC Web of Conferences* **114**, 03009. doi:10.1051/matecconf/201711403009.

MATLAB 2020 version 9.8.0 (R2020a), The MathWorks Inc., Natick, Massachusetts.

Megson, T. H. G. 2017, *Aircraft Structures for Engineering Students*, 6th edn, Elsevier Aerospace Engineering Series. Butterworth-Heinemann.

Min, G., **Suh, E. S.** & **Hölttä-Otto, K.** 2015 System architecture, level of decomposition, and structural complexity: Analysis and observations. *Journal of Mechanical Design* **138** (2), 021102; doi:10.1115/1.4032091.

Mohebbi, A., **Achiche, S.** & **Baron, L.** 2018 Multi-criteria fuzzy decision support for conceptual evaluation in design of mechatronic systems: A quadrotor design case study. *Research in Engineering Design* **29** (3), 329–349.

Mulani, S. B., **Slemp, W. C.** & **Kapania, R. K.** 2013 EBF3PanelOpt: An optimization framework for curvilinear blade-stiffened panels. *Thin-Walled Structures* **63**, 13–26. https://linkinghub.elsevier.com/retrieve/pii/S0263823112002601.

Nordlund, M., **Lee, T.** & **Kim, S.-G.** 2015 Axiomatic design: 30 years after. In *ASME International Mechanical Engineering Congress and Exposition*, vol. 57588, p. V015T19A009. American Society of Mechanical Engineers.

Optistruct 2018 *Detroid*. Altair Engineering.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press

Puik, E. & **Ceglarek, D.** 2016 A different consideration on information and complexity in axiomatic design. In *Axiomatic Design in Large Systems*, pp. 105–129. Springer.

Rice, R. C., **Jackson, J. L.**, **Bakuckas, J.** & **Thompson, S.** 2003 Metallic materials properties development and standardization (MMPDS). Technical report, Department of Transportation Federal Aviation Administration. https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/PB2003106632.xhtml.

Sigmund, O. 2007 Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization* **33** (4–5), 401–424.

Sigmund, O. & **Maute, K.** 2013 Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization* **48** (6), 1031–1055; doi:10.1007/s00158-013-0978-6.

Sinha, K., **Suh, E. S.** & **de Weck, O.** 2018 Integrative complexity: An alternative measure for system modularity. *Journal of Mechanical Design* **140** (5), 051101. doi: 10.1115/1.4039119.

Suh, N. P. 1990 *The Principles of Design*, Oxford series on advanced manufacturing, vol. 6. Oxford University Press.

Suh, N. P. 2005 *Complexity: Theory and Applications*. Oxford University Press on Demand.

Tom, R. 2021 Find multi-objective pareto front using modified quicksort. Retrieved from March 3rd 2021. https://www.mathworks.com/matlabcentral/fileexchange/73089-find-multi-objective-pareto-front-using-modified-quicksort.

Torii, A. J., **Lopez, R. H.** & **Miguel, L. F.** 2016 Design complexity control in truss optimization. *Structural and Multidisciplinary Optimization* **54** (2), 289–299.

Trefethen, L. N. & **Bau, D.** 1997 *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics.

Tyflopoulos, E., **Tollnes, F. D.**, **Steinert, M.**, **Olsen, A.** 2018 State of the art of generative design and topology optimization and potential research needs. In *DS 91: Proceedings of NordDesign 2018, Linköping, Sweden, 14–17th August 2018*. Design Society.

van Dijk, N. P., **Maute, K.**, **Langelaar, M.** & **van Keulen, F.** 2013 Level-set methods for structural topology optimization: A review. *Structural and Multidisciplinary Optimization* **48** (3), 437–472. doi:10.1007/s00158-013-0912-y.

Venkataraman, S. & **Haftka, R. T.** 2004 Structural optimization complexity: What has Moore's law done for us? *Structural and Multidisciplinary Optimization* **28** (6), 375–387.

Walbrun, S., **Witzgall, C.** & **Wartzack, S.** 2019 A rapid CAE-based design method for modular hybrid truss structures. *Design Science* **5**, e27.

Warwick, B. T., **Mechefske, C. K.** & **Kim, I. Y.** 2019 Topology optimization of a pre-stiffened aircraft bulkhead. *Structural and Multidisciplinary Optimization* **60** (4), 1667–1685. doi:10.1007/s00158-019-02284-w.

Zhang, W., **Liu, Y.**, **Wei, P.**, **Zhu, Y.** & **Guo, X.** 2017 Explicit control of structural complexity in topology optimization. *Computer Methods in Applied Mechanics and Engineering* **324**, 149–169. https://linkinghub.elsevier.com/retrieve/pii/S0045782516317443.

Zhang, W., **Zhou, J.**, **Zhu, Y.** & **Guo, X.** 2017 Structural complexity control in topology optimization via moving morphable component (MMC) approach. *Structural and Multidisciplinary Optimization* **56** (3), 535–552.

Zhu, J.-H., **Zhang, W.-H.** & **Xia, L.** 2016 Topology optimization in aircraft and aerospace structures design. *Archives of Computational Methods in Engineering* **23** (4), 595–622. doi:10.1007/s11831-015-9151-2.

https://doi.org/10.1017/dsj.2023.12 Published online by Cambridge University Press