Generative Design as a Configuration Problem

Jonathan Raines[1,*], David Barton[1] and Ben Hicks[1]

[1]University of Bristol, UK

Abstract

Generative design techniques such as topology optimisation can produce lightweight structures that significantly reduce emissions in aerospace and automotive applications. However, a gap exists between computationally generated designs and manufacturable parts: while topology optimisation produces optimal shapes for 3D printing or single-piece machining, industrial manufacturing relies on assemblies of standard components using processes like welding, stamping, and cutting. This paper formalises the problem of approximating topology-optimised designs using off-the-shelf parts and conventional manufacturing processes as a configuration problem. We define this problem as finding high-performing configurations of parts from industrial catalogues, modified by available processes, that minimise cost and weight while maximising geometric similarity to the target design. The key challenges include managing discrete part catalogues, representing complex 3D geometries, navigating solution spaces that grow exponentially, and handling mixed discrete-continuous optimisation variables. By framing generative design approximation as a configuration problem, we aim to bridge the gap between computational design tools and the reality of industrial manufacturing.

Keywords

Generative design, Topology Optimization, Configuration Problem, Manufacturing, Discrete Optimization, Standard Parts, Weight Optimization, Lightweighting, Design for Manufacturing

1. Introduction

The aerospace and automotive industries are both significant contributors to climate change. Aerospace contributes 2.5 % [1] of global carbon-dioxide emissions, and road passenger transport 10.8 % [2]. In both industries, lightweighting is a key means for reducing emissions. A study by the International Transport Forum concluded that if the mass of cars could be reduced back to 1970s levels (a 40 % reduction), then CO_2 emissions could be reduced by an additional 90 Mt (18 %) [3]. Topology optimisation techniques such as Solid Isotropic Material with Penalization (SIMP) [4] can search for the lightest part that meets a loading condition. Topology optimisation (and more broadly, generative design) tools are available in commercial software such as Autodesk Fusion [5] and COMSOL [6]. However, parts designed using these methods are not commonly used in these industries or other commercial projects due to the following limitations. Firstly, the geometry of the parts generally requires 3D printing, casting or machining the part in a single piece. Aerospace and automotive are safety-critical applications, inhibiting the adoption of 3D printing for structural parts. The 3D printing process can introduce microscopic cracks, leading to unacceptable part strength variations. Secondly, both industries need to manufacture parts at scale, and 3D printing costs do not scale with production volume. Machining and casting are practical at high volumes, but are not practical for every part. The generatively-designed bracket shown in Figure 1 would be economically infeasible to machine due to its complexity and the proportion of the blank that would be scrap.

The following industry cases illustrate the gap between the objectives of topology optimisation and industrial use. In 2016, Airbus unveiled a prototype "bionic partition". Created using generative design, the 3D printed design was 50 % lighter [7]. The prototype exceeded the capacity of 3D printers at the time, so the prototype was made in 122 parts and fastened together [8]. However, a later news report revealed that the approach was abandoned (due to manufacturing cost [9]), and the first installed

ConfWS'25: 27th International Workshop on Configuration, Oct 25–26, 2025, Bologna, Italy
*Corresponding author.
jonathan.raines@bristol.ac.uk (J. Raines); david.barton@bristol.ac.uk (D. Barton); ben.hicks@bristol.ac.uk (B. Hicks)
https://JonathanRaines.github.io (J. Raines); https://cityinthesky.co.uk (D. Barton)
0009-0001-8325-2005 (J. Raines); 0000-0002-0595-4239 (D. Barton); 0000-0001-8294-4694 (B. Hicks)

version was "a sandwich panel with a honeycomb core and carbon fibres (CFRP)" [10]. In 2017, Autodesk collaborated with the *Bandito Brothers* to create a hotrod car chassis [11]. The team used Autodesk's project DreamCatcher [12] to design the chassis using telemetry data from a prototype. They then manually approximated the design in such a way that it could be constructed from welded tubing [13, 14]. The team aimed to fully 3D print the chassis but, at the time of writing, we could not find a record of them succeeding.

The benefits of generative design cannot be realised when it is limited to a small subset of manufacturing processes available. Automating the manual approximation of a generated design to use off-the-shelf parts and processes would bridge the gap between available tools and industry use. This would properly integrate generative design into engineers' toolboxes as one way to lightweight parts and reduce emissions. To this end, we present the approximation process as a configuration problem.

2. Problem Definition

In this section, we present a generalised form of the problem and provide illustrative examples.

The Manufacturing Problem

Instance:

- a target shape generated using topology optimisation
- a library of parts
- a set of processes that can modify instances of parts in the library
- one or more optimisation criteria
- a target production volume

Task:
Find the configuration of part instances, each possibly modified by a sequence of processes, that minimises the optimisation criteria.

2.1. Configuration Definition

A configuration is a directed tree where:

- **Nodes** are manufacturing processes with parameters
- **Leaf nodes** are processes adding parts from a part library
- **Edges** are the flow of parts
- **Root** is the final assembly process, which outputs the completed product

An example configuration is shown in Figure 2.

2.2. Validity Constraints

A configuration is valid if:

- The tree is connected (single root),
- there are no intersecting parts,
- all joints/connections are physically realisable,
- each process node has compatible incoming edges,
- and the parameters of each process node are feasible.

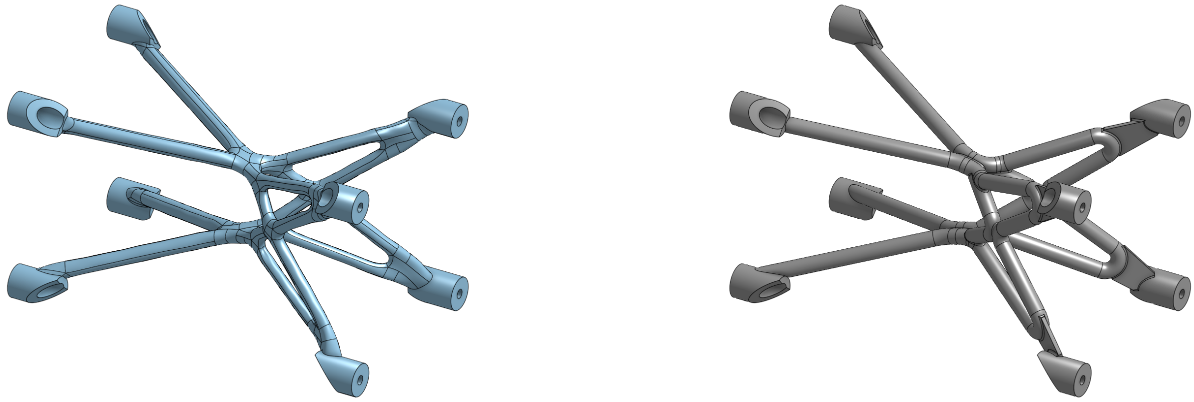

Figure 1: A generatively designed bracket (left) and a version redesigned to be manufactured at volume using CNC bending, sheet metal cutting and turning (right).

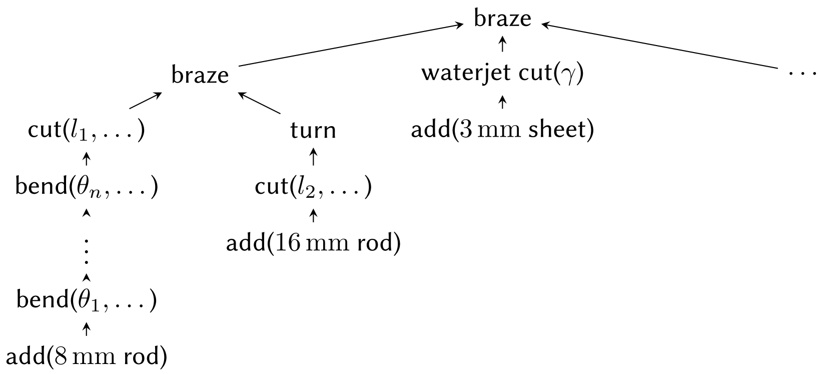

Figure 2: An abridged example configuration tree for the bracket shown in Figure 1 (right). The left-hand branch represents the main body of the bracket, the second branch the bosses at the corners, and the third branch the plates used to connect the bosses on the right-hand side.

2.3. Evaluation Criteria

Finding valid configurations is not sufficient. Many will be manufacturable but perform poorly against a given objective. Functional evaluations, such as Finite Element Analysis (FEA) and real-world testing are required in safety-critical industries such as aerospace and automotive. However, they are expensive (in terms of compute, time and resources). For searching through valid configurations, or training a system to generate them in a data-driven approach, a proxy is required. A configuration can be evaluated based on its geometric similarity to a target form generated using Topology Optimization. This can be done by instantiating 3D models of the stock material and applying the modifications of the processes in the tree. The resulting shape can be compared to the target using the Hausdorff distance (the maximum distance between any point on one shape and its nearest point on the other shape). Generally, this approach can be thought of as using a continuous representation and gradient descent to find a design, then approxmating that design with discrete operations. Using a shape-based metric also offers flexibility. A user of a configuration generator could provide the output of available generative design and topology optimisation tools, or model a freeform shape by hand. A drawback to this approach is that small changes in geometry can lead to large changes in deflection or peak stress. As such, if such a system were being used in the aerospace or automotive industries, configurations suggested by the tool of interest to a designer would be evaluated using functional evaluations such as FEA.

Many valid configurations will approximate the target shape, but may not be optimal in terms of cost or production volume. As such, a cost objective must also be applied. This can be achieved by assigning a cost to each part in the library and summing the costs of the parts used in the configuration.

Each process can be costed by assigning a set-up cost and an operation cost. The set-up cost is incurred once if the process is used in a configuration, and the operation cost is incurred for every instance of a process node in a configuration.

2.4. Data

The SELTO dataset [15] contains 9848 example parts generated using SIMP [4]. Each example is comprised of a voxel representation of the generated part, as well as the forces and boundary conditions used to generate it.

3. Challenges

The manufacturing problem presents three key challenges.

First, the number of possible configurations grows exponentially with the size of the component library, the process library, the number of nodes added to a configuration, and the number of design variables. Industrial part catalogues contain thousands of components. This is often referred to as the curse of dimensionality.

Second, the challenge of representing a configuration. The parts and processes can be represented as trees as described in Section 2.1. However, the configuration also represents a 3D shape. It is necessary to generate and check the 3D shape for self-intersection. A configuration that appears valid based on the tree structure may still be invalid due to a self-intersection. Consider a bar that has been bent 270 degrees. Potential 3D representations include Signed Distance Functions (SDFs) or Boundary Representations (B-reps).

Finally, the problem combines discrete and continuous variables, for example, tubing comes in fixed diameters but can be cut to any length, and sheet materials have standard thicknesses but arbitrary cut shapes. The number of variables also varies depending on the configuration. A bend will have a different number of parameters than a cut.

4. Related Work

In this section, we describe related problems and the ongoing research into them. We aim to differentiate this problem, as well as explain the inspiration for the research avenues detailed in the next section.

4.1. Configuration Design

Mittal and Frayman [16] defined a general framework for configuration design. The problem presented in this article adds the complication that components can be modified using a library of operations before being combined. As highlighted in [17], representing engineering components in a reusable manner has proven challenging. The problem presented in this article limits the component library to stock materials that can be represented as a set of parametric shapes.

4.2. Manufacturing Constraints for Topology Optimisation

Researchers have modified density-based methods (such as SIMP [4]) to respect minimum feature size and overhang constraints of 3D printing [18, 19, 20] and to impose constraints for 2.5 and 5 axis machining, using projections to penalize areas inaccessible to the tool during optimisation [21]. Greminger [22] adopted a data-driven approach, training a Generative Adversarial Network (GAN) on examples of machinable parts. These processes make progress towards manufacturability. However, the assumption of a solid isotropic material is intrinsic, so separate parts that may have been pre-processed cannot be represented.

4.3. Analog Circuit Synthesis

Circuit topology synthesis shares the goal of configuring a library of parts. Typically, the circuit is represented as a graph, with parts as nodes and connections as edges (e.g. [23]). Gao et al. [24] argued that this representation is ambiguous, as parts have pins with different functionalities. They proposed adding pins as an additional node type to the graph, allowing for explicit pin-to-pin connections. They also demonstrated the effectiveness of converting the graph to a set of sequences using Eulerian walks and applying a Transformer [25] in a system they dubbed AnalogGenie.

4.4. Program Synthesis

Program synthesis is the search for a program that generates a desired output. In the context of 3D modelling, the output is commonly a target shape. Before the popularisation of Large Language Models (LLMs), Domain Specific Languages (DSLs) were used to constrain the search space to a tractable size. Jones et al. [26] proposed a DSL called ShapeAssembly. They trained a hierarchical Variational Autoencoder (VAE) on ShapeAssembly programs reverse engineered from assemblies in PartNet [27]. They could then generate new programs, and thus assemblies, by sampling from the latent space of programs. Ellis et al. [28] proposed DreamCoder, that could expand its own DSL through a process of self-improvement.

A limitation of DSLs is that they constrain what can be expressed. We note that this may actually be a useful property in the context of ensuring manufacturability. However, to overcome this, researchers have recently favoured using LLMs to generate programs in Turing-complete languages, for example, Python. Notable work related to geometry generation includes CAD-CODER, which takes an image of a part and produces a parametric Computer-Aided Design (CAD) model [29]. The approach makes use of a Vision Language Model (VLM) to generate Python code that generates the model. While such approaches demonstrate the potential to convert generatively designed parts into parametric CAD models, they do not fully address the manufacturing problem presented in this paper, as CAD models are not necessarily manufacturable.

5. Research Avenues

We identify two broad categories of approaches for addressing the configuration problem presented in this paper. The first involves searching for a configuration directly, which we term an output-centric approach. The second focuses on searching for a program that generates a configuration, which we refer to as a program-centric approach.

For output-centric approaches, several promising directions emerge. Graph generation techniques offer a way to generate configurations directly. Transformer-based models show promise, as demonstrated in AnalogGenie [24].

The configuration described in Section 2.1 can be viewed as the Abstract Syntax Tree (AST) of a program. One program-centric approach is to use an LLM to write the program using a supplied library of functions. Another approach is to generate a program that searches for a configuration. Both approaches can be further enhanced by employing evolutionary algorithms on the output programs, feeding high-performing programs back into the model for iteration.

Acknowledgments

Jonathan Raines acknowledges funding from the UKRI for a Centre for Doctoral Training studentship in Interactive Artificial Intelligence at the University of Bristol. (EP/S022937/1)

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude Sonnet 4 (Anthropic) and Grammarly for: drafting content, paraphrasing and rewording, improving writing style, abstract drafting, grammar and spell check, peer review simulation, and content enhancement. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] H. Ritchie, What share of global CO_2 emissions come from aviation?, Our World in Data (2024). URL: https://ourworldindata.org/global-aviation-emissions, accessed: 2025-06-17.

[2] H. Ritchie, Cars, planes, trains: Where do CO_2 emissions from transport come from?, Our World in Data (2020). URL: https://ourworldindata.org/co2-emissions-from-transport, accessed: 2025-06-17.

[3] E. Windisch, V. Benezech, G. Chen, J. Kauppila, Lightening Up: How Less Heavy Vehicles Can Help Cut CO2 Emissions, Case-Specific Policy Analysis Report OECD/ITF 2017, The International Transport Forum, 2017. URL: https://www.itf-oecd.org/sites/default/files/docs/less-heavy-vehicles-cut-co2-emissions.pdf, accessed: 2025-06-17.

[4] M. P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, Computer Methods in Applied Mechanics and Engineering 71 (1988) 197–224. doi:10.1016/0045-7825(88)90086-2.

[5] Autodesk, Autodesk Fusion, 2024. URL: https://www.autodesk.com/uk/products/fusion-360/, accessed: 2025-07-11.

[6] COMSOL, COMSOL Multiphysics 6.0, 2021. URL: https://www.comsol.com/comsol-multiphysics, accessed: 2025-07-11.

[7] D. Nagy, D. Zhao, D. Benjamin, Nature-Based Hybrid Computational Geometry System for Optimizing Component Structure, in: K. De Rycke, C. Gengnagel, O. Baverel, J. Burry, C. Mueller, M. M. Nguyen, P. Rahm, M. R. Thomsen (Eds.), Humanizing Digital Reality: Design Modelling Symposium Paris 2017, Springer, Singapore, 2018, pp. 167–176. doi:10.1007/978-981-10-6611-5_15.

[8] K. Walmsley, Autodesk and Airbus generatively design a 3D-printed bionic partition for the A320, 2016. URL: https://www.keanw.com/2016/02/autodesk-and-airbus-generatively-design-a-3d-printed-bionic-partition-for-the-a320.html, accessed: 2025-06-18.

[9] B. Koenig, 3D Printing Cleared for Takeoff, 2020. URL: https://www.sme.org/technologies/articles/2020/january/3d-printing-cleared-for-takeoff/, accessed: 2025-06-18.

[10] M. Kirby, Press Release: Bucher's Bionic Partition installed successfully on A320 -, 2021. URL: https://runwaygirlnetwork.com/2021/04/buchers-bionic-partition-installed-successfully-a320/, accessed: 2025-06-17.

[11] TED, The incredible inventions of intuitive AI | Maurice Conti, 2017. URL: https://youtu.be/aR5N2Jl8k14?t=782, accessed: 2025-09-25.

[12] G. Fitzmaurice, Project Dreamcatcher: Generative Design Solutions in CAD, 2017. URL: https://www.research.autodesk.com/projects/project-dreamcatcher/, accessed: 2025-07-02.

[13] DEVELOP3D, Garage bandits, 2017. URL: https://develop3d.com/profiles/garage-bandits-industrial-design-hackrod-automotive/, accessed: 2025-06-18.

[14] D. Terdiman, Inside The Hack Rod, The World's First AI-Designed Car, 2015. URL: https://www.fastcompany.com/3054028/inside-the-hack-rod-the-worlds-first-ai-designed-car/, accessed: 2025-06-18.

[15] S. Dittmer, D. Erzmann, H. Harms, R. Falck, M. Gosch, SELTO Dataset, 2023. URL: https://zenodo.org/records/7781392.

[16] S. Mittal, F. Frayman, Towards a generic model of configuraton tasks., in: IJCAI, volume 89, Citeseer, 1989, pp. 1395–1401.

[17] R. Malik, M. Ahmad, J. Vepsäläinen, Modular representation of components to enable generative engineering, in: 2024 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC), 2024, pp. 1–9. doi:10.1109/ESARS-ITEC60450.2024.10819828.

[18] B. Barroqueiro, A. Andrade-Campos, R. a. F. Valente, Designing Self Supported SLM Structures via Topology Optimization, Journal of Manufacturing and Materials Processing 3 (2019) 68. doi:10.3390/jmmp3030068.

[19] E. Fernández, K.-k. Yang, S. Koppen, P. Alarcón, S. Bauduin, P. Duysinx, Imposing minimum and maximum member size, minimum cavity size, and minimum separation distance between solid members in topology optimization, Computer Methods in Applied Mechanics and Engineering 368 (2020) 113157. doi:10.1016/j.cma.2020.113157.

[20] B. M. Weiss, J. M. Hamel, M. A. Ganter, D. W. Storti, Data-Driven Additive Manufacturing Constraints for Topology Optimization, in: ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, 2018. doi:10.1115/DETC2018-85391.

[21] H. Y. Lee, M. Zhu, J. K. Guest, Topology optimization considering multi-axis machining constraints using projection methods, Computer Methods in Applied Mechanics and Engineering 390 (2022) 114464. doi:10.1016/j.cma.2021.114464.

[22] M. Greminger, Generative Adversarial Networks With Synthetic Training Data for Enforcing Manufacturing Constraints on Topology Optimization, in: ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, 2020. doi:10.1115/DETC2020-22399.

[23] Z. Dong, W. Cao, M. Zhang, D. Tao, Y. Chen, X. Zhang, CktGNN: Circuit Graph Neural Network for Electronic Design Automation, 2024. doi:10.48550/arXiv.2308.16406. arXiv:2308.16406.

[24] J. Gao, W. Cao, J. Yang, X. Zhang, AnalogGenie: A Generative Engine for Automatic Discovery of Analog Circuit Topologies, 2025. URL: https://arxiv.org/abs/2503.00205v1.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, 2017. doi:10.48550/arXiv.1706.03762. arXiv:1706.03762v1.

[26] R. K. Jones, T. Barton, X. Xu, K. Wang, E. Jiang, P. Guerrero, N. J. Mitra, D. Ritchie, ShapeAssembly: Learning to generate programs for 3D shape structure synthesis, ACM Trans. Graph. 39 (2020) 234:1–234:20. doi:10.1145/3414685.3417812.

[27] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, H. Su, PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[28] K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, J. B. Tenenbaum, DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning, in: Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI 2021, Association for Computing Machinery, New York, NY, USA, 2021, pp. 835–850. doi:10.1145/3453483.3454080.

[29] A. C. Doris, M. F. Alam, A. H. Nobari, F. Ahmed, CAD-Coder: An Open-Source Vision-Language Model for Computer-Aided Design Code Generation, 2025. doi:10.48550/arXiv.2505.14646. arXiv:2505.14646.