



# Symmetry-based decomposition for optimised parallelisation in 3D printing processes

Hayley Hatton<sup>1</sup> · Muhammad Khalid<sup>1</sup> · Umar Manzoor<sup>2</sup> · John Murray<sup>3</sup>

Received: 26 August 2022 / Accepted: 2 March 2023 / Published online: 9 June 2023  
© The Author(s) 2023

## Abstract

Current research in 3D printing focuses on improving printing performance through various techniques, including decomposition, but targets only single printers. With improved hardware costs increasing printer availability, more situations can arise involving a multitude of printers, which offers substantially more throughput in combination that may not be best utilised by current decomposition approaches. A novel approach to 3D printing is introduced that attempts to exploit this as a means of significantly increasing the speed of printing models. This was approached as a problem akin to the parallel delegation of computation tasks in a multi-core environment, where optimal performance involves computation load being distributed as evenly as possible. To achieve this, a decomposition framework was designed that combines recursive symmetric slicing with a hybrid tree-based analytical and greedy strategy to optimally minimise the maximum volume of subparts assigned to the set of printers. Experimental evaluation of the algorithm was performed to compare our approach to printing models normally (“in serial”) as a control. The algorithm was subjected to a range of models and a varying quantity of printers in parallel, with printer parameters held constant, and yielded mixed results. Larger, simpler, and more symmetric objects exhibited more significant and reliable improvements in fabrication duration at larger amounts of parallelisation than smaller, more complex, or more asymmetric objects.

**Keywords** 3D printing · Decomposition · Parallelisation · Multiple printers

## 1 Introduction

In the past decade, the technology around additive manufacturing has improved significantly, with fused filament modelling (“3D Printing”) becoming sufficiently efficient and intuitive to be deployed on a wide scale, from commercial use by household consumers to professional uses, e.g., in rapid prototyping and space engineering [25]. These improvements have involved advances in both printer technology, increasing the performance of printing itself, and technology that uses the printers, increasing the efficiency

of the fabrication process [29]. Nonetheless, 3D printing is still a young technology with limitations inherent to it. Costs of filament and the hardware, printing failures, discrepancies between the digital and fabricated forms, and a generally long time to print larger models at even fast settings (over a day) are some examples of these common limitations [28]. This paper focuses on the latter limitation.

Traditional methods of decreasing the time to print larger models usually involve some combination of increasing the printing rate at the cost of decreasing quality and reducing the density of the obscured internal space [21], or altering the orientation of the model to reduce support structures [37]. Current research in the area has largely focused on the computational optimisation of these parameters in various ways, often in the context of specific domains (Fig. 1).

But, in the context of an increasing availability of 3D printers, as is often encountered in lab settings, the decomposition research could be suboptimal in its use of the available resources, causing some printers to be doing large quantities of work and others to be sitting idle for significant lengths of time. This creates a need for an algorithm that is specifically tailored for optimising printing in parallel to make efficient

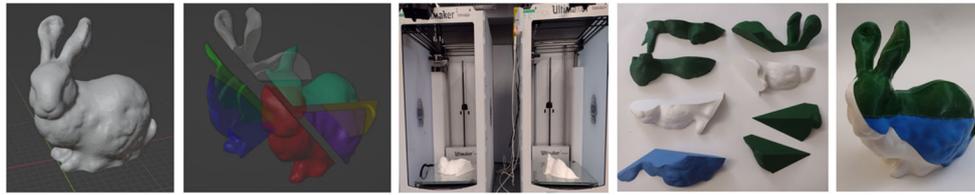
✉ Muhammad Khalid  
m.khalid@hull.ac.uk

Hayley Hatton  
h.a.hatton-2011@hull.ac.uk

<sup>1</sup> School of Computer Science,  
University of Hull, Hull, UK

<sup>2</sup> Department of Computer Science, Aston University,  
Birmingham, UK

<sup>3</sup> Faculty of Technology, University of Sunderland,  
Sunderland, UK



**Fig. 1** An overview of the approach applied to the Stanford Bunny

use of these resources, by devising logic that sees decomposition in terms of balancing printer utilisation, rather than minimising an overall aggregate printing time of parts.

In this paper a new approach is presented for improving the fabrication time of producing models on commercial FFM 3D printers by involving the use of multiple 3D printers in parallel. Current approaches that employ decomposition do so for volume packing, support reduction, or optimisation of printing time in serial; the aim of this research is to take advantage of the theoretical increased throughput afforded by multiple 3D printers to improve fabrication times in parallel. An algorithm is proposed that exploits reflective symmetry to partition models such that the distribution of the decomposed parts across the specified available printers is optimised for throughput, with the workload balanced in a way that utilised printers are not left idle with others still printing. This use of symmetry is coupled with a hybrid analytical-greedy algorithm that makes use of a maximin strategy to achieve an optimal use of a given amount of printers for fabricating an input model with minimal overhead computation time.

Various challenges were encountered in the process of this research. Initially, a pure greedy approach was modelled, but situations were encountered in which making suboptimal cuts with worse symmetries early on could result in better cuts being available on constituent parts that would result in a more optimal minimisation of the maximum printing time. Consequently, an analytical method based on a modified BSP tree representing multiple possible symmetric cut options at each stage was developed, however, this quickly experienced problems due to combinatorial explosions arising.

This led to the development of the proposed hybrid model: analytical early on, where models are larger and the improvements from a larger search scope are more significant, moving to greedy after a few iterations where models are smaller, to avoid the combinatorial explosion issues.

Early on, the process of decomposition also seemed quite slow as it was written in serial, and was substantially improved by taking advantage of computational parallelisation. The slicing function was the bottleneck, so, because slicing logic for options across a single model are not dependent on each other, parallelisation of this code enabled full utilisation of the multi-core machines available, bringing deeper subdivision times down from hours to minutes, or minutes to seconds.

This framework was demonstrated across a small randomised sample of the *Thingi10K* dataset extended with additional models to verify its application across a variety of shapes and sizes. A controlled range of parallelisation configurations are selected to test the framework's performance and limitations, with the times involved in computing, printing, and assembling the models taken to both validate the performance of the framework in the physical world, and determine the effect of diminishing returns.

Our results indicated that larger, simpler, more convex prints benefit far more from this approach to parallelisation than smaller, thinner, sparser, or more complex prints, with the former class of geometry outpacing diminishing returns up to around 8+ parallel printers, and the latter classes finding no improvement beyond 4. This is especially so when qualitative feedback is taken, with these latter classes looking considerably worse after substantial decomposition compared to the former class.

Thus, in summary, the main contributions of this research are:

- A proposed framework that optimises the net parallel fabrication time of models through the use of recursive symmetric decomposition as part of a mixed analytical and greedy algorithm that minimises the maximum part printing time in order to best utilise a specified amount of equivalent printers.
- A quantitative evaluation of symmetry-based parallel printing across a range of models at varying degrees of parallelisation and in comparison to serial printing. This accounts for both the diminishing returns of increasing parallelisation, and the increasing costs of computation and assembly with higher rates of decomposition, to test the performance in the physical world.

## 2 Related work

Decomposition in 3D printing has already been employed for improving the performance of 3D printing, and is adjacent or antecedent to the research presented in this paper.

**Large object fitting and optimisation** Research in this domain primarily focuses on larger models. Commercial 3D

printers are usually relatively small, with their build volumes constraining the size of models that can be printed. Models larger than this volume have to be split down and assembled in order to be realised in full, so optimising the process of decomposition for printing in a given volume is a rich domain.

“*Chopper*” [19] approaches this objective by utilising recursive planar cuts, with split parts represented using binary space partition (“BSP”) trees. The use of BSP trees with a beam search instead of arbitrary cuts led to improved performance in computation time compared to a depth-first search. This process of recursive cutting is repeated until all parts of the model fit into the printing volume. *Chopper* influenced subsequent decomposition for performance, for example *Interactive Partitioning of 3D Models into Printable Parts* [12], which instrumented the algorithm such that users can control the constraints on the model and the decomposition logic to improve performance in domain-specific ways.

*Chopper* provides the most significant advantages in parallel printing, in that its beam search algorithm is swift, and it employs symmetric cuts to preserve visual quality, which tends to prefer good use of printer volume and is therefore immediately useful in parallel domains. However, in the same approach of preserving quality, its objective function is equally tailored towards favouring large cross-sectional areas, permitting the introduction of connections, and limiting smaller, separated parts, which deviates the algorithm from the more pure symmetries that would be best in parallel use cases.

Substantially, the primary limitation of *Chopper* in the parallel context is that, as a fitting algorithm, its primary constraint is the bounding box size of the model. Once the model fits, the algorithm is complete. The worst case scenario here is the model already fits in a large printer, but, e.g., there are eight printers available and time is of the essence, so no cuts are made and a lengthy print ensues.

“*PackMerger*” [35] uses another method of decomposition to achieve the same result by extracting a “volumetric shell” from the mesh and segmenting this by seeding regions across it, and growing and merging these regions with an objective of reducing printing time and overhanging areas. These decomposed regions are then arranged and packed using a height-field based packing algorithm. The process proved more efficient, but was particularly limited on FDM printers, or on intricate models.

The advantages of *PackMerger*, if it could be used in the parallel domain, are that it segments models using tetrahedral clustering with an aim to produce models smaller than the printing area at roughly equal volume, which could lead to decently efficient parallel printing. It also preserves visual quality similar to *Chopper* by maximising cross-sectional areas.

However, the disadvantages follow with it being used for packing into a single print. For example, to optimise single prints, parts are further merged to make better use of the remaining printer space and decrease the amount of external surface area that can slow down prints, but in a parallel setting, this could result in unequal distribution of printing load across the multiple printers and therefore lead to suboptimal parallel printing.

“*Dapper*” (“*Decompose-and-Pack*”) [4] approaches the problem of fitting large models into a printer by utilising pyramidal primitives, which are easy to pack together, and progressively packing these primitives into a printing volume similar to *Tetris* in such a way that print and assembly times are reduced. A coarse decomposition is first applied creating a small number of pieces, and a prioritised beam search is used to determine an optimum packing solution. Bounds are used to minimise cuts that produce narrow cross-sections, to ensure efficient assembly. This even influenced subtractive manufacturing, with *VDAC: Volume-Decompose-and-Carve for Subtractive Manufacturing* [20].

*Dapper* offers significant advantages over *PackMerger* in parallel domains by removing the merge steps and focusing on packing the decomposed parts. However, the remaining advantages over *PackMerger* — such as printing time improvements, execution time efficiencies, and the pyramidal decomposition allowing for tighter fits into a volume — largely relate to the packing potential.

This exacerbates the drawbacks in the parallel domain: by incorporating more complex objectives than volume, and focusing on the collective parts efficient use of a single printer’s volume, it is more likely to produce parts less efficient in parallel settings.

A slightly more unorthodox approach imports another technology to improve performance in this domain. “*Cofi-Fab*” [31] selectively decomposes a model such that more detailed exterior facets are printed on a 3D printer, while sections of internal structure and smooth external shell are created on a laser cutter. This exploits the detail of 3D printing where it is needed, while exploiting the relative substantial speed of laser cutting where it can be afforded without highly compromising the visual quality.

The main advantages of this are that it allows the slower 3D printer to take care of external details, while the faster laser cutter takes care of the internal scaffolding. In parallel contexts, this distribution of work is a good idea. However, drawbacks are that it involves having laser cutters as well as 3D printers, and that — as an algorithm not designed with parallelism in mind — attention is not made to how much the load is balanced to maximise throughput. The 3D shell pieces are also cut in ways that are incident to the polyhedral assembly scaffolding, which is useful for stability and visual quality, but does not prioritise the distributing of printing load

evenly across printers.

So, while these are applicable within the setting of single printers, and could theoretically have their decomposed parts printed on multiple printers to achieve parallelism, it is likely that these result in suboptimal use of multiple printers in parallel. Packing imposes different requirements on geometry than does parallelisation, prioritising fitting shapes into a set volume, rather than producing constituent parts with lower maximum print times. Even CofiFab, which looks at using 3D printers and laser cutters, does not consider using them in parallel, instead using the laser cutter to merely optimise certain aspects of a given model. Particularly, all of these largely solve the problem of models too big to print in one go, focusing less on whether models that could be printed in one go could actually be printed faster in parallel.

**Improved efficiency, fidelity, and material use** Other research focuses on broadly improving many metrics of current 3D printing processes in tandem using decomposition, without specific targeting for volume fitting or support removal. This is an area broadly researched in additive manufacturing, for example in optimising the use of hardware in improving the pathing of printing heads [6, 27, 36] or improving infill structures [42], or in improving the characteristics of single prints geometrically [3, 43]. Nonetheless, decomposition has been found to offer improvements in this domain in a variety of ways.

“*Near-convex decomposition and layering for efficient 3D printing*” [45] attempts to improve upon the general metrics of 3D printing together with decomposition using a multi-phase pipeline that clusters mesh triangles and orients them optimally, followed by splitting and merging clusters using a relaxed convex decomposition strategy that minimises an energy function in order to create subdivisions that decreases printing time and material costs while retaining visual fidelity.

“*Part decomposition and evaluation based on standard design guidelines for additive manufacturability and assembly*” [26] has recently improved upon this process by employing a genetic algorithm for part decomposition, guiding it with standardised design guidelines for additive manufacturing, with improved material cost usage and build times. Other research has statistically evaluated the application of part decomposition via a voxelisation method to save material and energy, while optimising the utilisation of the build plate by interpreting it as equally sized parallel sub-spaces for decomposing models into [8].

Research in the use of additive manufacturing for engine blades for aeroplanes has made domain-specific use of decomposition to improve the aspects of manufacturing [17], by applying a self-adaptive spectral clustering algorithm to break a blade down into sub-blocks, such that expensive powder can be minimised, printing time can be improved,

and supports can be avoided. This also involves a stage of algorithmically determining the optimal amount of decomposition through a clustering algorithm.

Another specific application of decomposition for performance improvement involves an algorithm that decomposes shapes into opposing height fields, which can then be used to print rigid casts for then manufacturing the object, rather than printing and assembly the object specifically [1].

Another approach is seen in “*Axis-Aligned Height-Field Block Decomposition of 3D Shapes*” [24], which uses axis-aligned bounding boxes seeded at the vertices of the object and grown or shrunk, or split or combined with overlapping boxes in preference for larger boxes, to create a series of subdivisions that preserves the details of the outer surface, without generating overhang. The flat sides of the cuts produced by the boxes are guaranteed to be vertical or horizontal, with the details oriented up in the Z axis, while internal structures can be omitted entirely, increasing material efficiency and reducing print time.

Within the context of parallelisation for speed, these share similar limitations to the previous section: a focus on single printers, with printing time being interpreted as the summation of printing tasks, rather than parallel printing times. The papers focusing on rigid cast production and aeroplane engine blades are honed in to very specific domains, where the algorithms are tailored for a very specific type of model in a unique setting, and may not be applicable to generalised parallelisation. The other research is more generalised, but also focuses on improving multiple aspects of the process at once. This could be suboptimal if the objective is the minimum raw parallel printing time, with the informed possible cost of fidelity and additional material consumption.

**Support reduction** A wide range of research attempts to improve the printing performance of models by reducing or eliminating support structures, which serve to increase print time and material consumption, and require post-processing at the cost of time and introducing surface artefacts [14]. While methods continue to be researched which achieve this, such as using conical layer blocks to print overhangs without supports [39], by optimising orientation to reduce support [37], by changing printing parameters such as cooling settings, printing speed, and temperature to increase the tolerable overhang angle size [13], or even using GPUs for parallelising genetic algorithms to improve computational performance [18], these are generally restricted to single print jobs. Nonetheless, decomposition also finds use in this domain.

“*Clever Supports*” [34] initially approached this through the use of a global optimisation to determine the rotation of an input model that reduces support requirements the most as part of a broader algorithm, while “*Object Partitioning for Support-Free 3D-Printing*” [15] employs decomposition

for this approach specifically, discovering subparts and splitting junctions, and recursively and stochastically placing and rotating cutting planes to increase the printable areas, while identifying common geometric patterns related to printing difficulties and supports to mitigate.

Other recent research employs decomposition in combination with multi-directional 3D printing to also reduce supports more specifically. “*General Support-Effective Decomposition for Multi-Directional 3D Printing*” [40] uses a beam-guided search algorithm to position clipping planes for decomposing a model, with the intent of eliminating supports and allowing the multi-directional printer to assemble the model in-place.

*Towards Support-Free 3D Printing: A Skeletal Approach for Partitioning Models* [38] takes a different approach, analysing the shell of models to calculate a “curved skeleton”. Analysis can be performed on the skeleton to determine overhang, and a stochastic approach is taken to assess the relationship of branches with nodes to determine the optimal partitions of subgraphs for decreasing overhang and therefore reduce support structures.

“*Learning to Accelerate Decomposition for Multi-Directional 3D Printing*” [41] approaches this similarly, using a beam-guided search to train a neural network to determine decompositions for support reduction on five-axis 3D printers at 3x faster computational speeds.

“*Oriented to Multi-Branched Structure Unsupported 3D Printing Method Research*” [11] uses a clustering algorithm to begin the coarse partitioning of models, then using clipping planes for further decomposition, in the context of eliminating supports in branched models on five-axis 3D printers.

“*Near support-free multi-directional 3D printing via global-optimal decomposition*” [7] also minimises supports by minimising overhanging surface area specific to the use of five-axis 3D printers using a global optimisation applied to decomposition.

*Evolutionary Decomposition for 3D Printing* [44] similarly employs machine learning techniques (in this case, genetic algorithms) to reduce supports through optimising decomposition, identifying two algorithms — Multi-Objective Genetic Algorithm and Covariant Matrix Adaptation Evolution Strategy — for use, finding the former to reduce pieces and lead to improved prints, and the latter to be more computationally efficient.

*Optimising Object Decomposition to Reduce Visual Artifacts in 3D Printing* [5] uses decomposition to reduce supports and improve visual quality by using non-planar cuts, by oversegmenting the model and selectively merging these in ways that prioritise the preservation of salient regions, which are likely to require visual quality, and permit supports in occluded regions, where the visual quality can be compromised.

Within the context of optimising printing for multiple parallel printers, these methods all exhibit a same class of limitation. Though support structures are a potentially needless expense of time and material, decomposition and reorientation to avoid this as a primary objective does not necessarily translate into better parallel performance. It is conceivable that smaller pieces which still require supports have ultimately only limited impact on the parallel performance if the overall maximum part printing time is minimised. Equally, if the objective is raw speed, the research focusing on support reduction for improving visual quality may be suboptimal, particularly in the context of parallelisation.

**Assembly with interlocking parts** This domain of research improves upon the use of decomposition by focusing on the assembly process. Here, “interlocking” methods akin to jigsaws are used to achieve intuitive and secure joints between parts. “*Printing 3D Objects with Interlocking Parts*” [33] approaches this by using voxel-based geometric analysis to determine the decomposition for fitting in printers, and then using graph-based analysis of the interfaces between parts to decide how to locally deform the volume in order to create a strong and coherent joint.

“*Reconfigurable Interlocking Furniture*” [32] builds on this in the subdomain of furniture production by focusing on parameters relevant to furniture, such as part re-use, cost, and immobilising stability without adhesives to permit reassembly. It achieves this by importing research on common substructures, parts analysis, and multi-key connective joints, tied together with a backward interlocking model applied to a part graph to ensure the assembly of a locally final part completes a cycle and immobilises the involved structures.

*Surface2Volume* [2] takes a different approach to assembly by segmenting the models into surface conformal parts, rather than using traditional planar slicing, through analysing the feasible parts of the model’s surface regions. “Extractable” parts are chosen based on whether a linear assembly trajectory would cause collisions or otherwise be unfeasible. The interfaces are smoothed for optimal assembling, and their non-planar aspects can permit interlocking assembly without the need for explicit jigsaw parts.

Here, the limitations are the same as the first genre of research: focus on single printer optimisation. Much of this is also exploratory, focusing on optimising the assembly process specifically, rather than on the parallel processes as a whole. *Reconfigurable Interlocking Furniture* does look at improving variables such as cost and material use, but within a specific domain that may not apply more generally.

**Contribution** Summarising the limitations noted above, it can be observed that they all exist within the domain of single printers. While decomposition happens, producing multiple

parts, which could be printed on multiple printers at once, there is no guarantee that these are optimal ways of achieving such. Total build time of the aggregate of parts is often the metric employed, therefore the time being minimised is that of a single print job.

The large-scale model packing algorithms represented can elect to use smaller parts in order to achieve the fitting of the models into the volume, which would be inefficient in a multi-printer domain; the support minimising algorithms can prioritise this reduction of supports when it could result in suboptimal use of parallel printers; the research focusing on improving the metrics generally are also focusing on aggregate build time rather than employing multiple printers.

This research, in contrast, focuses on the application of decomposition within the specific domain of multiple homogeneous printers for the purposes of parallel build time. By taking advantage of the properties of symmetry to achieve a reasonable load-balancing effect with a trivial planar cut, while utilising a relaxation of the symmetry metric in order to adapt the algorithm to non-symmetric situations, and combining it with a hybrid analytical-greedy method for determining optimal subdivision configurations, a pipeline is produced for printing models in parallel efficiently, by reducing the amount of time printers are left idle. These techniques independently have their use in some of the previous work, but combined together in this way and optimised for this domain, it becomes a unique contribution in the domain of larger objects in lab settings.

### 3 Problem overview

#### 3.1 Formulation

The notion of optimising the printing of single models across multiple printers involves solving a different problem to those previously seen in the area. This fundamentally reduces to a problem analogous to that of parallel computation more broadly: of attempting to optimise the throughput of the process by maximising the use of the processing units through an appropriate workload sharing strategy.

For example, if there are four cores to perform a series of 1000 independent trivial arithmetic operations, the classic optimisation is to chunk the data into 4x 250 operations to run in parallel, achieving up to a 4x speed improvement compared to running in serial. Approaching this example suboptimally, one can imagine putting 700 of the calculations on one core, and 100 across the remaining cores. This is suboptimal because the remaining cores will complete their work and be sat idle for significant lengths of time while the other core is still computing, moving the situation closer to that of serial. It is a constraint of “the fleet moves as fast as the slowest ship”.

Our problem mirrors this one: the 3D printers are the computing units, and the geometry to be printed is the work to be done. On paper, two printers each doing a 4 h print is faster to the end user than a single printer doing a 7 h print and another doing a 1 h print.

Taking this analogy further expands the problem into a formulation of the specific problems the research must incorporate:

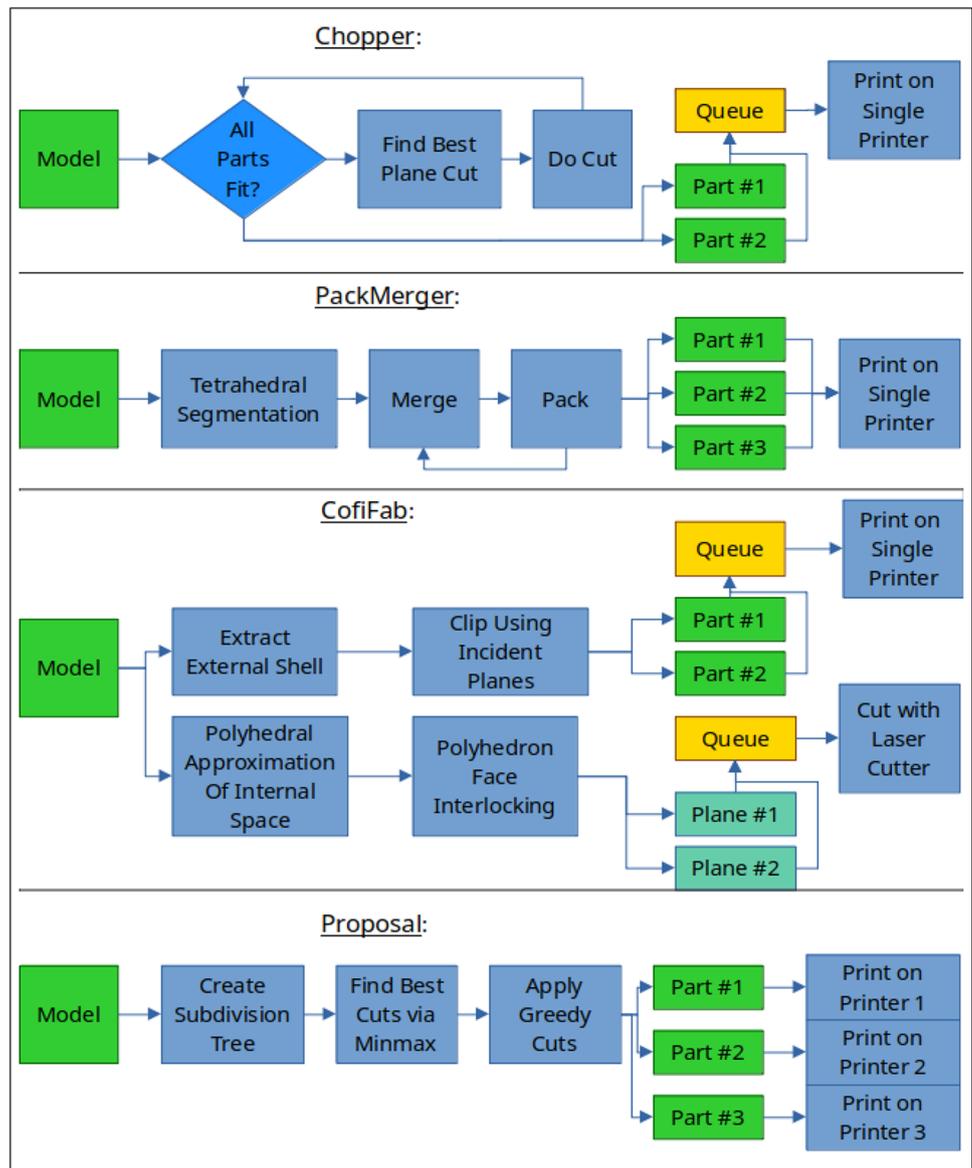
- **Decomposition strategy:** A process must be implemented which takes an input model, and decomposes it in such a way that the aforementioned constraint is observed, in order to produce sufficiently optimal output models for printing.
- **Printing factors:** As the models should be able to be fabricated in the real world, factors relevant to 3D printing need to be considered, such as quality, supports, infill, material, and printer hardware settings and parameters.
- **Assembly:** The resulting models must be able to be assembled into the complete model. This also incurs a cost: assembling could be difficult or time-consuming. Similar to a jigsaw puzzle, more pieces theoretically increase the assembly complexity, as well as the process of assembling taking time, for example in affixing or adhering.
- **Diminishing returns:** There is a concept in the field of parallel computing called “Amdahl’s Law”, which states that as more parallel computing units are added to the computation, the rate of improvement of the computation time will exponentially decay, approaching zero asymptotically [9]. This would certainly apply here, as further subdivisions split an increasingly small set of geometry. This is an important factor to consider — the costs of decomposition and assembly combine to produce a broadly increasing “cost of parallelisation”, which would gradually overtake the gains of parallelisation.

Thus emerges the problem statement: at what point do the improvements gained from parallelisation in 3D printing processes become outweighed by the increasing costs as the degree of parallelisation increases?

There are several ways this could be approached. One such approach could involve designing an all-encompassing algorithm that incorporates these limitations. This algorithm would take a model as input, determine the optimal amount of printers, then decompose the model to make efficient use of those printers. A main drawback of this approach is the increased complexity of multiple objectives, and the modelling of a heuristic to represent, e.g., the costs of assembly (Fig. 2).

In more realistic settings, there would usually be a set number of printers already available, and so another way the problem can then be read is as attempting to find the optimal

**Fig. 2** A highly simplified flowchart detailing the overview of how the current decomposition techniques most relevant to this research employ decomposition, in contrast to the proposed technique



way to decompose an input model such that it makes optimum use of those available printers. The drawback of this approach is that a larger number of printers being available than the optimal will result in deleterious effects, such as escalating assembly times due to increasingly small parts, exacerbating the bottlenecks in the production time.

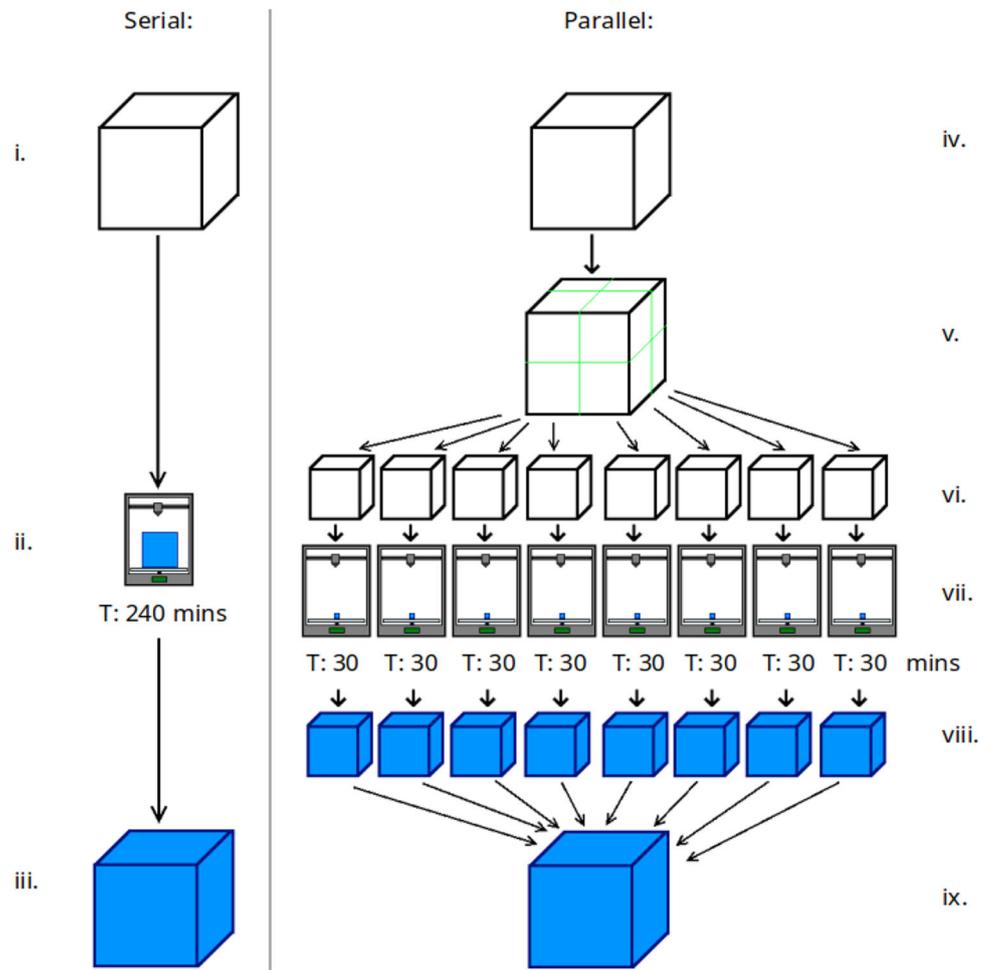
This research elects to implement the latter approach, as it is more concise and enables the quantitative evaluation of how parallelisation affects models across varying numbers of printers, to better address the question of whether parallelisation has reasonable utility in 3D printing. The former approach inherently depends on the development of a method to determine what an optimal number of printers per model is, which would be more rigorous if the quantitative analysis on parallelisation across a range of models had already been done.

### 3.2 Formalisation

As stated above, the general pattern is that the workload should be distributed as evenly as possible. While tailoring pre-existing approaches in the field to this objective were considered that can incorporate symmetry (e.g. *Chopper*), an algorithm is suggested based on symmetry as the core. The reasons for this are many-fold.

Primarily, symmetry represents the best case for the objective, as if an object can be divided symmetrically, this means there are two effectively equal parts in volume and structure, which should translate into a perfectly distributed workload. Secondly, symmetry is a widely seen phenomenon in nature and design, so applying this assumption can be justified in practice. While recursive symmetry (i.e. where splitting symmetrically results in halves that can be further split sym-

**Fig. 3** Overview diagram of the current serial process, and the proposed parallel process side-by-side, visualising the printing of a  $10\text{ cm}^3$  cube. *i*: Serial start. *ii*: Serial printing of cube, taking 240 mins. *iii*: Cube is immediately printed in full scale. *iv*: Parallel start. *v*: Cube is decomposed for 8 printers. *vi*: Resultant  $8 \times 5\text{ cm}^3$  cubes. *vii*: Parallel printing of the 8 cubes on 8 printers, each now taking 30 min (this entire stage takes 30 min). There is still 240 min of workload, just spread out. *viii*:  $8 \times$  resultant printed cubes. *ix*: Assembly into the total cube



metrically) is the ideal case, it may not to be observed outside of structures such as fractals, which impedes this approach, but it is likely to be of significant benefit early on, which is more important given the expectation of diminishing returns.

Where symmetry is often considered as a property an object has or does not have, it can also be interpreted as a quality an object has in varying degrees. For example, a ball is perfectly symmetric, a face is mostly symmetric, but a cloud is usually not very symmetric. By relaxing the definition in this way, it allows for the selection of “better” symmetries, permitting the application of the symmetric subdivision approach even in cases where perfect symmetry is not available, which should particularly help optimisation in the recursive subdivisions. This can be restated as a metric of symmetry: seeing perfect symmetry as an objective with zero error, and deviations towards progressive asymmetry as an increasing quantity of error.

From this approach, the problem can be formalised as a geometric objective: the minimisation of the maximum of volumes of a given sized set of geometry, through the strategy of recursive applications of symmetry derived by minimising

the “symmetric error” across each model and its constituent decomposed parts (Fig. 3).

Suboptimal solutions can theoretically arise in greedy applications of this objective, for example in a situation where better cuts can be made deeper in the search space if a less symmetric cut is made earlier. This is tackled by an analytic approach that considers the search space of possible recursive cuts to a certain depth, ensuring the best possible decomposition of models for parallel printing is determined.

As discussed prior, the formulation involves finding the best allocation of parts across a given number of printers, and suboptimal use of printers will occur once the optimal number for the model is passed. While detecting and analysing this data is a core part of the research, some constraints are built into the objective, in order to avoid incoherent or clearly deleterious allocations of the excess printers (for example, extraneous cuts of tiny pieces from the largest model). It is thus permitted to leave some of the assigned printers idle if further slicing would clearly not improve the throughput.

Note that it is presumed by this research that the deployed 3D printers are of the same type. This is more fitting for a

laboratory or factory setting, where collections of homogeneous printers are more likely to be observed, than in hobbyist settings.

## 4 Method

The implementation of the objective to solve this problem requires the implementation of an algorithm that includes many stages and is best thought of as a pipeline, taking a model and a few parameters as inputs, and outputting the models of the constituent decomposed parts that best represents an optimal solution, as well as assembly instructions for reincorporating the original model.

### 4.1 Symmetry determination

At the base of the algorithm is the symmetry determination logic. This involves the recognition that there are multiple types of symmetry in 3D space — reflection, rotation, axial, circular, and chiral. Determining the symmetries of arbitrary shapes and models is a substantially explored branch of computational mathematics, with various approaches existing in the literature, spanning ranges of complexity and efficiency [10, 23, 46].

For the purposes of this project, a more simple, iterative stochastic method for the determination of reflectional symmetry was selected, influenced by the algorithm proposed in *A Reflective Symmetry Descriptor for 3D Models* [16], as this enables the swift computation of sufficiently good symmetries while avoiding the implementation complexities or slower computational speeds of other proposed methods. The CGAL geometry library used for decomposition operations uses planes, which supports the use of reflectional symmetry and its colloquial “planes of symmetry” in the implementation here.

The process for computing symmetries in a given model begins with the determination of the centroid  $c$  as

$$c = \frac{\sum V}{|V|} \tag{1}$$

A set of points  $P$  is then randomly sampled such that  $\|p - c\| = 1; p \in P$ . These are used to form planes defined  $L = \langle c, p \rangle$  in plane-normal form, and used to sort vertices by the following expression:

$$f(T) = \begin{cases} B_{left} = B_{left} \cup S & S < 0 \\ B_{right} = B_{right} \cup S & S \geq 0 \end{cases} \tag{2}$$

where  $T = v \cdot L; \forall v \in V$  (3)

with  $B$  representing the sets of vertices on either side of the plane. The degree of symmetry associated with a given plane

is then determined using the mean absolute error of nearest-neighbour distances between the vertices in each set such that:

$$S = \bar{E}, \tag{4}$$

$$E = E_{left} + E_{right} \tag{5}$$

$$E_{left} = e(v_{left}, B_{right}); \forall v_{left} \in B_{left} \tag{6}$$

$$E_{right} = e(v_{right}, B_{left}); \forall v_{right} \in B_{right} \tag{7}$$

$$e(v, B) = \min_{\forall v_{other} \in B} |v - v_{other}| \tag{8}$$

$P$  is ordered from smallest error to largest, and the best three are selected to account for the majority of situations involving objects with multiple symmetries. As a stochastic iterative method, this process is repeated with refinement towards local minima until Expression (9) no longer holds true and the result is returned, and by modifying the sampling constraint to Expression (10) and (11) to focus the sampling around the best points:

$$|E_{n-1} - E_n| > \epsilon \tag{9}$$

$$\| |p - c| \| = 1, ; p \in P \tag{10}$$

$$r = r * K \tag{11}$$

$$\text{where } 1 \geq K \geq 0 \tag{12}$$

where  $K$  is a user-specified coefficient of scaling.

### 4.2 Modified binary tree search

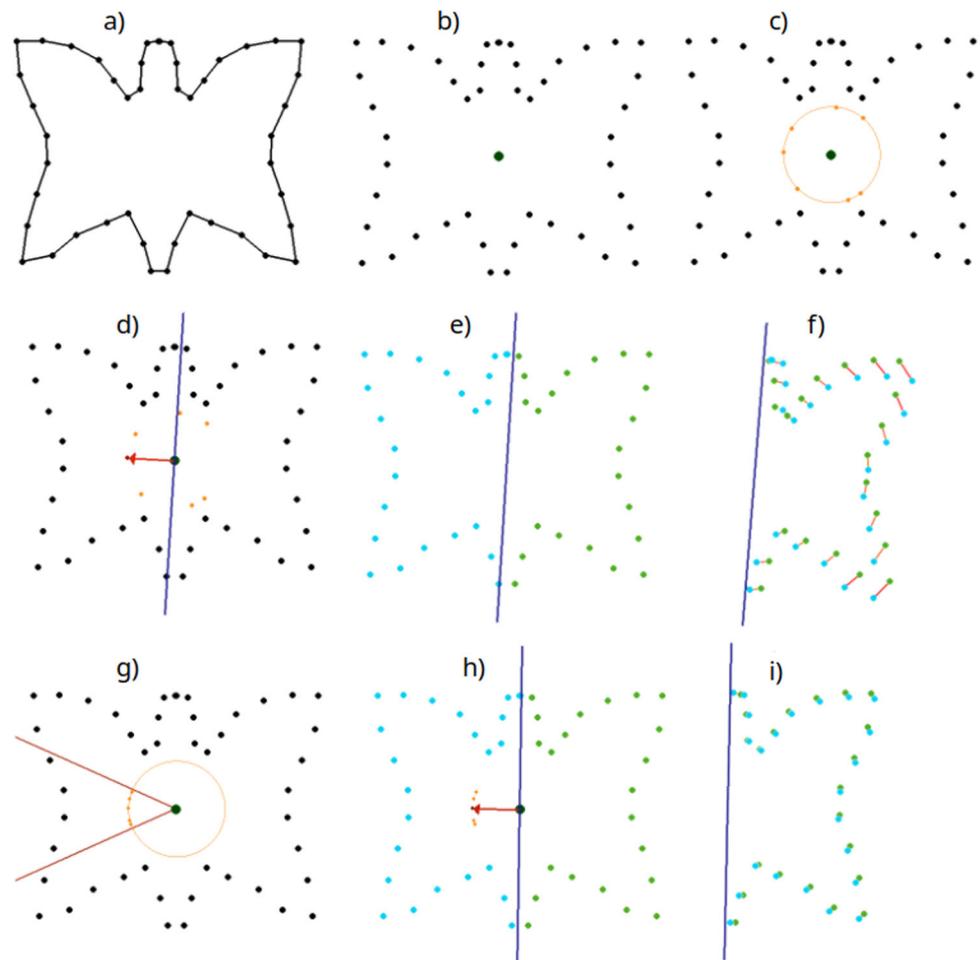
The process of decomposition into halves by planes of symmetry can be viewed as a series of recursive binary operations, seamlessly mapping onto a binary tree. But, the model can have different possible subdivision options, and these subdivisions can divide into further subdivision options, which results in a search space of binary trees interlaced within a broader non-binary tree. The algorithm must create this tree to map the possibilities, and then extract the best possible binary tree from the encompassing non-binary tree by means of a modified binary tree search algorithm.

#### 4.2.1 Construction

The construction of the search tree begins with the user-specified quantity of printers. Without such a constraint, the tree could hypothetically grow indefinitely, limited by computational resources. A depth override may be specified, after which the algorithm ceases the further expansion of the tree (Fig. 4).

The symmetric determination is run for the model (and then each of its constituent parts in turn) to determine the possible planes of symmetry. If some exist, they are applied by means of a planar subdivision operation into the constituent

**Fig. 4** A simplified 2D representation of the algorithm discovering symmetries on a butterfly-shaped model. **a)** Input model. **b)** Vertex-only view with centroid. **c)** Sampled points on a unit circle around centroid. **d)** A sample (the best one for example) + centroid used to construct a plane. **e)** Plane used to organise vertices into two buckets. **f)** Neighbour sampling between the two to determine aggregate error (redder lines = larger distance = more error). **g)** Second iteration: sampling restricted around the best sample point. **h + i)** Symmetry convergence depicted with reduced errors



models. Due to the computationally independent nature of the iteration logic, this operation is trivially parallelisable in itself, and achieves considerable performance boosts through the use of the *OpenMP* library.

Subdivision is also restrained by a set of constraints. For example, the requirement that the geometry be closed and free of self-intersections is imposed in order to make the algorithm function without requiring the more complex geometric operations that these require. A minimum size is also imposed to prevent the generation of pieces that are too small to print. Subdivisions that would violate any of these are rejected without further consideration, although quick attempts are made using CGAL's repair library to fix self-intersections or non-closed meshes beforehand.

The combined search tree embeds all the possible binary trees within it using an alternating series of node types, "subdivision option" and "subdivision options". The subdivision option nodes represent a given plane of symmetry used to decompose the parent, and two subdivision option nodes as children. The subdivision options nodes represent a given mesh, with an array of subdivision option children. By the nature of the binary subdivision, each option must have two

children, but the plural options nodes can have anywhere from zero (representing no further subdivisions) onwards. An example of this is shown in Fig. 5.

#### 4.2.2 Extraction

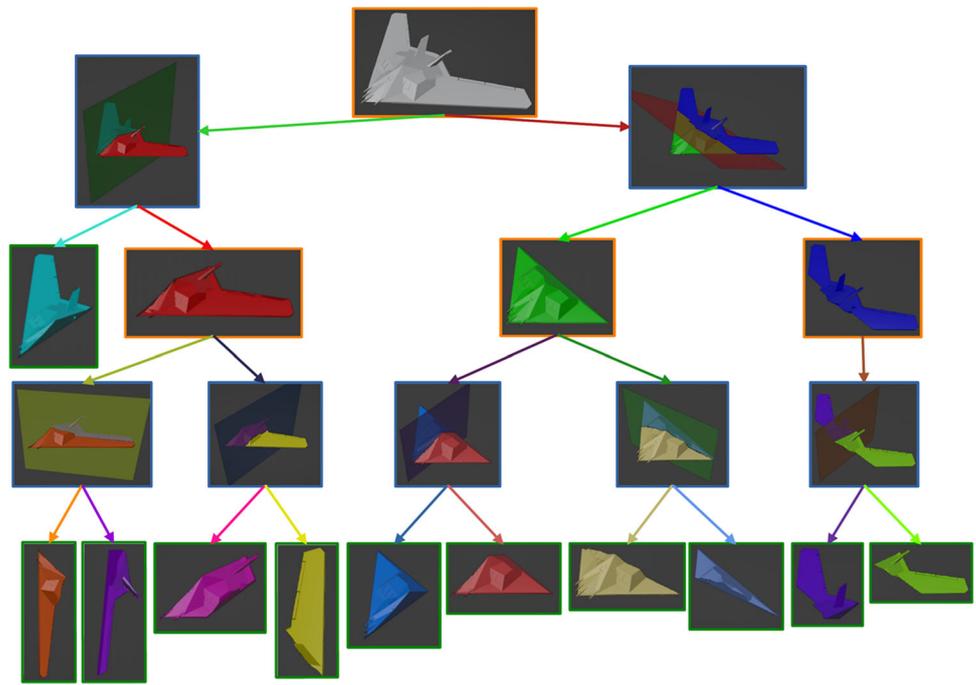
Once the tree is constructed, the algorithm walks through it extracting the possible combinations in order to find the optimum. This algorithm involves a right-handed traversal that replaces parent nodes with the two subdivision nodes in the printer array as they are visited, but is subject to some modifications in order to make it compatible with the modified nature of the tree compared to binary trees. This results in the algorithm outlined in Algorithm 1.

A visual demonstration of this algorithm in action can be seen in Fig. 6.

#### 4.3 Refinement

Initially, prior to the development of the tree logic, the entire algorithm was based on a greedy algorithm, whereby the best

**Fig. 5** A colour-coded example of a subdivision search tree produced from an input model. Nodes highlighted in blue are subdivision options, in orange are meshes with subdivision options, and in green are leaf meshes. Note how some paths have multiple options, some have just a single option, and the branch on the far left terminates early



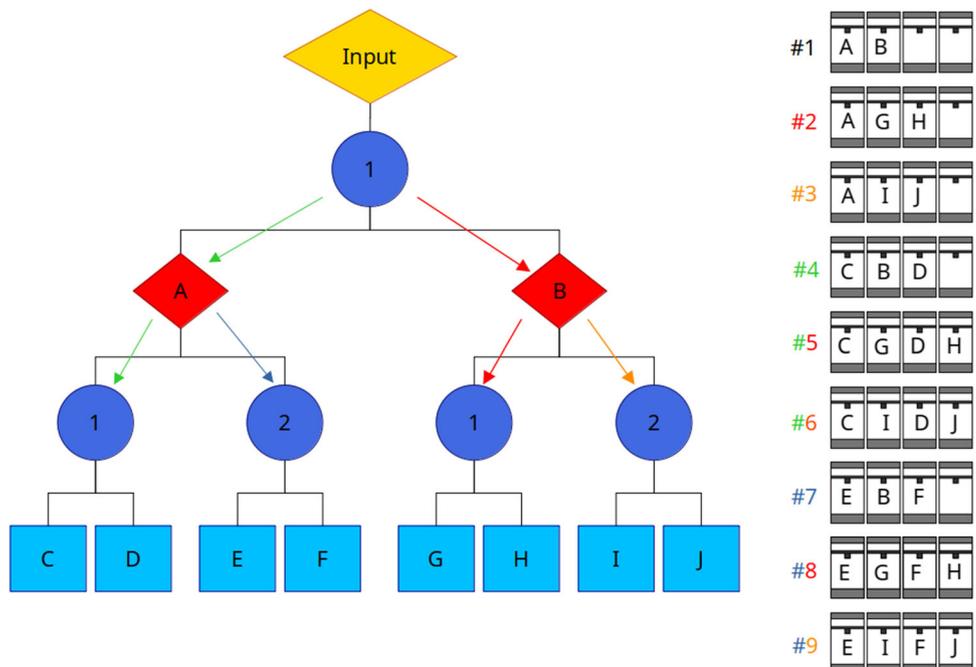
symmetry was sampled and used ad infinitum, but can theoretically lead to situations where less optimal subdivisions early that enable more optimal subdivisions at deeper parts of the search tree are missed, leading to the use of the tree logic to identify the better solutions in these cases.

With the development of the tree-based algorithm, it was quickly discovered that there was a significant combinatorial explosion issue in practice at even relatively shallow extents, due to the repeated duplicating of vertices causing a

rapid depletion of memory and multiple days of processing time. Further thought led to the recognition that each iteration of decomposition results in smaller geometry, and therefore less to gain from the smarter analytical logic, especially as it became exponentially expensive.

Therefore, it was decided to relax the analytical constraint at deeper depths of subdivision. After 3 iterations, the algorithm continues with the greedy approach, filling out the remainder of the printer arrays with much faster subdivi-

**Fig. 6** An example of a subdivision search tree (with the diamonds representing meshes with subdivision options, the blue circles representing the subdivision options, and the blue labelled boxes representing terminal meshes) with the extracted printer arrays on the right. This demonstrates the algorithm expanding B's options via the red then orange paths, then A's options via the green and blue paths, using the pivot in the A expansions to repeat the red and orange paths



**Algorithm 1** Extraction of configurations from subdivision options tree.

```

printerArrays ← ∅
printerArray ← [0..n] = (null)
printerArray[0] ← rootNode
pivotNodes ← ∅
ExploreOption(printerArray[0], pivotNodes, printerArray)

function EXPLOREOPTION(option, pivots, printerArray)
  if IsPrinterArrayFull(localPrinterArray) ≠ true then
    localPrinterArray ← printerArray (deep copy)
    ExpandIntoPrinterState(option, localPrinterArray)
    add localPrinterArray to printerArrays

    ExplorePivots(pivots, localPrinterArray)
    ExploreNode(option.rightNode, pivots, localPrinterArray)
    push node onto pivots
    ExploreNode(option.leftNode, pivots, localPrinterArray)
  end if
end function

function EXPLORENODE(node, pivots, printerArray)
  for all option ∈ options(node) do
    ExploreOption(option, pivots, printerArray)
  end for
end function

function EXPLOREPIVOTS(pivots, printerArray)
  localPivots ← pivots
  while localPivots ≠ ∅ do
    pop localPivots into pivot
    ExploreNode(pivot.rightNode, localPivots, printerArray)
  end while
end function

```

sions. This completes the “hybrid approach”, permitting the analytical logic early on where it is most useful while still permitting for the possible benefit of highly refined subdivisions.

#### 4.4 Optimisation discovery

From the above processes, an array of printer states holding mesh references is filled. The algorithm’s final major step is to find the optimal array. This relies on the minimax strategy; or, minimising the maximum cost. This is subject to the following expression:

$$i_{best} = \arg \min_{x \in S} \quad (13)$$

$$\text{where } S_n = \max_{\text{volume}(p) \in P_n} \quad (14)$$

where  $P$  represents the array of printer arrays, and  $i_{best}$  represents the index of the optimal array.

#### 4.5 Export

The final step of the programming logic is to export the models so that they can be printed. This is trivially performed by

exporting the models from the selected optimal printer state array into an OFF file using CGAL’s inbuilt API, and then using the meshconv UNIX tool to convert them to STL for wider compatibility.

Assembly instructions are also required to enable an intuitive process for combining the constituent models into the initial model. Here, a similar process to *Fabrication-aware Design with Intersecting Planar Pieces* [30] is followed, whereby the process of assembly is conceptualised as an inversion of the process of decomposition.

Thus, the idea is to see each subdivision from the root to the leaves, as assembly operations from the leaves to the root. By starting at the leaves and assigning numeric IDs to them incrementally, followed by their parent nodes and back to the root, and echoing them to a text file as combinatorial operations, the full assembly sequence can be rendered for the user, taking the intermediate steps and dependence hierarchy into account. STLs of the intermediate models can also be exported to give the user a view of how to assemble parts that may be unintuitive.

## 5 Results

In order to test the algorithm, it must be run across a series of representative models in a series of increasing rates of decomposition, giving a well-rounded perspective of its performance. The physical act of printing and assembling should be carried out as much as possible in order to gauge its performance in the real world, rather than just in simulation. The full cost of each model is a simple combination of the algorithmic computation time, the longest time to print of the set of decomposed parts, and the processing/assembly time. The same printer class and settings must be used across all prints to minimise any confounding variables that variations in these can introduce, to ensure that only the algorithmic performance is being tested.

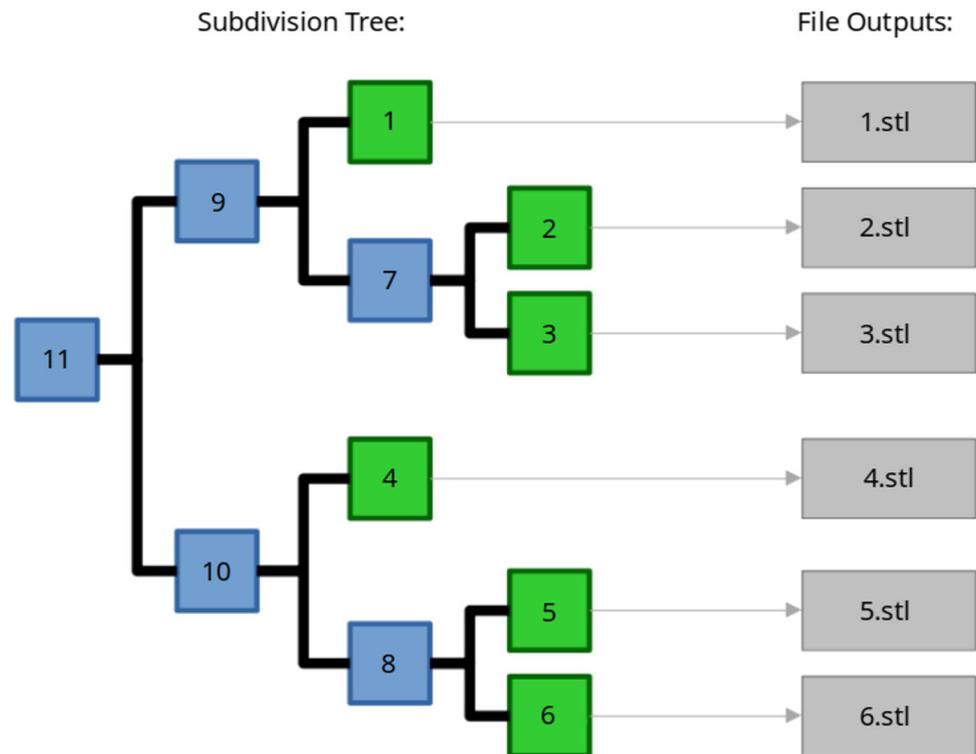
### 5.1 Sample selection

In an ideal situation, a large zoo of geometry is tested, with each sample being subjected to all possible subdivision counts a multitude of times in order to examine the performance of the algorithm as reliably as possible, but these would require a very large quantity of experimentation to achieve. So, due to time and resource constraints, 6 subdivision degrees were selected that represent a broad yet practical range of printer quantities, and a range of models.

The subdivisions selected:

- 1: The serial control test.
- 2: The trivial subdivision.
- 3: An odd trivial subdivision.

**Fig. 7** Example of the ID assignation ordering, and how this translates into the output files. Blue: Standard mesh nodes, Green: Leaves



- **4:** A deeper subdivision; power-of-two.
- **8:** A comprehensive subdivision; power-of-two.
- **11:** A reasonably sized prime number.
- **15:** A large odd-numbered quantity of printers that may be found in heavy duty lab settings.

A series of 25 models were drawn randomly from the *Thingi10K* dataset to use as the test set. This sample was then reduced further to 20 by eliminating incompatible or irrelevant models, subject to the following criteria:

- Model must be continuous.
- Model must fit in the printing volume (to permit serial comparison).
- Model cannot be too small ( $< 2\text{cm}^3$ ).
- Model must be closed, non-degenerate, and well-formed (in order to be used by CGAL).

A few additional models were added to increase the diversity of the sample set, such as cubes of varying sizes to test the scalability in a controlled environment, as well as popular 3D printed models to represent concrete models in the dataset, such as the Stanford Bunny to test asymmetry and Benchy3D to test topological complexity (Fig. 7).

## 5.2 Equipment setup

All models were printed on *Ultimaker 2+* printers, with the type of printer held constant to control for deviations

in performance between different printer types. These were available in significant quantity in a laboratory setting.

The printers featured [22]:

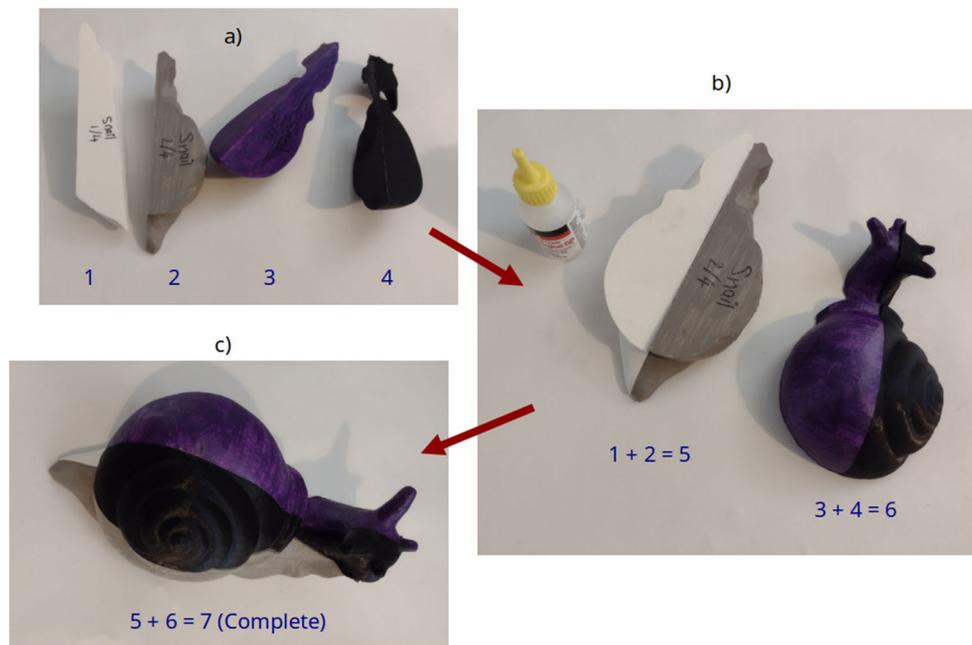
- 2.85 mm PLA filament, fed by bowden tube.
- A 0.4 mm nozzle, heated to 200°C.
- Glass build plate, heated to 50°C.
- Build volume of size 223 mm x 223 mm x 205 mm.
- Gantry-based 2D extruder motion, with 12.5 $\mu\text{m}$  positioning precision, with the bed on a vertical motion Z axis at 5.0 $\mu\text{m}$  positioning precision.
- Print speed of 30–300 mm/s, and travel speed of 30–350 mm/s.

## 5.3 Analysis

### 5.3.1 Process

The models in the dataset were decomposed according to the selected subdivision strategy on a high-powered quad-core desktop machine in bulk by a bash shell script, with a high-resolution timer used within the program to determine the computation times of each slicing task. The same parallel decomposition settings were used for all models: a search depth of 3, 20 cm maximum dimension, 100 symmetry samples with 3 stochastic iterations and an error tolerance of 1.0, a search radius of 0.1r, and a fitness filter of 5. The output models were then manually arranged and sliced in *Ultimaker*

**Fig. 8** Photographs showing an example in the real world. **a)** Assembled, processed, and labelled parts; **b)** Partial assembly; **c)** Completed model



*Cura* with settings optimised for speed (0.15 mm, 10% infill, with raft and support struts), with fabrication duration timings taken from the computed result.

One major issue accounted for was the vast quantity of printing required, and the inconsistent availability of lab access. Methods were improvised in order to maximise physical samples while maintaining the integrity of the data. A brief anecdotal test early on suggested that the actual printing times did not vary substantially from the computed times (within minutes), and the physical prints are mostly required for feasibility and assembly time calculation, so the computed times were used for the printing times. Further, manually recording each individual print while up to 15 are being printed at the same time requires a lot of human supervision time. So, on top of this, multiple models were often assigned to a single build plate to further increase throughput by chunking relevant prints (the timings were taken on an individual-part basis, nonetheless).

The models were then realised on Ultimaker 2+ printers in a fabrication laboratory using PLA filament. Models were identified by writing their ID on them in permanent marker, and organised into boxes on a per-model basis. These printed models were then cleaned up (involving removal of struts and supports, and sanding of interfacing planes) and assembled by hand with cyanoacrylate-based superglue, with the computed assembly instructions consulted where necessary. A simple stopwatch was used to determine the time of the complete operation per fully assembled model. A terse example of the assembly process can be depicted in Fig. 8.

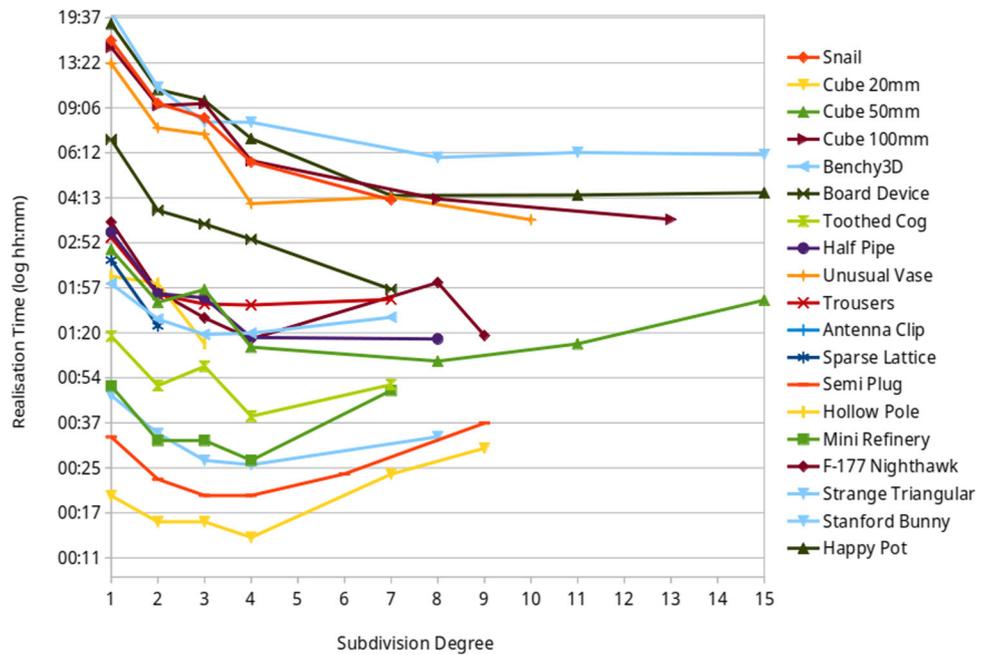
### 5.3.2 Data

The results were determined for each model at each subdivision degree by aggregating the temporal data associated with the formula  $t_{total} = C_t + \max_{P_t \in P_n} + A_t$ , where  $C_t$  is the computing time for decomposition,  $A_t$  is the time taken to assemble the model in full (until stable), and  $P_n$  is the set of times to print the set of decomposed models. Sometimes the algorithm determined that less than the specified quantity of printers was actually optimal, resulting in data in between the specified quantity of printers. This data is depicted logarithmically in Fig. 9 to best express the trends between the wide variance in printing times.

As can be seen in Fig. 9, there is indeed a rapid improvement in the time to realise the models with increasing degrees of subdivision early on, but that there are notable diminishing returns. As expected, this effect is much stronger with smaller models, where the diminishing returns outpace the improvements gained from parallelisation after a mere 4 subdivisions, whereas for larger models this point was beyond even the highest degrees tested (although the algorithm rejected most models at 15 subdivisions due to violating other constraints) (Fig. 10).

This is more easily expressed in Fig. 11, with larger degrees of subdivision experiencing more improvement at larger volumes, albeit with a general convergence from 8 degrees or more, showing the effects of diminishing returns. While *Cube 100mm* and *Unusual Vase* appear to be showing improvement at this point, *Stanford Bunny* and *Happy*

**Fig. 9** Total realisation times of printed and assembled models in a log-time graph



*Pot* converge at 8, showing the negative effects of asymmetry and sparsity respectively on even large models with this approach.

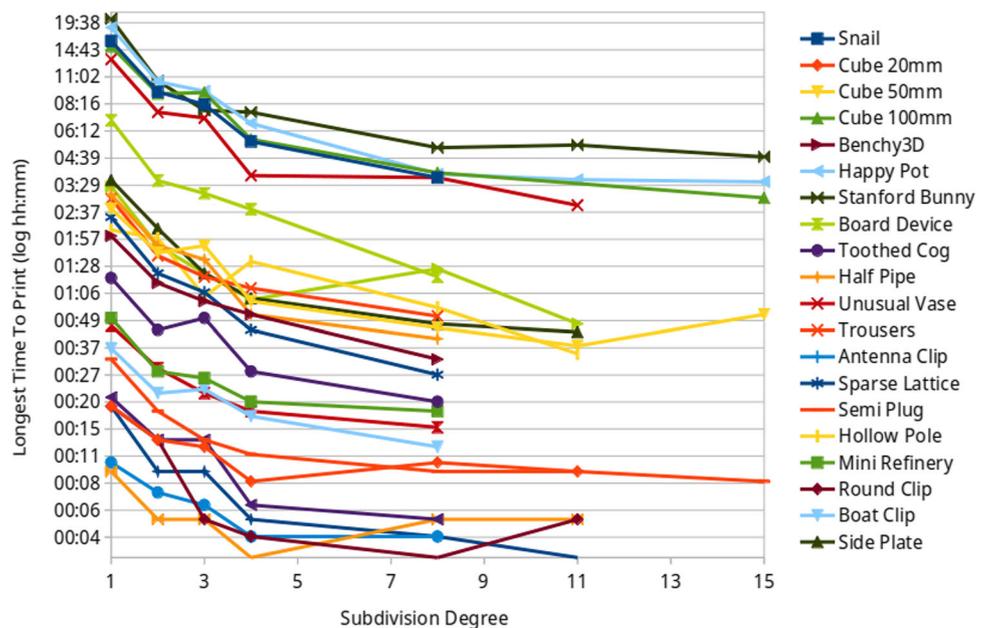
Another relationship that can be inferred is that model complexity is also related to reduced performance, with intricate models featuring increased sparsity, more facets, and thin parts (such as *Benchy3D*) having earlier inflection points with less temporal improvements even before this, compared to simpler models. Models that strongly expressed these characteristics often failed outright to slice, print, or assemble, for example *Antenna Clip* routinely failed to print on the 3D

printers, and *Hollow Pole*'s thin shells proved prohibitive to join together with cyanoacrylate.

Of note is how computation time was frequently on the order of seconds, and into the minutes for more complex models, although very complex models (millions of vertices) did not show in the data set. Compared to the scale of hours printing often operates at, the algorithm appears efficient enough at the specified parameters for the computation time to be a minuscule part of the overhead.

One deleterious effect observed is that an occasional model (or subdivision) was split in such a way that sin-

**Fig. 10** The longest time to print results of each sample model and its degrees of subdivision, in graphical form



gle subdivisions ended up as multiple disconnected parts. These were grouped together as a single part, and assembled together, although this potentially led to increased assembly time as multiple effective prints were performed for what is labelled a single print.

### 5.3.3 Qualitative comments

Although this research is primarily concerned with temporal performance, some qualitative feedback is appropriate. Notably, the algorithm struggled to handle certain types of geometry, such as those with long, thin, narrow parts, such as can be seen in antennae and trees, as well as models with large amounts of concavities. It appeared that the algorithm worked reliably with more convex models.

By the nature of planar decomposition and the use of rafts, the visual quality of the assembled prints was impeded in the form of misalignments between the connecting planes. With increased degree of decomposition, these artefacts compounded, which become more notable on smaller prints compared to larger prints, which maintain a smoother appearance. The nature of assembling by hand, as well as the slight inherent warping of PLA, also introduced small errors, which further compounded at higher degrees of decomposition and also proved particularly obvious in smaller models, but also presented difficulties in adhesion due to the accumulated misalignments.

Nonetheless, the relatively smaller impact of errors on larger models such that their structures are reasonably main-

tained, with larger cross-sectional surfaces increasing the strength of the superglued joints, is a significant qualitative advantage that works well with the quantitative data in that specific domain.

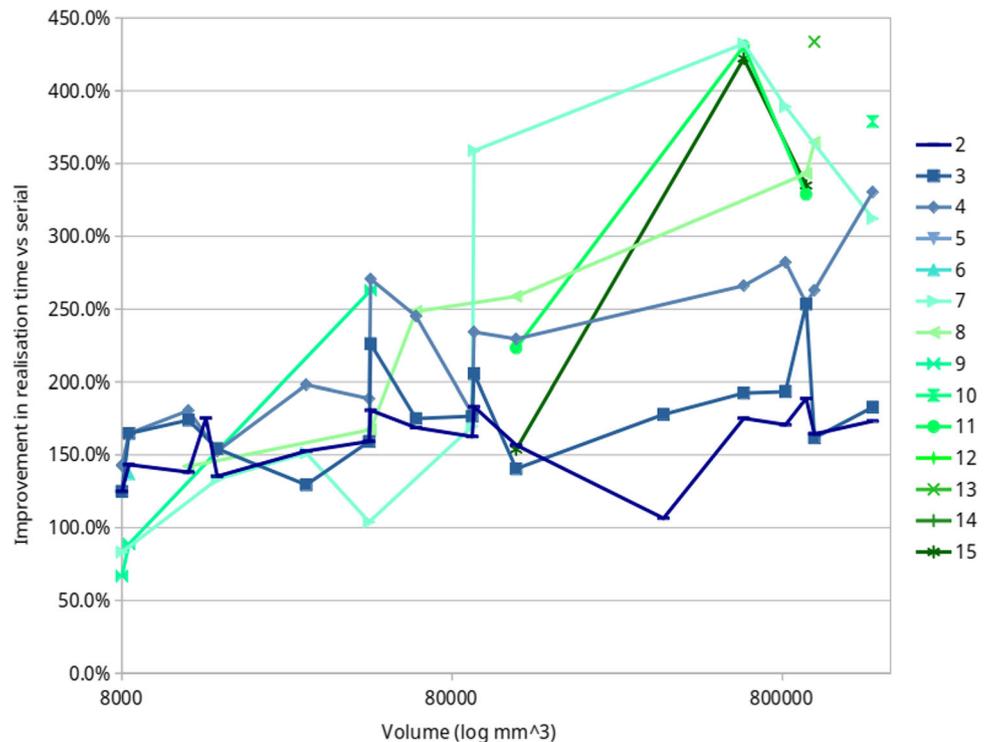
The process itself was also somewhat tedious. Organisation of the pieces was a concern with models involving higher degrees of decomposition, as the quantity of pieces increased clutter. Annotating the pieces by hand using permanent marker was also suboptimal and chosen only in the interest of time, with smudging and the limited surface area of smaller models becoming an intermittent issue. Finding and holding the correct orientation was not only costly in time, but also required a degree of cognitive effort, even with assembly instructions and models of the intermediate stages.

However, one apparent benefit that emerged was that printer failures resulted in only part of a full model failing, rather than the whole. This meant that, rather than having to restart the whole print and having lost potentially large quantities of filament, only a smaller amount of possible filament was lost and only the failed parts had to be re-printed.

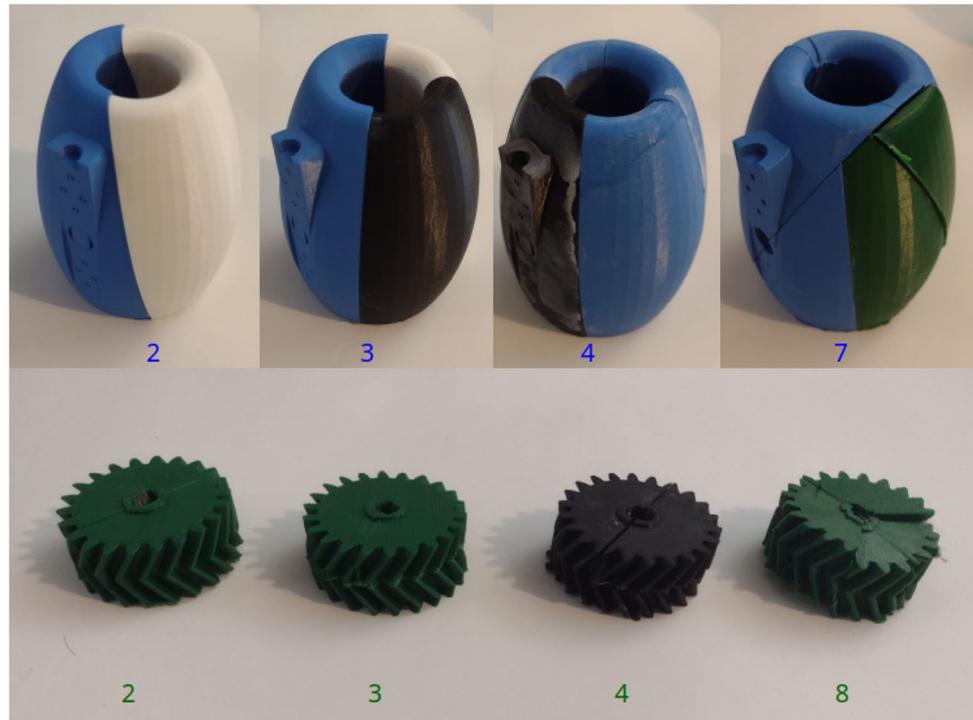
## 6 Conclusions

This paper explored the use of recursive reflective symmetry as a method of optimising the parallelisation of printing of models across a variable quantity of 3D printers. The data suggests that the efficiency and reliability of this method vary considerably between different types of models, with simpler

**Fig. 11** Results plotted to show how volume affects the quantity of improvement from the algorithm at varying subdivisions presented here. The gradient plot helps to observe how, generally, higher subdivision is associated with more reliable improvements at higher volumes, but reduced performance at lower volumes



**Fig. 12** Demonstration of the degradation of visual quality with higher subdivisions, and how this appeared more significant for smaller models. Note how Unusual Vase by 7 retains its structure more smoothly than Toothed Cog by 8



and larger models resulting in reasonable improvements in fabrication duration up to 11 printers, while smaller and more complex models result in poor performance beyond 4 printers. Thus, in domains involving simpler and larger models, this approach could offer suitable improvements to the net fabrication time (Fig. 12).

However, for smaller and more complex structures, the temporal gains are far more minuscule, and algorithmic determination of how to split it in four is unlikely to offer many advantages compared to manually splitting it using the tools available in modern 3D printing software. One major consideration that must be taken is that models had to be able to be printed in serial in order to afford such a comparison, which required that models cannot be larger than the printer build volume.

Based on our findings, it can be extrapolated that increasing model sizes beyond these limits would result in further benefits of parallelisation, with the diminishing returns overtaking the gains at larger quantities of parallel printers, subject to an exponential relationship.

As a whole, the process at higher degrees of subdivision proved highly cumbersome to organise. Managing multiple 3D printers can require significantly more human effort, and the assembly process for models split across more than five printers became tedious and exhausting. Thin and sparse models were highly fragile, prone to warping, and tended to be extremely difficult to assemble.

Regarding the algorithm itself, its computational performance seems reasonable up to medium complexity models,

although certain limitations in CGAL's supported model types led to some restrictions in using models. The configurability of the implementation affords a degree of user-specified flexibility for optimising in certain domains, but was not widely explored here due to the requirement of keeping parameters equal across the sample set to enable appropriate comparison of performance.

The process taking place as sequences of higher-level geometric operations on the models themselves enables the algorithm to be theoretically agnostic with regard to printing/slicing software, 3D printer type, configuration, or manufacturer, although the volumetric heuristic is strictly oriented towards additive manufacturing vs subtractive.

Some differences in performance are likely to show with different classes of 3D printer or configuration, e.g. robotic arm or infill settings, but the overarching trend should persist due to the significant impact of volume on printing time.

"Sufficient symmetry" appears to have been achieved at relatively low cost, but improved symmetry could potentially afford better results. Performance did become an issue with very complex models, where the requirement of keeping both the children and parent meshes in memory at the same time caused an explosion in RAM requirements, and CGAL's geometry clipping algorithm became prohibitively slow operating here. Given the demonstrated optimal use of the algorithm in larger models, it is arguable that the benefits of our analytical stage of the algorithm do not outweigh the costs in the domain of large, complex models with millions of vertices.

Ultimately, given the costs of 3D printers at this point, as well as the amount of effort required to assemble highly parallel prints, and the increasingly small gains in performance at higher degrees of decomposition — especially smaller models — it would appear that this specific application of symmetry for parallelisation only truly offers potential benefits to labs seeking rapid manufacturing of large, crude structures. But, this remains a fertile area for further research, with potential improvements that can be made.

## 6.1 Limitations

Various limitations exist within the research itself. Comparing against existing software that decomposes extremely large models for packing could not be performed, so in the domain of models too large for the printers it is uncertain what benefits this algorithm could afford. The broad sampling strategy forced us to constrain the parameters of both the algorithm and printing to optimise for speed, so more fine-tuned settings may afford improvements in the realisation time, but a person employing, e.g., a finer resolution is unlikely to appreciate the decreases in visual quality caused by this subdivision. This was also only tested on one specific type of 3D printer, with PLA only, so it is uncertain how the performance qualities are altered across other brands and filament types.

A limitation in the nature of our experimentation methods is that all of the models had to be printed before assembly took place. It is conceivable that in lab settings, parts could begin to be assembled while other parts were still being printed. This could considerably alter the dynamic between part dependencies, where aspects of assembly time are distributed into the printing time where other models are still being printed, essentially making it free and improving the net printing time.

One area that was not explored by this project was the impact of orientation on printing speed. Due to the difference in speeds of printing in the XY plane and Z axis, certain orientations could potentially offer significantly different timings. This is especially so when certain orientations producing support requirements are considered. While build plate optimisation was crudely done by hand when preparing to print parts, algorithmic awareness of this factor could lead to different results.

## 6.2 Improvements and future work

Given the significant impact of assembly time on performance, this would seem the most fruitful area to target for improvement. The use of ABS would have afforded notable improvements in assembly time, as the use of acetone for adhesion would have avoided the necessity of sanding and

supergluing with PLA, which caused issues in assembly. This may have also improved the assembly for thinner and more sparse models as the acetone melding offers a stronger joint at a quicker speed than cyanoacrylate in PLA.

The use of *Slic3r* instead of *Cura* may have significantly reduced the assembly time, as it currently employs support struts that are easier to remove through thinner connections. Of course, the use of a multi-material nozzle with PVA for support structures is frequently supported and the ability to rapidly and trivially dissolve these parts offers an even simpler solution, but was not available for this project. Mechanical or automated assembly processes would also significantly improve the performance and reliability of this step by reducing the inefficient human element, but it is difficult to conceptualise how this would work at this current point in time.

A few minor changes could be made to the algorithm based on our findings. For example, while volume was used as a heuristic for print time, sometimes this produced unnecessarily thin, long pieces, reflecting a notable influence of surface area on printing time. Thus, a heuristic that combines a relationship of volume and area could produce better results.

Regarding the issue with highly complex models, removing the analytical step entirely and merely using a greedy approach could be both simpler and more efficient in this domain, requiring substantially less RAM due to the removal of the need to keep parent meshes resident in memory. Similarly, an alternative way of slicing geometry to CGAL's native slicer that is specialised for models of this complexity could be a significant improvement here.

Considering the limitation involving orientation, this could potentially improve performance in multiple aspects simultaneously, by optimising the actual printing time, but also by possibly reducing support structures and therefore improving the assembly time. Orientation objectives could also be tooled to prioritise structural integrity, allowing the thinner and sparser models to be able to be printed.

This paper also focused the effort on decomposition optimisation as purely as possible, and so lacked the application of pre-existing solutions to some of the problems that emerged. Integrating the technology involved in *Clever Supports* [34] to achieve support avoidance could yield improvements in assembly time by reducing the pre-processing. Incorporating interlocking joints, seen in projects like *Printing 3D Objects with Interlocking Parts* [33] could improve assembly time through making the jigsaw-like aspects more intuitive, and potentially stable without the use of an adhesive.

**Author Contributions** Concept: H.H. Design: H.H. and J.M. Implementation: H.H. Experimentation: H.H. Supervision: J.M. and U.M. and M.K. Data analysis and interpretation: H.H. Writing: H.H. Reviewing and editing: M.K. Final approval of the article: M.K. and U.M. Publishing: M.K.

**Funding** This research was part of a PhD project at the University of Hull and consequently was indirectly funded by the institution.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Alderighi T, Malomo L, Bickel B, Cignoni P, Pietroni N (2021) Volume decomposition for two-piece rigid casting. *ACM Trans Graph (TOG)* 40:1–14
- Araújo, C, Cabiddu, D, Attene, M, Livesu, M, Vining, N, Sheffer, A (2019) Surface2volume: Surface segmentation conforming assemblable volumetric partition. *ACM Trans Graph*, vol 38(4)
- Bacciaglia, A, Ceruti, A, Liverani, A (2020) Advanced smoothing for voxel-based topologically optimized 3d models. In: 2020 IEEE 10th international conference nanomaterials: applications and properties (NAP), pp 02SAMA19–1–02SAMA19–5. IEEE
- Chen X, Zhang H, Lin J, Hu R, Lu L, Huang Q, Benes B, Cohen-Or D, Chen B (2015) Dapper: decompose-and-pack for 3D printing. *ACM Trans Graph* 34(6):1–12
- Filoscia, I, Alderighi, T, Giorgi, D, Malomo, L, Callieri, M, Cignoni, P (2020) Optimizing object decomposition to reduce visual artifacts in 3d printing. *Computer Graphics Forum*, vol 39
- Fok, K-Y, Cheng, C-T, Ganganath, N, Iu, HH-C, Tse, CK (2018) Accelerating 3d printing process using an extended ant colony optimization algorithm. In: 2018 IEEE international symposium on circuits and systems (ISCAS), pp 1–5. IEEE
- Gao Y, Wu L, Yan D-M, Nan L (2019) Near support-free multi-directional 3d printing via global-optimal decomposition. *Graph Model* 104:101034
- Garashchenko Y, Rucki M (2021) Part decomposition efficiency expectation evaluation in additive manufacturing process planning. *Int J Prod Res* 59:6745–6757
- Hill MD, Marty MR (2008) Amdahl's law in the multicore era. *Computer* 41(7):33–38
- Hruda L, Kolingerová I, Lávička M (2020) Plane space representation in context of mode-based symmetry plane detection. In: Krzhizhanovskaya VV, Závodszy G, Lees MH, Dongarra JJ, Sloat PMA, Brissos S, Teixeira J (eds) *Computational science - ICCS 2020*. Cham. Springer International Publishing, pp 509–523
- Hu, Q, Feng, D, Zhang, H, Yao, Y, Aburaia, M, Lammer, H (2020) Oriented to multi-branched structure unsupported 3d printing method research. *Materials*, vol 13(9)
- Jadoon AK, Wu C, Liu Y-J, He Y, Wang CCL (2018) Interactive partitioning of 3d models into printable parts. *EEE Comput Graph Appl* 38(4):38–53
- Jiang J, Stringer J, Xu X, Zhong RY (2018) Investigation of printable threshold overhang angle in extrusion-based additive manufacturing for reducing support waste. *Int J Comput Integr Manuf* 31(10):961–969
- Jiang, J, Xu, X, Stringer, J (2018) Support structures for additive manufacturing: a review. *Journal of Manufacturing and Materials Processing*, vol 2(4)
- Karasik E, Fattal R, Werman M (2019) Object partitioning for support-free 3d-printing. *Computer Graphics Forum* 38(2): 305–316
- Kazhdan M, Chazelle B, Dobkin D, Funkhouser T, Rusinkiewicz S (2003) A reflective symmetry descriptor for 3d models. *Algorithmica* 38(1):201–225
- Li, C, Wu, B, Zhang, Z, Zhang, Y (2022) A novel process planning method of 3 + 2-axis additive manufacturing for aero-engine blade based on machine learning. *Journal of Intelligent Manufacturing*
- Li, Z, Xiong, G, Zhang, X, Shen, Z, Luo, C, Shang, X, Dong, X, Bian, G-B, Wang, X, Wang, F-Y (2019) A gpu based parallel genetic algorithm for the orientation optimization problem in 3d printing. In: 2019 International conference on robotics and automation (ICRA), pp 2786–2792. IEEE
- Luo L, Baran I, Rusinkiewicz S, Matusik W (2012) Chopper: partitioning models into 3D-printable parts. *ACM Trans Graph* 31(6):1–9
- Mahdavi-Amiri A, Yu F, Zhao H, Schulz A, Zhang H (2020) Vdac: volume decompose-and-carve for subtractive manufacturing. *ACM Trans Graph* 39(203):1–203
- MakerBot (2022) Printing at super speed: how to 3D print faster. Accessed: 2022-12-18
- MakerBot (2022) The ultimaker 2+ specifications. Accessed: 2022-12-18
- Mitra NJ, Pauly M, Wand M, Ceylan D (2013) Symmetry in 3d geometry: Extraction and applications. *Computer Graphics Forum* 32(6):1–23
- Muntoni A, Livesu M, Scateni R, Sheffer A, Panozzo D (2018) Axis-aligned height-field block decomposition of 3d shapes. *ACM Trans Graph* 37:1–15
- NASA (2014) Printing challenges for first 3D printer aboard international space station. Accessed: 2017-07-05
- Oh Y, Ko H, Sprock T, Bernstein WZ, Kwon S (2021) Part decomposition and evaluation based on standard design guidelines for additive manufacturability and assemblability. *Additive Manufacturing* 37:101702
- Generation of Continuous Toolpaths for Additive Manufacturing Using Implicit Slicing (2021) volume Volume 2: 41st computers and information in engineering conference (CIE) of international design engineering technical conferences and computers and information in engineering conference
- Quoraishi, N, (2021) What are the limitations of 3d printing technology?. Accessed: 2022-12-18
- Saxena, A (2016) A comprehensive study on 3d printing technology. *3D Printing Technology*, 6:63–69
- Schwartzburg Y, Pauly M (2013) Fabrication-aware design with intersecting planar pieces. *Computer Graphics Forum* 32(2): 317–326
- Song P, Deng B, Wang Z, Dong Z, Li W, Fu C-W, Liu L (2016) CofiFab: coarse-to-fine fabrication of large 3D objects. *ACM Trans Graph* 35(4):1–11
- Song, P, Fu, C-W, Jin, Y, Xu, H, Liu, L, Heng, P-A, Cohen-Or, D, (2017) Reconfigurable interlocking furniture. *ACM Trans Graph*, 36(6):174:1–174:14

33. Song, P, Fu, Z, Liu, L, Fu, C-W (2015) Printing 3d objects with interlocking parts. *Computer Aided Geometric Design*, **350-36**:137–148. *Geometric Modeling and Processing 2015*
34. Vanek J, Galicia JAG, Benes B (2014) Clever support: efficient support structure generation for digital fabrication. *Computer Graphics Forum* 33(5):117–125
35. Vanek J, Galicia JAG, Benes B, Mech R, Carr N, Stava O, Miller GS (2014) PackMerger: a 3D print volume optimizer. *Computer Graphics Forum* 33(6):322–332
36. Wang, D, Wang, H, Wang Y (2021) Continuity path planning for 3d printed lightweight infill structures. In: 2021 IEEE conference on telecommunications, optics and computer science (TOCS), pp 959–962. IEEE
37. Wang W, Shao H, Liu X, Yin B (2020) Printing direction optimization through slice number and support minimization. *IEEE Access* 8:75646–75655
38. Wei X, Qiu S, Zhu L, Feng R, Tian Yb, Xi J, Zheng Y (2018) Toward support-free 3d printing: a skeletal approach for partitioning models. *IEEE Trans Vis Comput Graph* 24:2799–2812
39. Wüthrich, M, Gubser, M, Elspass, WJ, Jaeger, C (2021) A novel slicing strategy to print overhangs without support material. *Applied Sciences*, vol 11(18)
40. Wu, C, Dai, C, Fang, D, Liu, Y-J, Wang, C (2019) General support-effective decomposition for multi-directional 3-d printing. *IEEE Transactions on Automation Science and Engineering*, **PP**:1–12
41. Wu C, Liu Y-J, Wang CCL (2020) Learning to accelerate decomposition for multi-directional 3d printing. *IEEE Robotics and Automation Letters* 5(4):5897–5904
42. Xu, W, Liu, Y, Yu, M, Wang, D, Hou, S, Li, B, Wang, W, Liu, L (2021) A support-free infill structure based on layer construction for 3d printing. *IEEE Transactions on Visualization and Computer Graphics*, pp 1–1
43. Yang Y, Ohtake Y, Suzuki H (2020) Mesh processing for improved perceptual quality of 3D printed relief. *Journal of Computational Design and Engineering* 8(1):115–124
44. Yu, EA, Yeom, J, Tutum, CC, Vouga, E, Miikkulainen, R (2017) Evolutionary decomposition for 3d printing. In: *Proceedings of the genetic and evolutionary computation conference (GECCO 2017) (Best Paper Award), GECCO '17*, pp 1272–1279. New York, NY, USA. Association for Computing Machinery
45. Demir İ, Aliaga DG, Benes B (2018) Near-convex decomposition and layering for efficient 3d printing. *Additive Manufacturing* 21:383–394
46. Žalik, B, Strnad, D, Kohek, S, Ivana Kolingerová, Andrej Nerat, Niko Lukač, and David Podgorelec (2022) A hierarchical universal algorithm for geometric objects: reflection symmetry detection. *Symmetry*, vol 14(5)

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