Design of lightweight tree-shaped internal support structures for 3D printed shell models

Article *in* Rapid Prototyping Journal · September 2019

DOI: 10.1108/RPJ-04-2019-0108

5 authors, including:

Ruiliang Feng
Shanghai Jiao Tong University

Xiangzhi Wei
The Hong Kong University of Science and Technology

Design of lightweight tree-shaped internal support structures for 3D printed shell models

Lin Zhu, Ruiliang Feng, Xianda Li, Juntong Xi and Xiangzhi Wei
Institute of Intelligent Manufacturing and Information Engineering, Shanghai Jiaotong University, Shanghai, China

Abstract

Purpose – The purpose of this paper is to design a lightweight tree-shaped internal support structure for fused deposition modeling (FDM) three-dimensional (3D) printed shell models.

Design/methodology/approach – A hybrid of an improved particle swarm optimization (PSO) and greedy strategy is proposed to address the topology optimization of the tree-shaped support structures, where the improved PSO is different from traditional PSO by integrating the best component of different particles into the global best particle. In addition, different from FEM-based methods, the growing of tree branches is based on a large set of FDM 3D printing experiments.

Findings – The proposed improved PSO and its combination with a greedy strategy is effective in reducing the volume of the tree-shaped support structures. Through comparison experiments, it is shown that the results of the proposed method outperform the results of recent works.

Research limitations/implications – The proposed approach requires the derivation of the function of the yield length of a branch in terms of a set of critical parameters (printing speed, layer thickness, materials, etc.), which is to be used in growing the tree branches. This process requires a large number of printing experiments. To speed up this process, the users can print a dozen of branches on a single build platform. Thereafter, the users can always use the function for the fabrication of the 3D models.

Originality/value – The proposed approach is useful for the designers and manufacturers to save materials and printing time in fabricating the shell models using the FDM technique; although the target is to minimize the volume of internal support structures, it is also applicable to the exterior support structures, and it can be adapted to the design of the tree-shaped support structures for other AM techniques such as SLA and SLM.

Keywords Additive manufacturing, Particle swarm optimization, Topology optimization, Internal support structure

Paper type Research paper

1. Introduction

Additive manufacturing (AM) or three-dimensional (3D) printing has been widely used as a technique for fabricating arbitrarily complicated geometric features. In AM processes with a fixed build orientation, it requires adding support structures under the overhangs to prevent them from falling down (e.g. in fused decomposition melting processes) or curving up (e.g. in selective laser melting (SLM) processes). Support structures need to be removed after the models are fabricated; therefore, minimizing the support volume can save the consumed materials.

If the precise shape of the model is not of critical concern, altering the shape or topology of the model can save printing materials. For example, Hu *et al.* (2015) proposed an orientation-driven shape optimizer to a self-supported manner to reduce the support volume. Mirzendehdel and Suresh (2016) proposed a robust and efficient algorithm for support structure constrained topology optimization that can optimize at the same time the model shape and the amount of the support volumes. This procedure relies on the topological sensitivity analysis of the support structure quantity. Langelaar (2016) proposed a topology optimization formulation that can exclude unprintable geometries from the design space, resulting in fully self-supporting optimized designs.

Partitioning the model into support-free parts and then assembling them together is an effective means of reducing support materials (Vanek *et al.*, 2016). To print large objects, Luo *et al.* (2012) decomposed a large object into printable parts by considering the issues of printability, assemble-ability and efficiency. Hu *et al.* (2014) decomposed a model into approximate pyramidal parts that can be 3D printed individually without using support materials. Yao *et al.* (2015) developed a level-set method to partition the model using curved surfaces, their partition results in less printing time and a compact packing space for the parts. Chen *et al.* (2015) explored the decomposition and packing space by a prioritized and bounded beam search that is guided by local and global objectives to minimize support material, build time and assembly cost. Khan *et al.* (2018) developed an interactive tool for partitioning a 3D model into printable parts by using an exhaustive search in the binary space partitioning (BSP) tree. Song *et al.* (2016) proposed an approach to fabricate large-scale models by a coarse-to-fine fabrication process with the guarantees of aesthetics, stability and balancing. Recently, Muntoni *et al.* (2018) proposed an algorithm for decomposing general 3D objects into a small set of non-overlapped height field blocks that can be fabricated by three-axis CNC milling or

The current issue and full text archive of this journal is available on Emerald Insight at: **www.emeraldinsight.com/1355-2546.htm**

Rapid Prototyping Journal
25/9 (2019) 1552–1564

[DOI 10.1108/RPJ-04-2019-0108]

Received 18 April 2019
Revised 22 May 2019
Accepted 19 August 2019

3D printing. To save the support materials and printing time, a method was proposed for decomposing a shell model into the least number of support-free parts based on partitioning the Laplacian skeleton of the model with a randomized algorithm (Wei *et al.*, 2018).

When the build orientation is allowed to be altered, choosing a nice build direction may reduce the support volume significantly (Frank and Fadel, 1995; Thrimurthulu *et al.*, 2004; Luo *et al.*, 2016; Das *et al.*, 2017). Particularly, Zhao (2005) proposed a multi-objective function to find an optimal part orientation to minimize volumetric error, build time and support structure. Paul and Anand (2015) proposed a voxel-based algorithm to analyze the flatness and cylindricity errors of the parts with a minimal amount of support structures by choosing the optimal part orientation. To protest the visual impact of the finished parts, Zhang *et al.* (2015) presented a neural network method to find printing directions that avoid placing supports in user-preferred features.

For most engineering applications, the build orientation is fixed; in this case, a dozen of topologies have been proposed for the support structures. The geometric shapes of the traditional structures such as line support, grid support and zigzag support are widely studied for the lightweight design of the support structures (Jiang *et al.*, 2019). In the following, we shall focus on three major structures that are the frontier research topics of the literature; they are the strut structures, the cellular structures and the tree-shaped structures. Among these, the latter two are also explored by Autodesk Netfabb (2019) and Autodesk Meshmixer (2014).

In terms of strut structures, Dumas *et al.* (2014) proposed a support structure consisting of horizontal and vertical branch for fused deposition modeling (FDM), where the horizontal branch were successfully generated by controlling the printing speed. Shen *et al.* (2016) proposed a bridge support structure generation algorithm based on a sweeping strategy of the nozzle; this approach first enumerated and selected horizontal bridges through a scoring function and then connected the bridges and model with a series of vertical pillars. Wang *et al.* (2013) and Wang *et al.* (2017) proposed a self-supporting strut frame to support skin-frame models and designed a sparsity optimization framework to ensure the printability of models while guaranteeing a given structural strength. Gan and Wong (2016) explored Y-shaped, inverse Y-shaped and a combination of vertical lines and Y-shaped support structures for manufacturing thin plates and cuboids in SLM.

In terms of cellular structures, lightweight and porous cellular structures have been proved to be effective in saving the support materials for metal AM (Wang *et al.*, 2005; Dong *et al.*, 2017). To minimize the support volume and meanwhile taking care of thermal stress and distortion, Hussein *et al.* (2011) explored two types of lattice support structures (i.e. diamond and gyroid) to reduce the material and fabrication time. However, the low volume fraction of cellular structure may be too fragile to be consistently manufactured with an SLM process at the desired resolution (Hussein *et al.*, 2013). Strano *et al.* (2013) proposed a graded cellular support structure where dense cells were placed under the heavy overhangs, and sparse cells were placed under the light overhangs. Vaidya and Anand (2016) discretized the free space of the model into a set of unit cells, and then used Dijkstra's shortest path algorithm (Dijkstra, 1959) to generate shortest paths to connect a sample of the interface cells (touching points) below the overhangs to proper cells that were on the model surface or the substrate. However, the shortest path system may not correspond to a minimum volume system. Lu *et al.* (2014) proposed a honeycomb-cell structure based on a Voronoi-diagram algorithm; the structure was further tessellated to maximize the hollowing space while sustaining the structural strength of the model. Wu *et al.* (2016) introduced rhombic structures as self-supporting infill structures. Lee and Lee (2016) proposed an inner support-free structure generation algorithm based on a 3D block partitioning method to reduce the manufacturing time and the amount of material. Mao *et al.* (2018) proposed a novel scheme of generating hybrid interior support structures that mainly consist of sticks and balls (at the junctions of the sticks) to achieve a fairly high strength-to-weight ratio.

In terms of tree-shaped structures, Zhang *et al.* (2015) proposed an internal supporting frame based on a medial axis tree, where tree-shaped branches were rooted on the medial axis with dense leaves supporting the regions where a large structural strength was required and coarse leaves for the other parts of the model. But, their approach may not be applied to external support structures because no medial axis is available for external free space. Wei *et al.* (2016) presented an algorithm (in Chinese) for generating a sparse tree support, which may be further improved if surface quality is considered in the future; in their design of the tree support, a branch is a flat wall with large volume, which can be further reduced. Vanek *et al.* (2015) showed that the problem of generating a tree support with minimum length is NP-hard. They also proposed a greedy algorithm to grow tree-shaped support structures from top to down, but no topology optimization was conducted to minimize the support volume. Zhao *et al.* (2015) used a particle swarm optimization (PSO) algorithm to minimize the contacting areas of the vertical supporting lines, which can be viewed as a special tree-shaped structure without branches. However, the connection topology of the support lines was not optimized. Zhu *et al.* (2019) proposed a hybrid of traditional PSO and greedy algorithm for reducing the support volumes. Zhang *et al.* (2018) investigated the influence of different support parameters on the efficiency and mechanical properties of tree supports for SLM, but the tree supports are simple ones with single internal nodes. Liu *et al.* (2019) proposed an finite element method-based (FEM-based) algorithm for generating tree supports to save printing materials; however, their work is restricted to two-dimensional (2D) problems. Boyard *et al.* (2017) proposed an octree's discretization of the part and represent the supports in a matrix form and set voids to the matrix to reduce the support volume. Although they pointed out the possibility of transforming the support structures into trees, but no implementation is discussed.

In this paper, we shall focus on the problem of supporting the interior of a shell model with the least amount of materials. Refer to Figure 1 for an example of a shell model. For this topic, Wang *et al.* (2017) recently proposed a support-free hollowing framework that sliced the models into layers and then hollowed the interior of each layer. However, it is unclear whether the proposed approach can archive a minimum amount of infill structures. In this paper, we propose a tree-shaped infill structures with less support materials. This is a step toward minimum infill structures for shell models.

The major contributions of the paper are as follows:

- A tree-shaped infill structure is proposed, which guarantees the printability of the shell models while using less volume than existing approaches do.
- In terms of the methodology, an improved PSO is proposed to realize the volume minimization task; different from the traditional PSO process that uses the best solution (a single particle) as a global best particle (denoted as *gBest*) for the current swarm, our *gBest* is generated by combining the best ingredients of different particles. Furthermore, in this work, we combine PSO with greedy strategy by linking an internal node to the model surface as well as an already existing tree branch whenever this results in a support structure with less volume.

The rest of the paper is organized as follows: Section 2 presents of an improved PSO approach for minimizing the infill tree-shaped structures for 3D printed shell models; Section 3 presents our simulation results; Section 4 presents our experimental results and comparisons with other approaches; Section 5 closes the paper with some discussion.

2. Optimization of tree supports for three-dimensional shell models

In this section, we present our approach for constructing a tree support with minimum volume for an arbitrary 3D shell model. Given a shell model M with inner surface M', our algorithm for growing the tree support for the overhangs on M' is given as follows:

Algorithm: Generate_Tree_Support (*M*, *H*)

```
Input: A shell model M with interior surface M';
Output: A tree-support with minimum volume;
Step 1: Compute the support area of M' and generate
        a uniform sampling of points in the area as
        the leaves of the tree supports;
Step 2: Construct a uniform grid G in the free
        space of M', generate a random set of I
        distinct tree supports by using the nodes
        of G;
Step 3: For each tree support obtained in Step 2,
        perturb it into a swarm of N distinct
        structures with the same connection
        topologies by moving the internal tree
        nodes vertically up and down;
Step 4: Run PSO for each swarm: For each swarm
        obtained in Step 3, run the PSO on it for H
        iterations: for the k-th iteration (k = 1,
        2, ..., H), gBest^k is derived as the combi
        nation of the best component of each
        pBest_i^k; run a greedy strategy to link the
        internal nodes of the tree support to the
        model surface or the other internal nodes
        with a branch with less volume.
```

The details of each major step are given as follows:

In Step 1, refer to Figure 2, we can detect the following three types of overhangs: the point overhang, the edge overhang and the face overhang. For a face overhang, it is considered as an overhang to be supported if angle θ is less than a threshold angle θ_c, which is chosen to be 35° for our models. In general, one can choose the threshold value based on the experimental results of a particular machine.

In Step 2, G is a grid with uniform edge length r, and r is determined as 3.5 mm in our case, which is also the safe horizontal bridge between two supporting points in our FDM 3D printer by experiments. In general, to derive a more accurate result, r can be determined as a smaller value for the non-supporting points (on the overhang surface) of G, but this will increase the running time of the algorithm. The initial tree

Figure 2 Illustration of point, edge and face overhangs

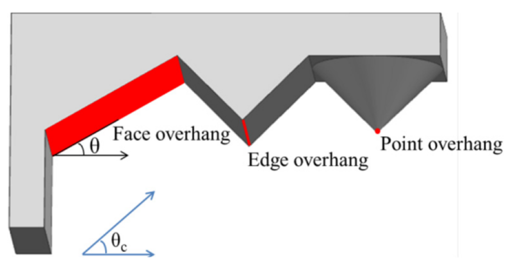
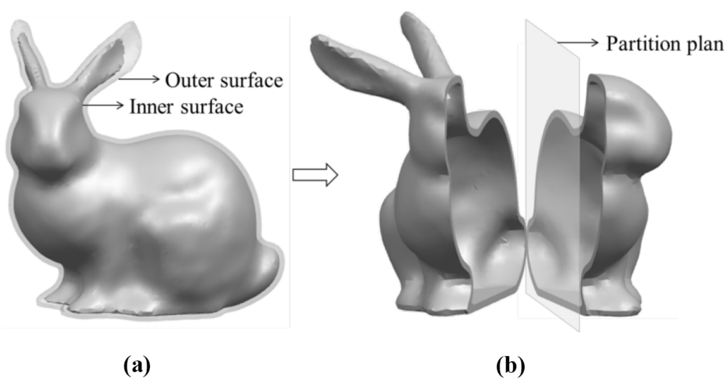


Figure 1 Illustration of a shell model



Notes: (a) A shell model with uniform skin thickness; (b) the interior of the shell model topologies are obtained by connecting the nodes of G where a node is connected to only one lower node to guarantee the tree structure, i.e. no loop is formed in the structure. Refer to Figure 3(a) for an illustration. $I = 100$ initial topologies can be selected as the ones with minimum volumes from a large set of topologies (e.g. 500 in our case); this follows from the convention that the initial number of guess solutions for the PSO is less than 100 for most applications (Coello *et al.*, 2002; Coello *et al.*, 2004).

In Step 3, refer to Figure 3(b), the positions of the internal tree nodes can be moved vertically up and down to obtain N new variants of an initial tree topology.

For Step 4, we shall first introduce traditional PSO, and then we show how to improve it to generate nice tree supports.

2.1 Traditional particle swarm optimization

PSO was first introduced by Kennedy and Eberhart (1995); it mimics the swarm behavior of the birds in finding foods: when a bird finds food, its neighboring birds are attracted to its vicinity and the swarm flies to the region of food gradually. In this approach, there is a built-in learning mechanism: the birds learn the position of the current best bird and try to converge to it. PSO can be hybridized with different methods such as chaos, genetic algorithm and ant colony optimization to increase the success rate (Liu *et al.*, 2005; Kiran *et al.*, 2012). PSO has found its powerful applications in a number of applications, e.g. finding the shortest path, task assignment, shop scheduling and so on. Particularly, in the application of AM, it has been shown to be effective in reducing the touching areas of the supporting points on the overhangs (Zhao *et al.*, 2015). In this section, to generate the structure of the tree supports, we adopt the PSO strategy by adding some geometric constraints and a greedy strategy to the optimization system.

2.2 Particle swarm optimization adapted for our tree supports

To implement the PSO, we need to define the meaning of "particle" first. In a PSO process, a particle is a guess solution to the problem; in our case, it is a support structure to the overhangs of the model. Refer to Figure 3; a tree support may contain multiple trees as the support structures for an overhang.

Each particle P_i is defined as $P_i = [P_{i1,\ldots,}P_{ij,\ldots,}P_{im}]$, where P_{ij} is the j-th tree of P_i, m is the trees number of P_i and P_i is associated with two vectors, i.e. the velocity vector $V_i = [V_{i1,\ldots,}V_{ij,\ldots,}V_{im}]$, and the position vector $X_i = [X_{i1,\ldots,}X_{ij,\ldots,}X_{im}]$, where $V_{ij} = [v_{ij}^{1}, v_{ij}^{2},\ldots, v_{ij}^{Dij}]$, $X_{ij} = [z_{ij}^{1}, z_{ij}^{2},\ldots, z_{ij}^{Dij}]$, D_{ij} is the number of branch nodes in the P_{ij}, v_{ij}^{k} denotes the velocity components of z_{ij}^{k}, z_{ij}^{k} denotes the z-coordinates of the P_{ij} in the k-th dimension.

The initial positions of the particles are given by N tree supports, and the initial velocity is given as $[0, 0, \ldots, 0, 0]$. As

Figure 3 (a) A tree support with two trees generated from the nodes of G; (b) a variant of the tree support is obtained by perturbing the internal nodes of the tree-support in (a) vertically up and down

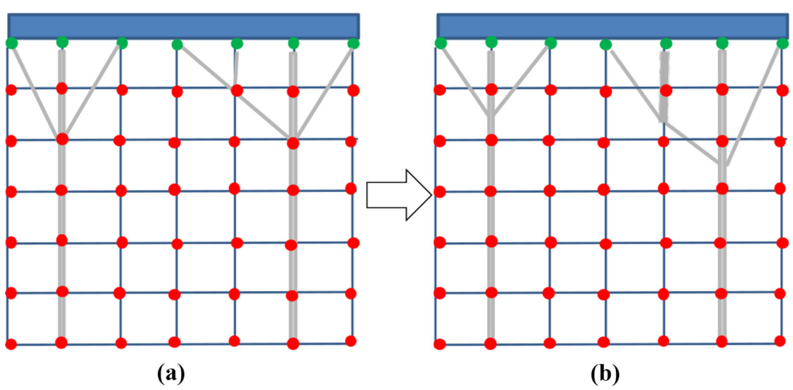


Figure 4 A tree support (a) is perturbed into a new tree support with local minim point (b)

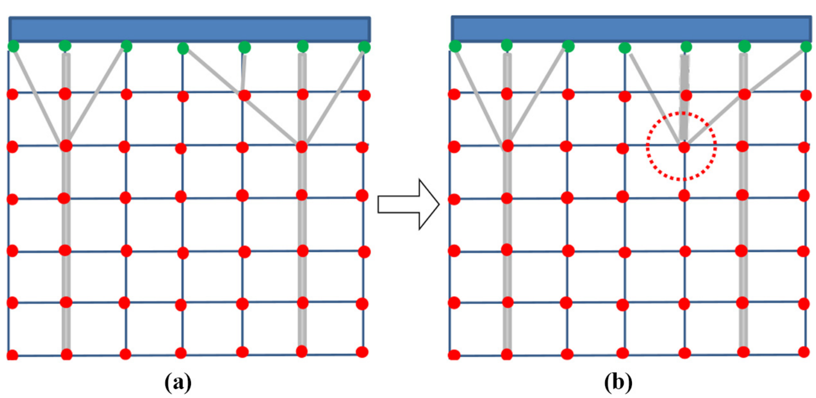
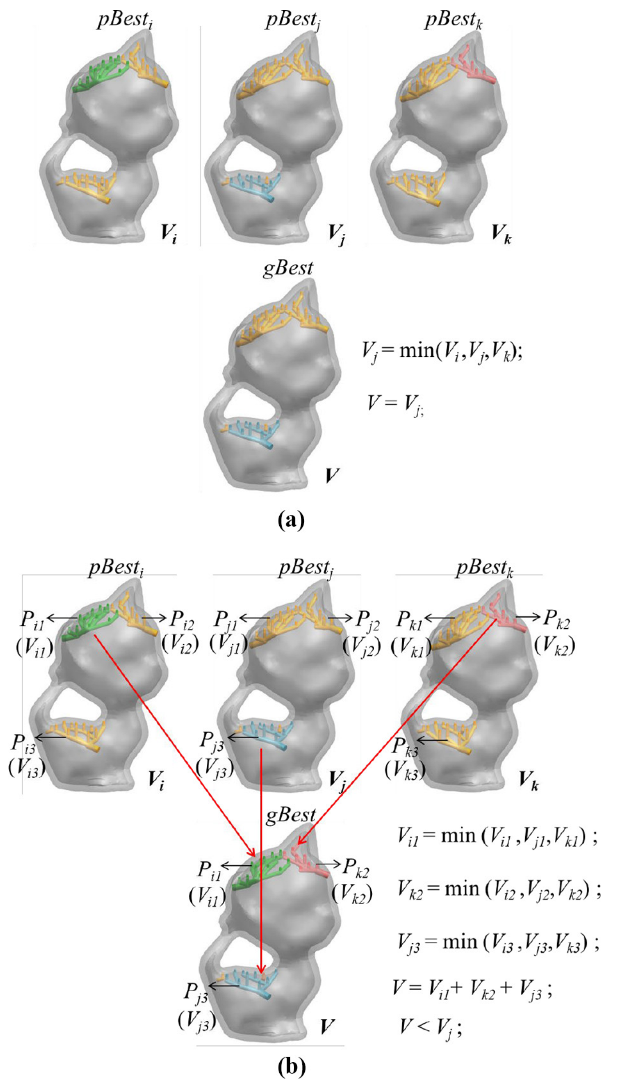


Figure 5 (a) *gBest* is derived from the best particle in the traditional PSO; (b) *gBest* is derived from the best component of different particles in our improved PSO



the evolution goes on, the velocity and position of P_i in the *k-th* iteration can be updated as follows:

$$V_i^k = w * V_i^{k-1} + c_1 * r_1 * \left(pBest_i^k - X_i^{k-1}\right) + c_2 * r_2 * \left(gBest^k - X_i^{k-1}\right) \quad (1)$$

$$X_i^k = X_i^{k-1} + V_i^k \quad (2)$$

where V_i^k is the velocity of P_i in the k-th iteration, X_i^k is the new state of P_i in the k-th iteration, w is the inertial weight used to control the influence of the previous velocity, $pBest_i^k$ is the historically best position of particle i ($i = 1, 2, \ldots, N$), $gBest^k$ is the historically best position of the entire swarm, c_1 and c_2 are two parameters to weigh the influence of $pBest_i^k$ and $gbest^k$, respectively. Equation (2) means that the current position of a particle (the z-coordinates of the branch of a tree support, i.e. X_i^k) is obtained by changing the previous position of the particle (X_i^{k-1}) by adding a perturbing vector (V_i^k). Here, the perturbing amount for each entry of V_i^k is determined by equation (1), which is the weighted sum of the current best position of the swarm ($gBest^k$) and the historically best position of P_i ($pBest_i^k$).

To implement the PSO scheme, following the convention (Liu *et al.*, 2005; Kiran *et al.*, 2012; Zhu *et al.*, 2019), c_1 and c_2 are set as 2.0, and r_1, r_2 are two independent random variables in the range of (0, 1], the inertial weight w is linearly decreasing from 0.9 to 0.4, which means that the swarm converges fast in the later iterations.

To realize our target, we choose the support volume as the fitness function of the PSO: the support structure with the minimum volume is chosen as the *gBest*. Particularly, let d_i be the diameter of the branch connecting a node i downward, and let l_i be the length of the branch; we can express the objective function as follows:

$$\textbf{Objective function}: \min \sum \pi (d_i/2)^2 l_i \quad (3)$$

To guarantee that the topology of the support remains a tree-shaped structure, it requires that a node of the tree is linked to a unique node below it, i.e. if D_i is the degree of a node that consists of the lower neighbors, then we have the constraint as follows:

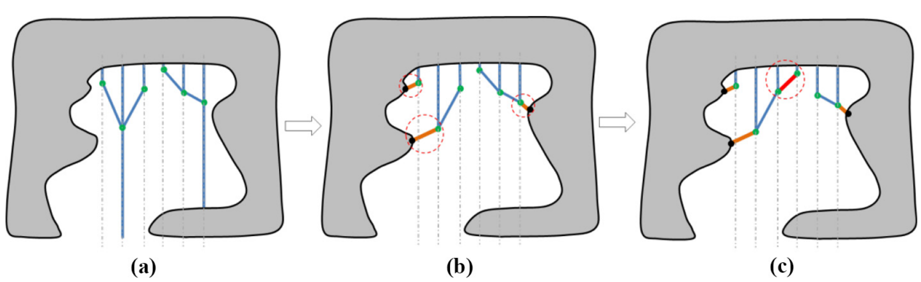
$$D_i = 1 \quad (4)$$

Furthermore, to make sure that the tree is a stable one, it requires a lower branch to be smaller than an upper branch in terms of diameter. More precisely, let d_i be the diameter of a cylindrical link connects to node i from below, then we have the following constraint:

$$d_i \leq d_j \quad (5)$$

where nodes i and j are connected, and node i is higher than node j.

Figure 6 (a) A tree support; (b) a new tree-support obtained by linking three internal nodes of the tree support in (a) to the model surface; (c) a new tree support obtained by linking a tree node of the tree support in (b) to another tree node



Rapid Prototyping Journal
Volume 25 · Number 9 · 2019 · 1552–1564

To avoid fragile features such as thin walls and sharp corners that may result in part failures during the 3D printing process (Adams and Turner, 2018), we design the tree branches using cylindrical bars, and we design the joint of each fork using a solid sphere that contains all the incident branch-ends inside, where the radius of the sphere is the radius of the lower (and unique) cylindrical bar.

In addition, we need to build up the function of the length of a tree branch (a cylindrical bar) l in terms of the process parameters. Note that factorial design method has been proved to be an effective approach for determining the factors that influence the yields for various applications in AM (Vanek *et al.*, 2015; Lanzotti *et al.*, 2015). Therefore, we exploited the factorial design method by expressing l in terms of the diameter of the branch (x_1), the printing layer thickness (x_2) and the tilted angle of the branch(x_3). When a branch has more branches grown on its tip, the stability of the branch may be weakened, and a safety ratio is required to constraint the growing length. Therefore, we set the safety ratio as 0.7 for all levels of the branches on the tree. In general, one can conduct a

Figure 7 Examples of the tree supports obtained by a hybrid of the improved PSO and greedy strategy: (a) linking a node with a branch with less volume to another node of the same tree; (b) linking a node with a branch with less volume to a node of another tree

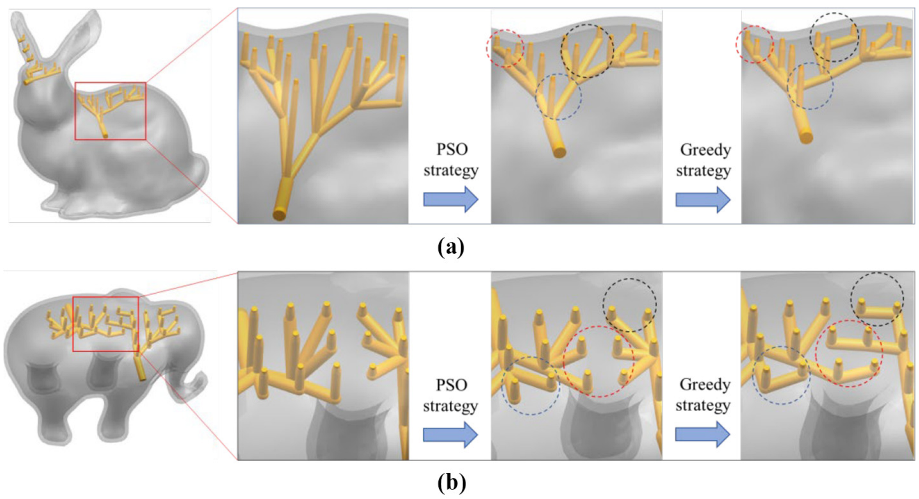


Figure 8 Illustration of the simulation processes for a set of models, from left to right: the initial state, the 100th iteration, the 500th iteration, the 2,000th iteration, the effects of the final internal support structures

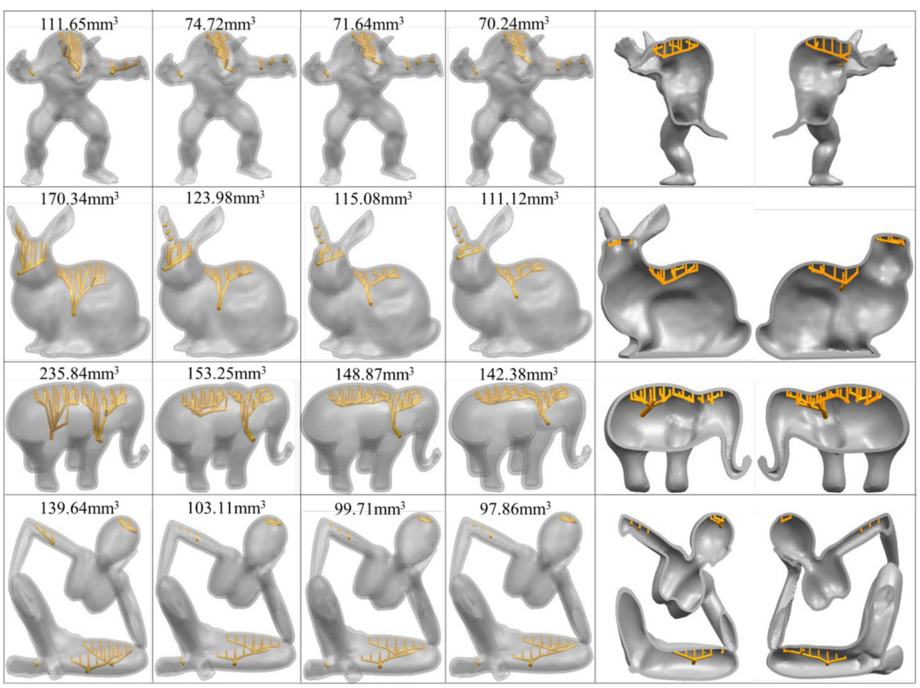
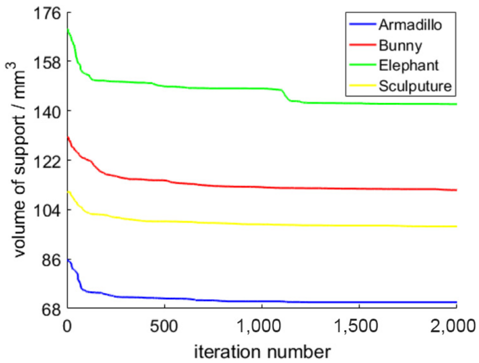


Figure 9 The curve of support volume with respect to the number of iterations



simple 3D printing experiment for tree supports to determine a reasonable safety ratio. To summarize, we have the following expression for l:

$$l \leq 0.7(-66.17 + 145.9x_1 + 113.4x_2 + 1.447x_3 - 98.0x_1^2 - 0.00721x_3^2 + 54.1x_1x_2 - 0.870x_1x_3 - 4.581x_2x_3 + 28.42x_1^3 - 0.2558x_1^2x_3 + 0.01273x_1x_3^2 + 1.297x_1x_2x_3) \quad (6)$$

The experiment setting is as follows: nozzle temperature: 200° C; deposition speed: 60 mm/s for the infill and 20 mm/s for the contour healing; machine: Zortax M200 FDM 3D printer; material: ABS; layer thickness(x_2): 0.14 mm. When recording the data, the yield length (l) is determined as the distance from the bottom of a branch to the first point where the branch starts to deform. The printing experimental results with different groups of parameters are available in the submitted supplemental file.

For users with different 3D printers and materials, the function of l can be derived by conducting 3D printing experiments of a large batch of branches on a single build platform each time. Once the function of l is determined, the users can use it to fabricate the 3D models with tree supports all the time.

In general, the factors influence l can also include temperature, materials, etc. Although considering these factors can guarantee a more precise result, it will require tremendous efforts in conducting the experiments. By a reasonably large set of 3D printing experiments (more than 150), we find that equation (6) is sufficient to estimate the growth of a tree branch.

In the iteration process, if the no new local minimum points are introduced, then the tree is self-supported and is, therefore, valid for the support structure. Otherwise, some new local minimum is generated, and the tree is not self-supported (Figure 4), and the particle resumes to its shape in the previous iteration.

2.3 A novel approach for deriving gBest of particle swarm optimization

Refer to Figure 5(a); in the evolution of a swarm, traditionally *gBest* is taken as the best particle ever found. Refer to Figure 5(b); to speed up the convergence process of to achieve a better result, we take the best component of different components and integrate them into *gBest*. More precisely, if V_{ij} is the minimum volume of a tree (a component of a tree support) among all $pBest_i$ (for all i), then its corresponding tree P_{ij} is taken into *gBest*.

2.4 A hybrid of the improved particle swarm optimization with a greedy strategy

To enable a topology change such that the support volume may be further reduced, we integrate a greedy strategy into the PSO process, which links an internal node to a near point on the model surface or an internal node whenever this results in a support structure with less volume (Figures 6(b-c)). As shown in Figure 7, when a node is linked to another node of the tree support, the linked nodes may belong to different trees of a tree support (Figure 7(b)). Note that calling the greedy strategy may interrupt the PSO process and slow down the convergence rate. To mitigate this, the greedy strategy is called for every interval of 100 iterations, which allows the PSO to run in a stable manner for a relatively long time.

When this topology change leads to a reduction of support volume, then this topology is used as the seed of a new swarm of N particles; the swarm that performs the worst is replaced by the new swarm, and the remaining number of iterations of the worst swarm is conducted on the new swarm instead. In this manner, the total number of particles used in the iterations are maintained the same, and a better result is derived by using almost the same computation time.

3. Simulation and experiment result

We implemented the algorithm with Matlab in a PC with Intel i7-4790 and 8 GB RAM. We set $I = 100$ as the number of initial warms. For each initial tree support (a particle), we perturbed it into $N = 100$ distinct tree supports while maintaining their topologies as the same. Note that $I = N = 100$ is sufficient for the evolution of the PSO process by convention (Coello *et al.*, 2002; Coello *et al.*, 2004); further, we set the maximum number of iterations as $H = 2{,}000$. In practice, I can be determined by choosing the best 100 particles from a large sample of particles (e.g. 500 in our implementation). Based on

Table I Simulation statistics of the example models

Models	Dimensions (mm * mm * mm)	Facet number	Running time (min)	Support volume (mm^3)
Armadillo	53.88 * 46.30 * 63.92	35,014	8.32	70.24
Bunny	59.31 * 44.73 * 56.26	35,232	9.82	111.12
Elephant	46.19 * 25.13 * 32.48	36,310	18.74	142.38
Sculpture	58.44 * 27.72 * 69.21	39,718	13.86	97.86

printability experiments, the minimum branch diameter is set as 0.8 mm; this suffices to start the optimization system of equations (1)-(6). In addition, to ensure the printability of a shell model the wall thickness is set as 0.8 mm (Wang *et al.*, 2017).

We randomly chose four commonly used computer graphics models for the simulation (Figure 8); the models with tree supports are available on the GitHub website[1]. The support volume curve of each model is provided in Figure 9. We can see that the support volume remains constant after running the simulation for more than 1,500 times, which means that perturbing the positions of the tree nodes makes little difference to the result, and the result is a fairly stable one (corresponding to the choice of the initial set of swarms). The simulation statistics of the models are summarized in Table I.

To investigate the effects of the tree supports generated by our approach, we conducted a set of 3D printing experiments using these models. Figure 10 shows the effects of the 3D printed models. From this figure, we can see that the supported models can be successfully manufactured with infill structures of fairly small amount of materials while guaranteeing the fabrication stability and surface quality.

4. Comparison experiments

To validate our proposed method, we compared the support volume reduction of our approach with two methods:

1 hybrid of traditional PSO and greedy strategy (Zhu *et al.*, 2019); and
2 support hollowing method (Wang *et al.*, 2017).

Figure 11 illustrates the process of optimizing the support structures by using the hybrid method of traditional PSO and

Figure 10 The 3D printing processes of the models designed by our proposed approach, from the first row to the last row: the armadillo model, the bunny model, the elephant model and the sculpture model

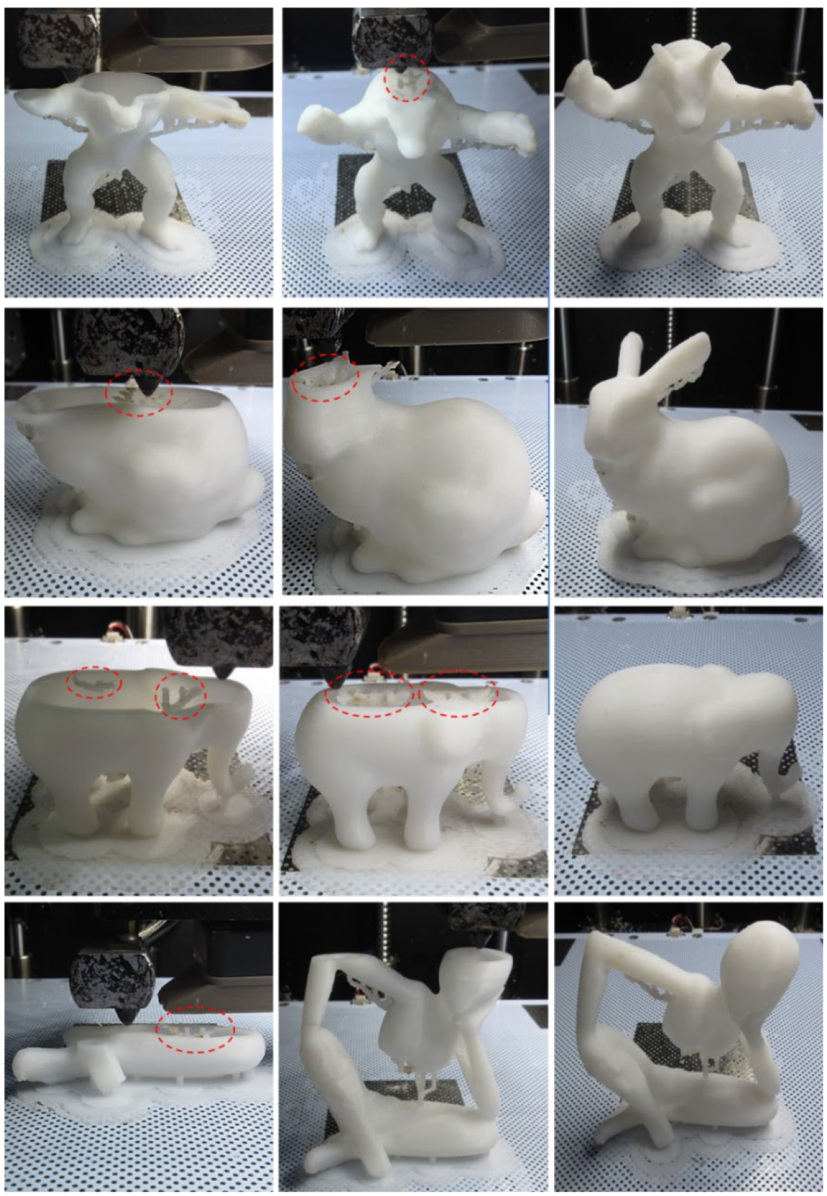
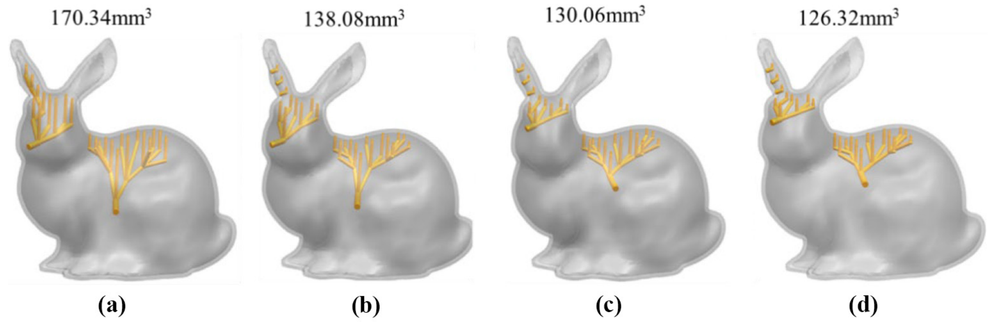


Figure 11 Illustration of the results generated by hybrid of traditional PSO and greedy strategy



Notes: (a) The initial state; (b) the 100-th iteration; (c) the 500-th iteration; (d) the 2000-th iteration

greedy strategy, where the PSO scheme used traditional method for deriving *gBest* as the best particle of the evolution in each iteration (Figure 5(a)) and the greedy scheme did not link a node to any existing node of a tree support (Figure 6(b)). Taking the bunny model for example, the curves of support volumes of the hybrid of traditional PSO and greedy strategy and our approach are provided in Figure 12; it can be seen that our proposed approach saves significantly more volume. As shown in Figure 13, our approach generates the support structure with optimized topology structure.

Figure 12 The curves of support volumes of the bunny model for a hybrid of traditional PSO and greedy strategy and our improved PSO and greedy strategy

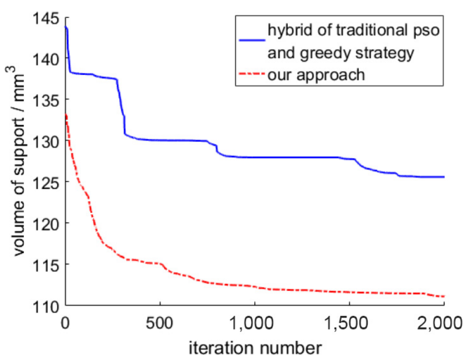


Table II summarizes the statistics of the experiments on four models in Figure 8 (and Table I). Our proposed approach can achieve a larger support volume reduction than traditional PSO and greedy strategy (6.19-18.39 per cent). Although it looks as if the reduced volume increases as the support volume increases, it does not mean that our approach is more significant for any shell model with large support volume. If a large shell model contains a tall support space (i.e. the space that is vertically beneath of the support areas), then our tree-shaped supports are more useful because they can grow tree branches in a more flexible and material-saving manner. When the vertical support space is very short (e.g. less than 5 mm), our tree-shaped supports might be less useful because the trees degenerate into a set of vertical bars, even if the entire support volume is very large.

To further validate our proposed method, we also compared the save time and save volume of our method with the support-free hollowing method of Wang *et al.* (2017). To make the comparison a fair one, we used the same models, the same tilted angle (i.e. 45°) and the same shell thickness (i.e. 0.8 mm) in our experiments. Figure 14 shows the simulation results on eight different models provided by Wang *et al.* (2017). The computational statistics are summarized in Table III. The

Figure 13 A comparison of our approach with the work of a hybrid of traditional PSO and greedy algorithm: (a)-(d) are the tree supports obtained by using a hybrid of traditional PSO and greedy algorithm; (e)-(h) are the tree supports obtained by our proposed approach

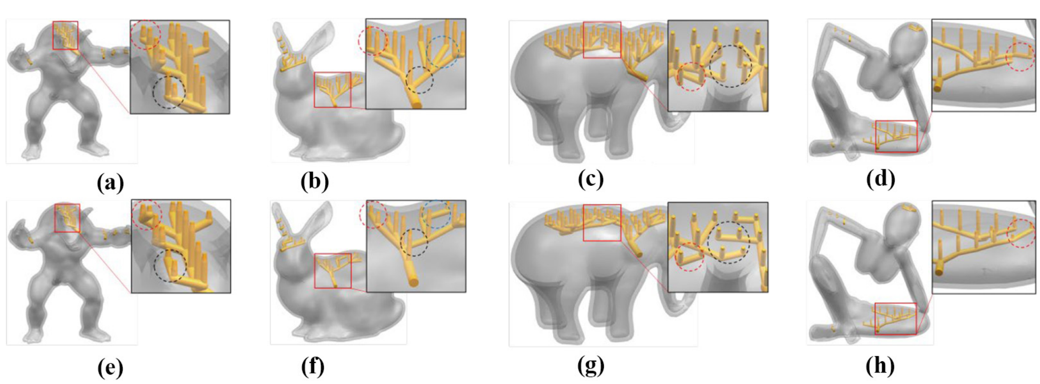
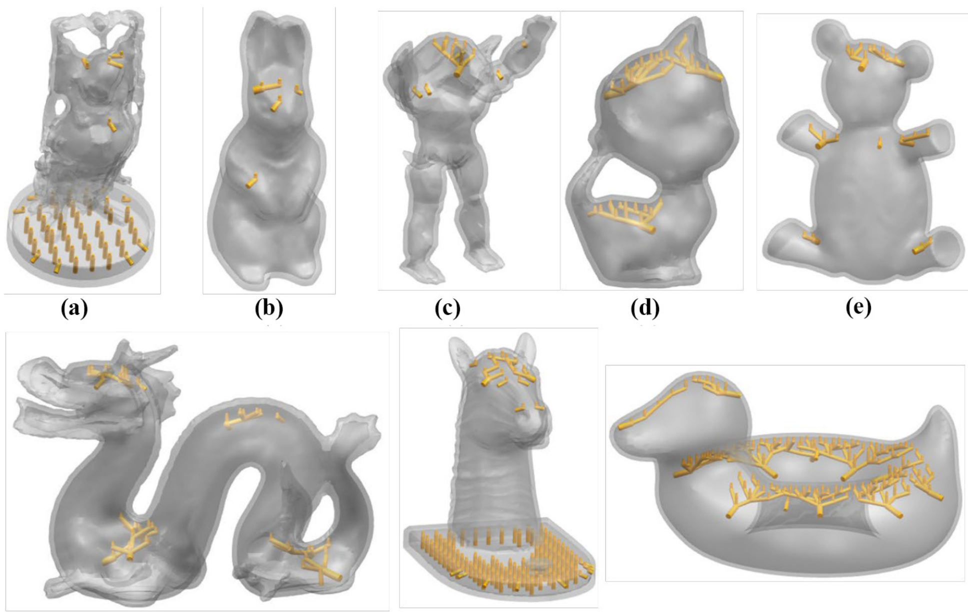


Table II Comparison of internal supports volume with the hybrid of traditional PSO and greedy algorithm

Models	Hybrid of traditional PSO and greedy strategy Support volume (mm^3)	Our approach Support volume (mm^3)	Reduced volume (%)
Armadillo	74.88	70.24	6.19
Bunny	126.32	111.12	12.03
Elephant	174.47	142.38	18.39
Sculpture	104.46	97.86	6.32

Figure 14 The example models provided by Wang *et al.* (2017) with the tree supports generated by our approach



support volume is calculated by using Autodesk Meshmixer. From the table, we can see that the save volume of our method is about 3-10 per cent larger than the save volume of the support-free hollowing method of Wang *et al.* (2017). As shown in Figure 15, the new method presented in this paper is able to generate the internal supports with large volume reduction, and the supports can be successfully 3D printed.

5. Conclusion and discussions

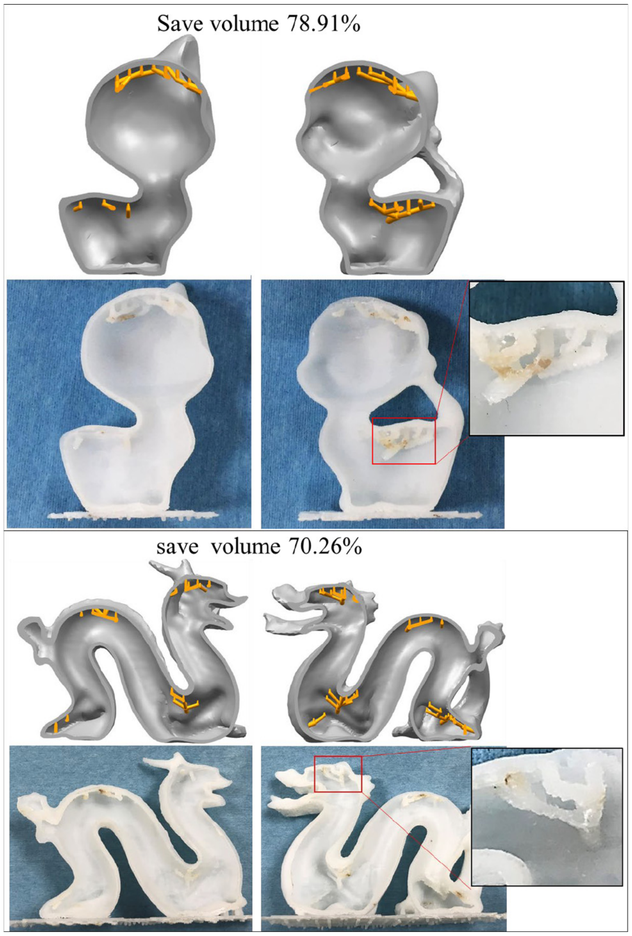
In this paper, we propose a tree-shaped structure to support the overhangs of the inner surface of an arbitrary shell model. We also propose an improved PSO scheme by integrating a greedy strategy to address the problem of designing a tree support of minimum volume, and the growing of tree branches is based on the statistics of FDM 3D printing of a large set of tree branches.

Table III Comparison with the support-free hollowing method of Wang *et al.* (2017)

Models	Input model volume (cm^3)	Support-free hollowing method Output model volume	Support-free hollowing method Reduced volume (%)	Our method Output model Volume (cm^3)	Our method Reduced volume (%)
Buddha	7.99	3.05	61.83	3.79	65.10
Rabbit	9.46	2.51	73.47	2.18	76.96
Armadino	10.16	3.77	62.90	3.16	68.90
Kitten	15.6	3.85	75.33	3.29	78.91
Bear	20.39	4.79	76.51	4.34	78.72
Dragon	27.35	9.14	66.56	8.134	70.26
Horse	33.99	9.44	72.23	7.22	78.76
Duck	86.11	20.26	76.35	12.44	85.56

Note: The unit of volume is set as "cm^3" as in Wang *et al.* (2017)

Figure 15 The 3D printed kitten model and dragon model with tree supports generated by our proposed approach



During the evolution of the PSO, we also propose to use the best components of different particles to achieve a nicer global best particle to help the iterations converge at a better result in a short time. Through comparison with the work of using traditional PSO with a greedy strategy, and a recent work of support hollowing method (Wang *et al.*, 2017), the experimental results show that our proposed approach can lead to less support volume for 3D printed shell models. However, Wang *et al.* (2017) consider more comprehensive conditions, such as making the final 3D models static stable (i.e. standing stably by itself), which are not considered in our work.

Although our target application is designing internal support structures, our proposed approach can be directly applied to the design of exterior support structures without any problem. Because multiple forms of exterior support structures have been proposed and hence comparison works require significant efforts, this can be investigated in the future work.

Furthermore, whether our proposed approach realizes a minimum support volume is worthy of future research because it depends on the initial choice of seed particles. Although using a large set of initial particles can lead to fairly nice result (the support volume is almost reduced to constant after a large number of iterations), how to use fewer initial particles and fewer iterations to make the swarm converge is also worthy of future research.

Finally, taking the constraint of heat distribution into consideration, our proposed method can also be adapted to design tree-shaped structure to support the interior of the 3D printed models during the powder-bed AM processes (e.g. SLM), and that our approach enables an efficient removal of the trapped powders by drilling a hole into the shell model (Wei *et al.*, 2019). Multiple voids generated by hollowing different portions of a solid model (e.g. by the method proposed in Wang *et al.* (2017)) can be used in FDM techniques, but its adaption to the powder-bed AM processes may require drilling a hole for each void for the removal of the powders, which may require more post-processing efforts.

Note

1 https://github.com/FengRuiliang/models/tree/Shell

References

Adams, D.W. and Turner, C.J. (2018), "An implicit slicing method for additive manufacturing processes", *Virtual and Physical Prototyping*, Vol. 13 No. 1, pp. 2-7.

Autodesk netfabb™ (2019), available at: www.autodesk.com/products/netfabb/overview

Boyard, N., Christmann, O., Rivette, M., Kerbrat, O. and Richir, S. (2017), "Support optimization for additive manufacturing: application to FDM", *Rapid Prototyping Journal*, Vol. 24 No. 1, pp. 28-36.

Chen, X., Zhang, H., Lin, J., Hu, R., Lu, L., Huang, Q., Benes, B., Cohen-Or, D. and Chen, B. (2015), "Dapper: decompose-and-pack for 3D printing", *ACM Transactions on Graphics*, Vol. 34 No. 6, Article No. 213.

Coello, C.A.C., Lamont, G.B. and Veldhuizen, D.A.V. (2002), "*Evolutionary Algorithms for Solving Multi-Objective Problems*", 1st ed., Kluwer, Norwell, MA, ISBN 0-3064-6762-3.

Coello, C.A.C., Pulido, G.T. and Lechuga, M.S. (2004), "Handling multiple objectives with particle swarm optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 8 No. 3, pp. 256-279.

Das, P., Mhapsekar, K., Chowdhury, S., Samant, R. and Anand, S. (2017), "Selection of build orientation for optimal support structures and minimum part errors in additive manufacturing", *Computer-Aided Design and Applications*, Vol. 14 No. sup1, pp. 1-13.

Dijkstra, E.W. (1959), "A note on two problems in connexion with graphs", *Numerische Mathematik*, Vol. 1 No. 1, pp. 269-271.

Dong, G., Tang, Y. and Zhao, Y.F. (2017), "A survey of modeling of lattice structures fabricated by additive manufacturing", ASME, *Journal of Mechanical Design*, Vol. 139 No. 10, p. 100906.

Dumas, J., Hergel, J. and Lefebvre, S. (2014), "Bridging the gap: automated steady scaffoldings for 3d printing", *ACM Transactions on Graphics*, Vol. 33 No. 4, Article No. 98.

Frank, D. and Fadel, G. (1995), "Expert system-based selection of the preferred direction of build for rapid prototyping processes", *Journal of Intelligent Manufacturing*, Vol. 6 No. 5, pp. 339-345.

Gan, M.X. and Wong, C.H. (2016), "Practical support structures for selective laser melting", *Journal of Materials Processing Technology*, Vol. 238, pp. 474-484.

Hu, K., Jin, S. and Wang, C.C.L. (2015), "Support slimming for single material based additive manufacturing", *Computer-Aided Design*, Vol. 65, pp. 1-10.

Hu, R., Li, H., Zhang, H. and Cohen-Or, D. (2014), "Approximate pyramidal shape decomposition", *ACM Transactions on Graphics*, Vol. 33 No. 6, pp. 1-12.

Hussein, A., Hao, L. and Yan, C. (2013), "Advanced lattice support structures for metal additive manufacturing", *Journal of Materials Processing Technology*, Vol. 213 No. 7, pp. 1019-1026.

Hussein, A., Yan, C., Everson, R. and Hao, L. (2011), "Preliminary investigation on cellular support structures using SLM process", *In Innovative Developments in Virtual and Physical Prototyping*, Taylor & Francis Group, London, pp. 609-612.

Jiang, J., Lou, J. and Hu, G. (2019), "Effect of support on printed properties in fused deposition modelling processes", *Virtual and Physical Prototyping*, Vol. 14 No. 4, available at: https://doi.org/10.1080/17452759.2019.1568835

Kennedy, J. and Eberhart, R.C. (1995), "Particle swarm optimization", *Processing of the IEEE International Conference on Neural Networks*, *Perth, Australia*, pp. 1942-1948.

Kiran, M.S., Gündüz, M. and Baykan, O.K. (2012), "A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum", *Applied Mathematics and Computation*, Vol. 219 No. 4, pp. 1515-1521.

Langelaar, M. (2016), "Topology optimization of 3D self-supporting structures for additive manufacturing", *Additive Manufacturing*, Vol. 12, pp. 60-70.

Lanzotti, A., Martorelli, M., Staiano, G. and Martorelli, M. (2015), "Understanding process parameter effects of RepRap open-source three-dimensional printers through a design of experiments approach", *Journal of Manufacturing Science and Engineering*, Vol. 137 No. 10, p. 011017.

Lee, J. and Lee, K. (2016), "Block-based inner support structure generation algorithm for 3D printing using fused deposition modeling", *International Journal of Advanced Manufacturing Technology*, Vol. 89 Nos. 5/8, pp. 1-13.

Liu, Y., Li, Z., Wei, P. and Chen, S. (2019), "Generating support structures for additive manufacturing with continuum topology optimization methods", *Rapid Prototyping Journal*, Vol. 25 No. 2, pp. 232-246.

Liu, B., Wang, L., Jin, Y., Tang, F. and Huang, D. (2005), "Improved particle swarm optimization combined with chaos", *Chaos Solitons & Fractals*, Vol. 25, pp. 1261-1271.

Lu, L., Sharf, A., Zhao, H.S., Wei, Y., Fan, Q.N., Chen, X.L., Savoye, Y., Tu, C.H., Cohen-Or, D. and Chen, B.Q. (2014), "Build-to-last: strength to weight 3D printed objects", *ACM Transactions on Graphics*, Vol. 33 No. 4, Article No. 97.

Luo, L., Branchan, I., Rusinkiewicz, S. and Matusik, W. (2012), "Chopper: partitioning models into 3d-printable parts", *ACM Transactions on Graphics*, Vol. 31 No. 6, Article No. 129.

Luo, Z., Yang, F., Dong, G., Tang, Y. and Zhao, Y.F. (2016), "Orientation optimization in layer-based additive manufacturing process", *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, *Charlotte, NC, USA*, pp. 1-10.

Mao, Y., Wu, L., Yan, D.M., Guo, J., Chen, C.W. and Chen, B. (2018), "Generating hybrid interior structure for 3D printing", *Computer Aided Geometric Design*, Vol. 62, pp. 63-72.

Mirzendehdel, A.M. and Suresh, K. (2016), "Support structure constrained topology optimization for additive manufacturing", *Computer-Aided Design*, Vol. 81, pp. 1-13.

Muntoni, A., Livesu, M., Scateni, R., Sheffer, A. and Panozzo, D. (2018), "Axis-aligned height-field block decomposition of 3D shapes", *ACM Transactions on Graphics*, Vol. 37 No. 5, Article No. 169.

Paul, R. and Anand, S. (2015), "Optimization of layered manufacturing process for reducing form errors with minimal support structures", *Journal of Manufacturing Systems*, Vol. 36, pp. 231-243.

Shen, Z.H., Dai, N., Li, D.W. and Wu, C.Y. (2016), "Bridge support structure generation for 3D printing", *Materials, Manufacturing Technology*, *Electronics and Information Science*, pp. 141-149.

Song, P., Deng, B., Wang, Z., Dong, Z., Li, W., Fu, C.W. and Liu, L. (2016), "CofiFab: coarse-to-fine fabrication of large 3D objects", *ACM Transactions on Graphics*, Vol. 35 No. 4, Article No. 45.

Strano, G., Hao, L., Everson, R.M. and Evans, K.E. (2013), "A new approach to the design and optimisation of support structures in additive manufacturing", *The International Journal of Advanced Manufacturing Technology*, Vol. 66 Nos. 9/12, pp. 1247-1254.

Thrimurthulu, K.P.P.M., Pandey, P.M. and Reddy, N.V. (2004), "Optimum part deposition orientation in fused deposition modeling", *International Journal of Machine Tools and Manufacture*, Vol. 6, pp. 585-594.

Vaidya, R. and Anand, S. (2016), "Optimum support structure generation for additive manufacturing using unit cell structures and support removal constraint", *Procedia Manufacturing*, Vol. 5, pp. 1043-1059.

Vanek, J., Galicia, J.A.G. and Benes, B. (2015), "Clever support: efficient support structure generation for digital fabrication", *Computer Graphics Forum*, Vol. 33 No. 5, pp. 117-125.

Vanek, J., Galicia, J.A.G., Benes, B., Měch, R., Carr, N., Stava, O. and Miller, G.S. (2016), "Packmerger: a 3D print volume optimizer", *Computer Graphics Forum*, Vol. 33 No. 6, pp. 322-332.

Wang, W., Liu, Y.J., Wu, J., Tian, S.J., Wang, C.C.L., Liu, L. G. and Liu, X.P. (2017), "Support-free hollowing", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 24 No. 10, pp. 2787-2798.

Wang, W., Qian, S., Lin, L., Li, B., Yin, B., Liu, L. and Liu, X. (2017), "Support-free frame structures", *Computers & Graphics*, Vol. 66 No. Supplement C, pp. 154-161.

Wang, W., Wang, T.Y., Yang, Z., Liu, L., Tong, X. and Tong, W. (2013), "Cost-effective printing of 3d objects with skin-frame structures", *ACM Transactions on Graphics*, Vol. 32 No. 6, Article No. 177.

Wang, H., Chen, Y. and Rosen, D.W. (2005), "A hybrid geometric modeling method for large scale conformal cellular structures", *ASME 2005 International Design Engineering Technical Conferences & Computers & Information in Engineering Conference*, *CA*, Vol. 3, pp. 421-427.

Wei, X.R., Geng, G.H. and Zhang, Y.H. (2016), "Steady and low consuming supporting for fused deposition modeling", *Acta Automatica Sinica (in Chinese)*, Vol. 42 No. 1, pp. 98-106.

Wei, X.Z., Li, X.D., Wen, S.S., Zheng, Y. and Tian, Y.B. (2019), "Channel design for 3D models with applications in powder-bed additive manufacturing", *Rapid Prototyping Journal*, available at: https://doi.org/10.1108/RPJ-06-2018-0156.

Wei, X., Qiu, S., Zhu, L., Feng, R., Tian, Y., Xi, J. and Zheng, Y. (2018), "Toward support- free 3D printing: a skeletal approach for partitioning models", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 24 No. 10, pp. 2799-2812.

Wu, J., Wang, C.C.L., Zhang, X. and Westermann, R. (2016), "Self-supporting rhombic infill structures for additive manufacturing", *Computer-Aided Design*, Vol. 80, pp. 32-42.

Yao, M., Chen, Z., Luo, L., Wang, R. and Wang, H. (2015), "Level-set-based partitioning and packing optimization of a printable model", *ACM Transactions on Graphics*, Vol. 34 No. 6, Article No. 214.

Zhang, X., Le, X., Panotopoulou, A., Whiting, E. and Wang, C.C.L. (2015), "Perceptual models of preference in 3d printing direction", *ACM Transactions on Graphics*, Vol. 34 No. 6, Article No. 215.

Zhang, Z., Wu, C., Li, T., Liang, K. and Cao, Y. (2018), "Design of internal branch support structures for selective laser melting", *Rapid Prototyping Journal*, Vol. 24 No. 9.

Zhang, X., Xia, Y., Wang, J., Yang, Z.W., Tu, C.H. and Wang, W.P. (2015), "Medial axis tree – an internal supporting structure for 3d printing", *Computer Aided Geometric Design*, Vol. 35-36 No. C, pp. 149-162.

Zhao, G., Zhou, C. and Das, S. (2015), "Solid mechanics based design and optimization for support structure generation in stereolithography based additive manufacturing", *Proceedings of the ASME Design Engineering Technical Conference & Computers and Information in Engineering Conference*, *Boston*, *MA*, pp. 1-8.

Zhao, J. (2005), "Determination of optimal build orientation based on satisfactory degree theory for RPT", *Ninth International Conference on Computer Aided Design and Computer Graphics*, *Hong Kong*, *China*.

Zhu, L., Feng, R., Li, X., Xi, J. and Wei, X. (2019), "A tree-shaped support structure for additive manufacturing generated by using a hybrid of particle swarm optimization and greedy algorithm", ASME", *Journal of Computing and Information Science in Engineering*, Vol. 19 No. 4, p. 041010.

Further reading

Jadoon, A.K., Wu, C., Liu, Y.J., He, Y. and Wang, C.C.L. (2018), "Interactive partitioning of 3D models into printable parts", *IEEE Computer Graphics and Applications*, Vol. 38 No. 4, pp. 38-53.

Schmidt, R. and Umetani, N. (2014), "Branching support structures for 3D printing", in *ACM SIGGRAPH 2014 Studio*, Article No. 9.

Corresponding author

Xiangzhi Wei can be contacted at: antonwei@sjtu.edu.cn