Procedia Manufacturing

Volume 5, 2016, Pages 1043–1059

44th Proceedings of the North American Manufacturing Research Institution of SME http://www.sme.org/namrc

Optimum Support Structure Generation for Additive Manufacturing using Unit Cell Structures and Support Removal Constraint

Rohan Vaidya and Sam Anand
Department of Mechanical and Materials Engineering
Center for Global Design and Manufacturing
University of Cincinnati, Cincinnati, OH 45221
vaidyark@mail.uc.edu, anands@ucmail.uc.edu

Abstract
Additive Manufacturing (AM) is the process in which a part is built using a layer by layer approach. Due to the inherent nature of the process, support structures are required to support overhanging features while building a part by AM. Support Structures increase the build time and cost of manufacturing and also have an adverse effect on the surface finish of the part. This paper presents a new approach for minimizing support structures using space filling cellular structures in conjunction with Dijkstra's shortest path algorithm to generate optimized support structures. Further, additional support accessibility constraints are applied to the support generation algorithm to ensure the ease of removal of the supports after manufacturing the part. The algorithm is validated by simulating the supports for two test parts while performing FEA analysis to test whether the generated structures are capable of supporting the weight of the part. A third test case is presented to verify the results of the algorithm using the support accessibility constraint.

Keywords: Additive Manufacturing, Support Structures, Optimization, Unit cells, Shortest Path Algorithm, Support Accessibility.

1 Introduction

Additive Manufacturing (AM) has been gaining widespread popularity due to its ability to easily manufacture complex and intricate geometries and hence overcome the limitations of conventional manufacturing processes. The application of this technology has traditionally been in rapid prototyping, but recently, AM has been used in mainstream manufacturing as well. Due to its immense potential, this technology has attracted the attention of many research groups across the world. Several recent papers have focused on improving the part quality by optimizing the process parameters of an AM process

Selection and peer-review under responsibility of the Scientific Programme Committee of NAMRI/SME

doi:10.1016/j.promfg.2016.08.072

(Paul & Anand, 2014) (Paul & Anand, 2011) (Das, et al., 2015). The final quality of a part built by AM depends on many factors such as slice thickness, build orientation, support structures, thermal shrinkage and deformation (Thomas, 2009). Support Structures are one of the important factors to be considered in an AM process. Overhanging features require external support structures and hollow parts require internal support structures (Dutta & Kulkarni, 2000).These support structures increase the build time and cost required to manufacture the part and result in a poor surface finish wherever they are in contact with the part (Thomas, 2009).Thus, it is critical that these redundant structures are optimized such that the build time and cost is minimized and the surface finish quality is improved.

After building the part, it is important to remove all the support structures of the part during post processing. But sometimes, due to the complexity of the part, it is not possible to access and remove all the supports structures. Therefore, it is necessary to take into account the accessibility of support structures while designing them to reduce post processing time and cost of support removal.

This work presents a new approach for building optimum support structures using cellular unit cells while employing Dijkstra's shortest path algorithm (Dijkstra, 1959) for minimizing the support volume. Additional constraints are applied to the algorithm to build optimum supports taking into consideration the ease of removal and post processing of support structures.

This paper is divided into 4 sections: literature review, methodology, results and conclusions. The literature review presents the prior work performed in support minimization. The methodology describes the Dijkstra's shortest path based cellular structure algorithm for support generation. This section also describes the accessibility constraint introduced in the algorithm to facilitate the ease of removal of the support. Support contact area, support volume, and the total sintering area are then calculated in the results section and compared to a solid support for two sample parts. FEA is performed on the supports to validate and check if they support the weight of the feature.

2 Literature Review

Support Structures are an important factor to consider for additively manufactured parts. This section presents prior literature in the area of support structure generation and accessibility and is divided into three parts: the first part presents different support generation methodologies to minimize support structures, the second part explores other approaches towards support minimization while the third part discusses the literature on support structure accessibility.

Chalasani et al. (Chalasani, et al., 1995) presented a computational algorithm for support minimization for fused deposition modeling using a ray casting approach. Huang et al. (Huang, et al., 2009) proposed a sloping wall approach to minimize the support structures in FDM. Using this method, they were able to reduce the support volume by 30%. But, they did not consider reduction of total support contact area with the part. Majhi et al. (Majhi, et al., 1999) implemented geometric algorithms to minimize the support contact area along with the support volume. Strano et al. (Strano, et al., 2013) used cellular structures generated by mathematical equations for support generation. Calignano (Calignano, 2014) demonstrated the use of teethed support structures to reduce the support contact area. They built supports for aluminum and titanium parts and proposed optimum dimensions for the teethed support structure. Hussein et al. (Hussein, et al., 2013) investigated the use of different lattice structures and their volume fractions for support generation. Although, due to thin sections in the lattice structure, manufacturability was an issue.

Apart from various support generation techniques, there has been research focused on finding optimum build orientation for minimizing support structures. Allen and Dutta (Allen & Dutta, 1994) developed a convex hull based approach to find the best orientation from the point of view of reducing support volume. Paul and Anand (Paul & Anand, 2014) implemented a voxel method to calculate the support volume required for a part in a given orientation. They also developed an optimization function for part orientation which reduces the overall support volume. Other researchers have tried to modify

the design process or the manufacturing process to eliminate the use of support structures in AM. Leary et al. (Leary, et al., 2014) developed a method for designing optimal structures which do not require supports. But this approach towards design modification may not be possible for intricate parts. Yang et al. (Yang, et al., 2003) developed a multi-oriented deposition method in FDM in order to reduce the amount of support structures required.

Support structure accessibility has been scarcely studied in literature. Within the co-authors research group, Samant et al. (Samant, 2015) investigated an octree based methodology for analyzing the support structure accessibility for a given part. Other researchers have published similar work in traditional manufacturing. The concept of using visibility maps for NC machines was studied by Chen and Woo (Chen & Woo, 1989). Binary spherical maps were used for the determination of machinability and setup configuration for 5 axis machining by (Kang & Suh, 1997). Kweon and Medeiros (Kweon & D.J., 1998) developed a method in which they used visibility maps for representation of directions which are accessible for inspecting tolerances in a CMM.

3 Methodology

This section explains the methodology used for generating support structures for a given part. Two different unit cells, truncated octahedron and rhombic dodecahedron are used to build supports. The 3D space containing the sample parts is first divided into a voxelized space consisting of these unit cells. The regions of the part requiring support are then identified using the 45^0 criterion (Thomas, 2009). Supports are then built using the Dijkstra's shortest path algorithm (Dijkstra, 1959). The support generated is then compared to traditional solid supports based on total sintered area, total support volume and total contact area with the part. Figure 1 shows the overview of the support generation methodology.

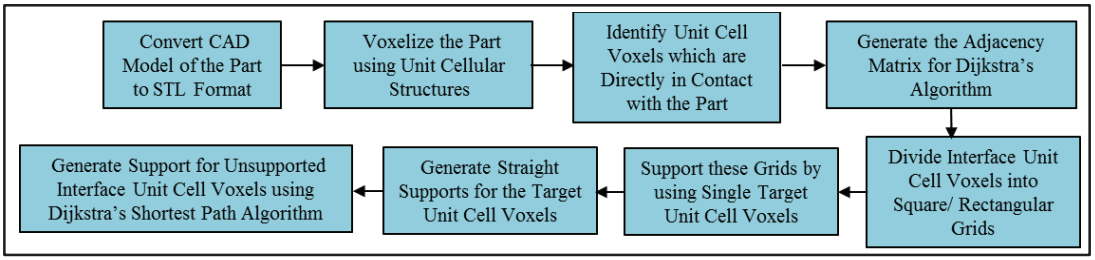
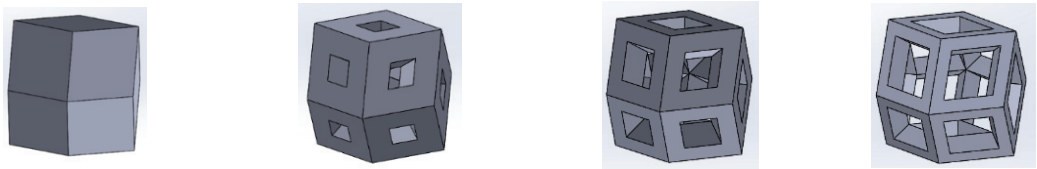


Figure 1. Overview of Methodology

3.1 Unit Cells Voxels Used for Support Generation

Truncated Octahedron (Figure 2a., 2b., 2c., 2d.) and Rhombic Dodecahedron (Figure 2e., 2f., 2g., 2h.) are the two unit cells used in this study for support generation. Figure 2 also shows the volume fractions of each of these cells. These unit cells are chosen as they are space filling and have 14 and 12 planar faces respectively that could serve as planar joint surfaces between unit cells.

a. Solid (Vol. Frac. - 1) **b. Hollow-1 (Vol. Frac. - 0.64)** **c. Hollow-2 (Vol. Frac. - 0.46)** **d. Hollow-3 (Vol. Frac. - 0.23)**



e. Solid (Vol. Frac. - 1) **f. Hollow-1 (Vol. Frac. - 0.72)** **g. Hollow-2 (Vol. Frac. - 0.48)** **h. Hollow-3 (Vol. Frac. - 0.25)**

Figure 2. Truncated Octahedron and Rhombic Dodecahedron unit cell models

The multiple faces of these unit cells also provide more flexibility for support generation in different directions. In this work, support structures are generated using different wall thicknesses of the unit cells. Manufacturability of these kinds of lattice cells was studied by Chua et al. (Chua, et al., 2003) (Chua, et al., 2003). Sudardamji et al. (Sudarmadji, et al., 2010) built an array of truncated octahedron lattices using selective laser sintering. Mechanical Properties of truncated octahedron and the rhombic dodecahedron have also been studied by researchers (Roberts & Garboczi, 2002) (Babaee, et al., 2012).We can conclude from the research cited above that these lattice structures are manufacturable using AM processes and thus can be used for building light weight minimal support structures.

3.2 Division of 3D Part Space into Unit Cell Voxelized Space

Before applying the support generation algorithm, the part space has to be divided into a voxelized space consisting of unit cellular structures. This unit cell voxelized space is then used as a basis for the shortest path algorithm which traverses through this space to build support structures. The 3D space containing the part and the substrate is first divided into an array of cubes voxels using a MATLAB algorithm (Adam, 2010). The cubic voxel space is then converted to unit cell voxel space using the method presented in this section.

To start with, the CAD model of the part is converted into the STL format and serves as input to the voxelization algorithm. This generates a $N_x \times N_y \times N_z$ grid where N_x, N_y, N_z are the number of voxels in each direction. A value of '1' is assigned to the cubic voxels containing the part and a value of '0'is assigned to the remaining cubic voxels in the 3D grid. The algorithm for converting this cube voxel grid to cellular octahedral voxel grid is represented in Figure 3. Each cubic voxel in the grid is checked and a one to one correspondence with the octahedral and the cubic voxel grid is established. After replacing each cube voxel with octahedral voxels, a void is generated for every 8 neighboring octahedral voxel as

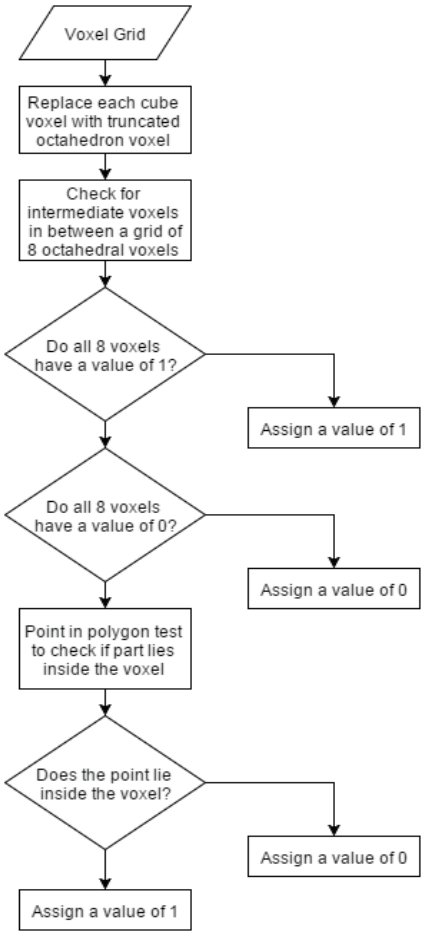


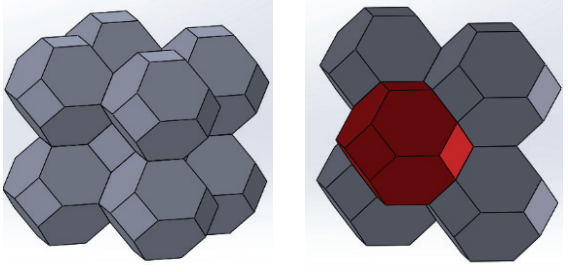
Figure 3. Voxelization using Unit Cellular Structures

show in Figure 4. The truncated octahedral voxel which will fit in this void is termed as 'intermediate octahedral voxel'. To determine whether the intermediate octahedral voxel is a part voxel, the surrounding 8 octahedral voxels are checked and if all of them have a value of '1' assigned, the intermediate octahedral voxel is also assigned a value of '1' indicating that it is a part voxel. If all the surrounding octahedral voxels are void (i.e. have a value of '0'), the intermediate octahedral voxel is assigned a value of '0' indicating a void. For surface octahedral voxels, since neither case is satisfied, surface points of the part are generated and a point in polygon test is performed to determine if the intermediate octahedral voxel is a part voxel.

The result is a matrix of size $N_x \times N_y \times N_z$ which contains the information about the octahedral voxels which directly replaced the cube voxels. Another matrix of size $(N_x - 1) \times (N_y - 1) \times (N_z - 1)$ which contains the information about intermediate octahedral voxels is also created. Thus, the cubic voxelized 3D space is converted in to a unit cell voxelized space using the truncated octahedron. In a similar manner, the cube voxel space can also be converted appropriately to rhombic dodecahedron voxel space.

3.3 Minimum Overhang Angle Criteria

After voxelizing the 3D part space using the cellular voxels, it is necessary to identify the facets of the STL file that require support in a part. This section describes the methodology for identifying such facets by applying the minimum angle criteria. After identifying these facets, all the unit cell voxels in contact with the facets are marked as support facets. Daniel Thomas (Thomas, 2009) proved that the all the overhang surfaces making an angle of 45^0 or more with the horizontal require support structures. Cloots et al. (Cloots, et al., 2013) suggested that all the surfaces making an angle of 35^0 or more with the horizontal require support structures. In this study, 45^0 is chosen as the threshold value. All the facets of the STL file, whose normals make an angle of more than 45^0 with the horizontal axis (normal to build axis) are identified as facets requiring support. These facets are then discretized into points as shown in Figure 5. All the unit cell voxels which contain at least one of these discretized points are marked as support voxels. These voxels which are directly in contact with the part are termed as 'interface unit cell voxels' and are assigned a value of '2' in the 3D voxel grid that was generated in the previous



8 Octahedral voxels with 1 intermediate octahedral voxel

Cut Section showing intermediate octahedral unit cell voxel (in red)

Figure 4. Intermediate octahedral unit cell arrangement

section. The interface unit cell voxels identified in this section are the voxels that need to be eventually supported.

3.4 Adjacency Matrix Generation for Dijkstra's Algorithm

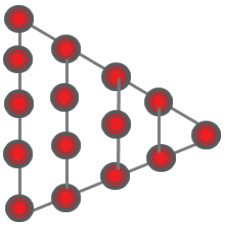


Figure 5. Discretization of Facets

This section describes the development of an adjacency matrix which is used as an input to the support generation algorithm. Dijkstra's (1959) shortest path graph search based algorithm is used to traverse the 3D unit cell voxel space and generate the minimal supporting path of unit cell voxels in between the interface unit cell voxels and the substrate/part unit cell voxel. Dijkstra's algorithm employs a graph matrix which contains the information of the cost of traveling from one unit cell voxel node to its adjacent node. The algorithm solves an optimization problem which minimizes the cost of traveling from initial unit cell voxel to the final unit cell node voxel. Thus, to facilitate this algorithm, an 'Adjacency matrix' is defined for the entire cellular voxel space. The adjacency matrix has a size of $N \times N$ where,

$$N = \left(N_x \times N_y \times N_z\right) + \ (N_x - 1) \times (N_y - 1) \times (N_z - 1) \ \ldots\ldots\ldots (1)$$

The adjacency matrix assigns a cost of traversal from each voxel in the 3D unit cell voxel grid to each adjacent voxel. Each truncated Octahedron will have 14 adjacent unit cell voxels while rhombic dodecahedron will have 12 adjacent unit cell voxels. The cost of traversing to each adjacent unit cell voxel is assigned on the basis of the table shown in Table 1:

Table 1. Table showing Weights of Voxels in Adjacency Matrix

Type of Unit Cell Voxel	Cost
Unit Cell Part Voxel	10 (Large number)
Substrate	1
Void	1
Adjacent Unit Cell Voxel	2

Unit cell part voxels are assigned a high cost (10) to prevent the support from traversing through the unit cell part voxel. Substrate and void voxels are assigned a cost of 1. Adjacent voxels at the same z level are assigned a higher cost to prevent the support from traversing in the horizontal direction to minimize the possibility of overhang. This matrix is used as a basis for Dijkstra's algorithm for building branches of supports connecting the interface unit cell voxels with the substrate or part unit cell voxels below.

3.5 Support Generation

After voxelizing the part space and identifying the interface unit cell voxels, actual supports are generated. The following section describes in detail the methodology for generating these supports.

3.5.1 Interface Unit Cell Voxel Division

This section discusses the division of the interface unit cell voxels into square or rectangular grids. The main objective of this kind of a division is to reduce the number of unit cell voxels for which

a support path has to be generated. The algorithm described below iterates through all the z levels present in the voxel grid.

After identifying the 'interface unit cell voxels' which are directly in contact with the part, they are divided into 3 × 3 grids. All the interface unit cell voxels at a particular z level in the unit cell voxel grid are checked and 3 × 3 grids are detected. Figure 6 presents the process of division of a 3 × 3 grid of unit cell voxels. The initial 3 × 3 grid (Figure 6a) is supported by 4 unit cell voxels below it (Figure 6b) which are then supported by a single voxel below them (Figure 6c). The single unit cell voxel at the end of this structure is termed as 'target unit cell voxel' and is stored in a different matrix. By identifying such a type of 'target unit cell voxel', the number of unit cell voxels for which a support path needs to be generated are reduced from nine to one. These kinds of 3 × 3 grids may not be feasible at every part - support interface. There may be instances where less than nine unit cell voxels are present and the voxels cannot be divided into a grid as large as 3 × 3. If no such 3 × 3 grid is detected, the possibility of a 2 × 2 grid is checked, and a similar structure (Figure 7) is used to support the 2 × 2 grid (Figure 7a) which in turn is supported by a single target unit cell voxel below it (Figure 7b). Figure 8 depicts the terminology of the different types of unit cells in a 3D voxel grid. After dividing the interface unit cell voxels into 3 × 3 or 2 × 2 grids, there may be unit cell voxels which are left unsupported. These remaining unit cell voxels are divided into rectangular grids of 3 × 1 or 1 × 3 and the center voxels of these grids are stored in the target unit cell voxel matrix. There may still be interface unit cell voxels which fail to qualify for any of the three criteria mentioned before. The support for these unit cell voxels is generated in the latter stages of support generation. All the unit cell voxels which are divided into grids and their corresponding supporting unit cell voxels are assigned a value of '3' in the 3D unit cell voxel grid. The target unit cell voxel is assigned a value of '5'. The next section describes the methodology of generating supports for the target unit cell voxels identified in this section.

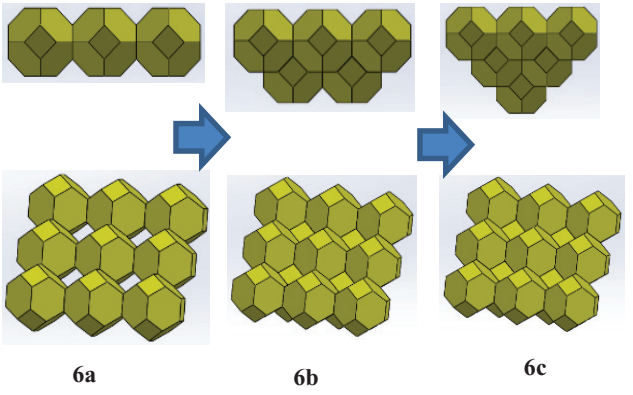


Figure 6. 9-4-1 Support Structure for 3x3 grid

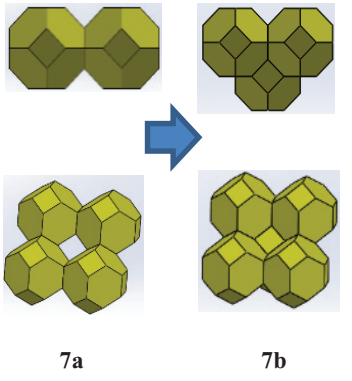


Figure 7. 4-1 Support Structure for 2x2 grids

3.5.2 Support Generation for Target Unit Cell Voxels

The support generation algorithm runs through all the target unit cell voxels (value = 5) and checks for a part or a substrate unit cell voxel directly below it (refer Figure 8). This is achieved by iterating through all the unit cell voxels directly below the target unit cell voxel and then identifying the unit cell voxel which has a value of '1' assigned to it. This is termed as the 'supporting unit cell voxel'. Unit cell voxels between the supporting unit cell voxel and the target unit cell voxel are generated and are assigned a value of '4' in the 3D unit cell voxel grid. Table 2 represents the different types of unit cell voxels present in the grid and their corresponding values. After building supports for all the target unit cell voxels, the unsupported interface unit cell voxels (value = 2) are then detected. These are the

unit cell voxels which could not be divided into any of the grids which were explained in the previous section. The support building methodology for these unit cell voxels is presented in the next section.

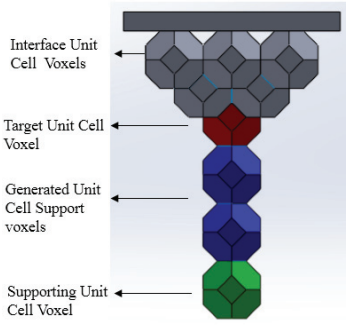


Figure 8. Target Unit Cell Voxel and Supporting Unit Cell Voxel

3.5.3 Support Generation for Unsupported Unit Cell Interface Voxels

All the unsupported interface unit cell voxels (value = 2) are identified and stored in a matrix. For all such interface unit cell voxels, the supporting unit cell voxels which are at a lower z level than the unsupported interface voxel (value = 2) are identified. The supporting unit cell voxels identified in this case consist of all the part/substrate voxels (value = 1) and the support unit cell voxels generated in the previous section (value = 4) (refer Table 2). The supporting unit cell voxel which is closest to the unsupported unit cell interface voxel is chosen for support generation. Figure 9 provides an overview of the process. These two unit cell voxels are then passed to the Dijkstra's algorithm (1959) to build support.

Table 2. Values of unit cells voxels and their interpretation

Type of Unit Cell Voxel	Value
Unit Cell Part Voxel	1
Substrate	1
Void	0
Interface Unit Cell Voxel without support	2
Interface Voxel Unit Cell with support	3
Unit Cell Support Voxels	4
Target Unit Cell Voxel	5

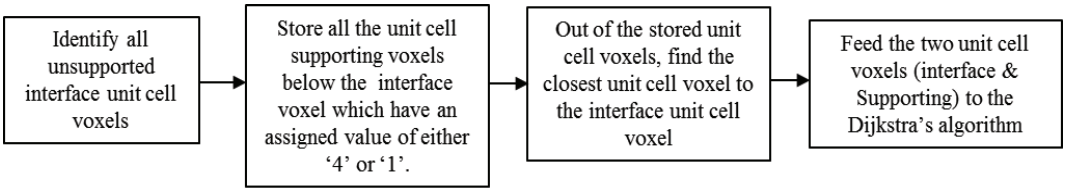


Figure 9. Process Flow for Unsupported Voxel support Generation

Dijkstra's Shortest Path Algorithm:

The Dijkstra's algorithm (1959) is used to generate the shortest path between the two unit cell voxels through the voxel grid. The unsupported interface unit cell voxel (value = 2) serves as the initial starting node and the support unit cell voxel (value = either '1' or '4') serves as the final goal node. The adjacency matrix generated in section 3.4 serves as the graphical input matrix for the algorithm. A path of unit cell voxels between the initial and the final node is generated such that the cost of traversal is minimum. The high cost assigned to the unit cell part voxels prevents the support from traveling through the part. The unit cell voxels in this shortest path are assigned a value of '4' in the 3D voxel grid. An

example of support generation by Dijkstra's for one unsupported unit cell voxel is shown in Figure 10. Figure 11 shows the Dijkstra's algorithm applied for the support generation for a sample part.

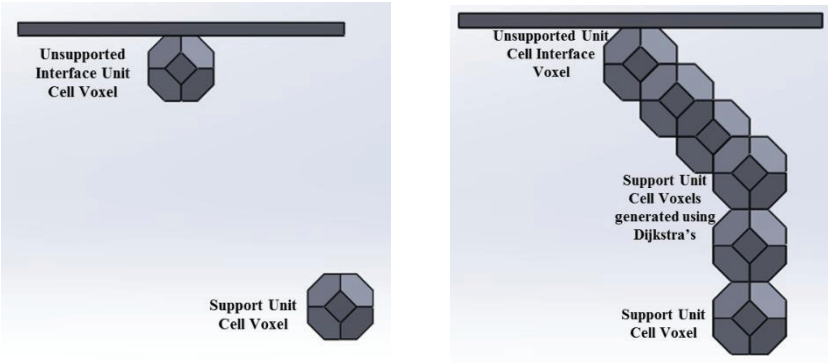


Figure 10. Target Unit Cell Voxel & Supporting Unit Cell Voxel

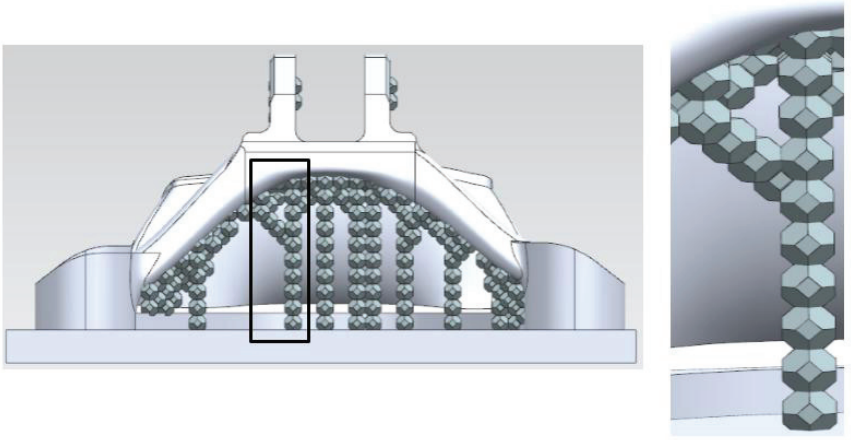


Figure 11. Example of Dijkstra's algorithm applied to a sample part

3.5.4 Building the Support in a CAD Environment

The methodology described above generates a matrix containing the center points of the supporting unit cell voxels as an output. In the next step, the unit cell voxels generated as supports are integrated into the CAD model (NX is used for this study) of the part using an NX open API interface (Siemens PLM, 2015). To start with, the support unit cell voxel center point data is passed to the NX API module in order to generate copies of the voxels within NX CAD (Siemens PLM, 2015). These voxels are subsequently united using the unite function in NX and the voxel support is joined with the original part model in NX CAD.

3.6 Support Structure Generation Using Accessibility Constraint

While creating support voxels, it is prudent to perform an assessment of the ease of removal of support structures during post processing. This assessment is included during the support generation step in the form of an additional accessibility constraint by checking the accessibility of supports from outside the part along 6 orthogonal directions. This emulates the possible directions of tool travel for support removal during post processing. The sample part shown in Figure 12 is used for demonstration of the concept of support generation using the accessibility constraint. The support structures for the part are initially generated using the methodology stated in the prior sections (Figure 13, 14). It can be observed in Figure 14 that the support voxels in the middle section will not be accessible in any of the orthogonal directions.

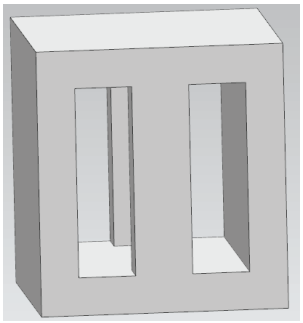


Figure 12. Sample Part

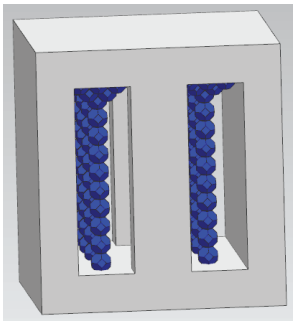


Figure 13. Support for sample part

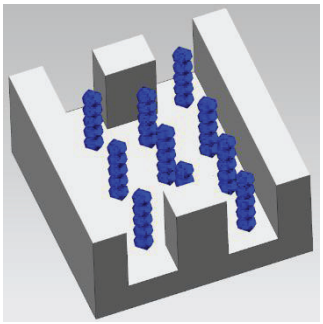


Figure 14. Support Cut section

Support Accessibility Evaluation:

The algorithm for support accessibility calculation is shown in Figure 15. For every unit cell support voxel in the grid, voxel traversal to the end of the 3D unit cell voxel grid (outer boundary of part) in all 6 directions is performed. While traversing to the end of the grid, if a unit cell part voxel (value = 1) is not encountered in any of the 6 directions, the unit cell support voxel is deemed as accessible from the outside. This accessibility check is performed for only the unit cell voxels in the grid which act as supports to the target unit cell voxels.

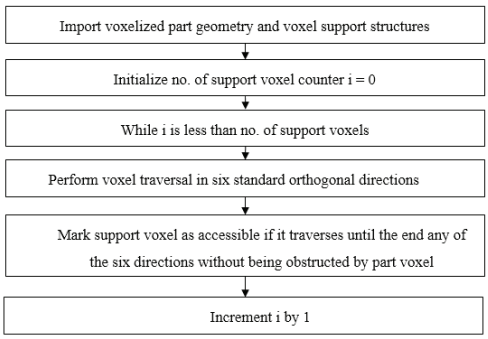


Figure 15. Support Accessibility Calculation

After this step, all the unit cell support voxels which are not accessible are deleted. The adjacency matrix for the Dijkstra's algorithm is updated to include the costs of traversing through the inaccessible unit cell voxels. All the inaccessible voxels are assigned a higher cost to prevent the support from traversing through such voxels. For each of the unsupported target unit cell voxels, the nearest accessible supporting unit cell voxel is detected and support is built between these 2 voxels using Dijkstra's shortest path algorithm.

Overhang Filter

After building the support, any overhangs which may have been generated are identified. Any unit cell voxel in the generated support path is defined as an overhanging voxel if it does not have a supporting unit cell voxel directly below it. For the purpose of this study, the maximum allowable overhang has been set at 5 mm (Thomas, 2009) (Cloots et al., 2013). If such an overhang exists in the build support,

they are supported by applying the support generation algorithm which was explained earlier. This process continues until no such 'overhang' remains in the generated support. To avoid an infinite loop within the algorithm an additional filter is added. This filter compares the x-y plane resultant distance (R) and the z direction (Z) distance from the unsupported unit cell voxel to the nearest accessible unit cell support voxel. The algorithm proceeds to build support only if R<=Z. This prevents the algorithm from building a large overhang support which cannot be further supported by accessible supports and may lead to an infinite loop.

4 Results

This section presents the results for the proposed support generation methodology. Two test cases are presented in this section to validate the support generation algorithm. The results of the test cases are compared to traditional solid supports on the basis of parameters support volume, total sintered area and the contact area of the part. The next section briefly explains the methodology for computing these parameters.

4.1 Calculation of Solid Support Volume, Total Sintered Area and Contact Area of the Part

Total Sintered Area:

In an earlier publication, the co-author (Paul & Anand, 2012) calculated the sintered area for each slice by slicing the CAD part and then adding the area of each slice. In this work, a cubic voxel based method to calculate the total sintered area is applied. After building the support in NX, an STL file is exported which acts an input to the voxelization algorithm in MATLAB (Mathworks Inc, 2015). The cubic voxel size used for this is 0.1 mm, which is equal to the layer thickness by which the part will be manufactured. The voxel size can be adapted to any change in the layer thickness. The total number of voxels having a value of '1' (n) in the voxelized space are determined. The total sintering area is then calculated as follows:

$$Total\ sintered\ area = n \times (area\ of\ voxel\ face) \quad \ldots\ldots\ldots (2)$$

where, $area\ of\ voxel\ face = 0.1 \times 0.1$.

Total Support Volume Calculation for Solid Supports:

The total support volume is computed by applying the algorithm proposed by Paul and Anand (Paul & Anand, 2014). Initially, the STL file of the part along with the substrate and the solid support is voxelized using cubic voxels. Voxel traversal along the z direction (vertical) in the voxel grid is performed and all the trapped empty cubic voxels in between two part voxels or a part and substrate voxel are stored. These trapped voxels closely approximate the solid support for the given part. The total support volume is given by,

$$Total\ support\ volume = Total\ no.of\ trapped\ voxels \times Volume\ of\ each\ voxel \quad \ldots. (3)$$

Support Contact Area:

The support contact area is the summation of the area of the contact face of all the interface unit cell voxels. Thus,

$$Support\ contact\ area = ni \times area\ of\ interface\ voxel\ face \quad \ldots\ldots\ldots (4)$$

4.2 Test Case 1

In this example, an industry bracket (Tomek, 2013) has been chosen for cellular support generation. Figure 16 shows the solid support and the cellular supports for the bracket are shown in Figure 17 and 18. Stress analysis of the unit cell voxel support was performed using ANSYS (ANSYS Inc, 2015) under self-loading. Table 3 shows the Von Mises stress induced in the support structure for truncated octahedron. The material considered for FEA analysis was Ti-6Al-4V as this alloy is most commonly used in Direct Metal Laser Sintering Process. FEA results showed that the structure was safe with a factor of safety of more than 1000 for the truncated octahedron support. The results for the supports using truncated octahedron and rhombic dodecahedron cellular structures are compared with the solid support in Table 4. The results show a substantial decrease in the contact area, support volume, and total sintered area. The results also indicate that the truncated octahedron support will lead to better part finish because of the reduced support contact area while the rhombic dodecahedron will lead to reduced build time and cost.

Support with Hollow Cellular Structures:

Supports were also built using hollow cellular structures of variable wall thickness and volume fractions which were discussed in section 3.1. The results for the truncated octahedron and rhombic dodecahedron are tabulated in the Tables 5 and 6. The tables show that the support structure volume, sintered area, and total contact area are substantially reduced. An alternate method, wherein the only the interface voxels are replaced with hollow structures is also proposed in this study. Supports for the part using this method are shown in Figure 19. Replacing only the interface unit cell voxels with hollow cellular unit cells reduces the support contact area and improves the surface finish while keeping the

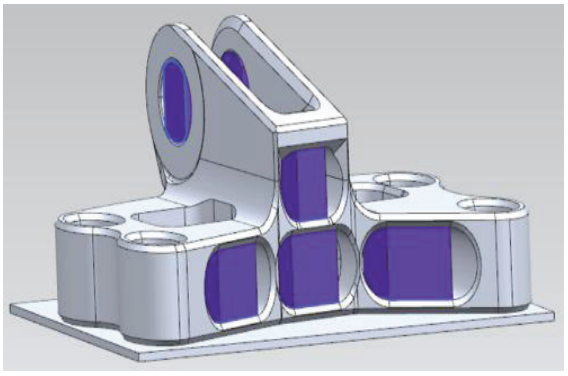


Figure 16. Fully Solid Support

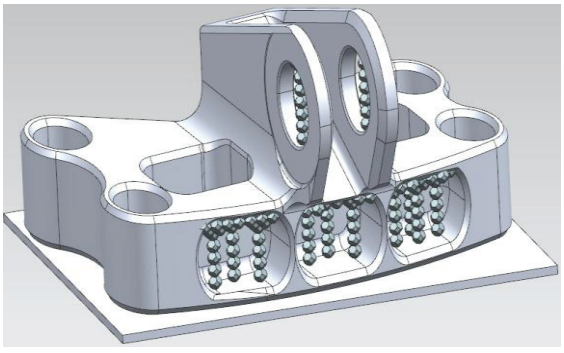


Figure 17. Solid Truncated Octahedron Support

Table 3. FEA Results for the test cases for Truncated Octahedron

Test Case	Maximum Stress (N/m^2)	Factor of Safety
Bracket Test Case 1	$7.2e^5$	1222.2
Turbine Test Case 2	$8.78e^6$	100.23
Accessibility Part Test Case 3	$3.8e^5$	2315.8

Table 4. Comparison Table for Cellular Supports with Solid Unit Cell Voxels

Support Voxel Type	% Reduction Total Support Volume	% Reduction Support Contact Area	% Reduction Total Sintering Area
Solid Truncated Octahedron	65.82	88.61	71.31
Solid Rhombic Dodecahedron	75.26	76.13	77.42

stress levels developed in the structure at a minimum. Comparison of a solid cellular structure support with a support with the hollow interface is tabulated in Table 7.

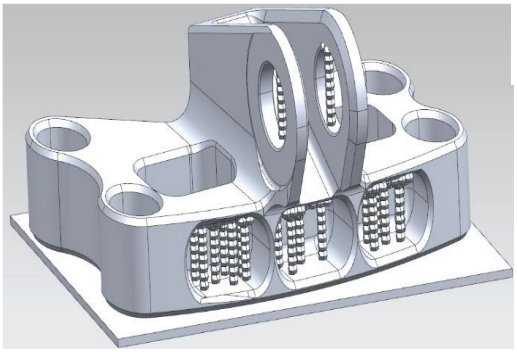


Figure 18. Solid Rhombic Dodecahedron Support

Table 5. Hollow Truncated Octahedron Support Results

Voxel Type	% Reduction - Total Support Volume	% Reduction - Total Support Contact Area	% Reduction - Total Sintering Area	FEA Validation for Support
Solid – Volume Fraction (1)	65.8	88.5	71.31	✓
Hollow 1 – Volume Fraction (0.64)	84.7	90.65	88.24	✓
Hollow 2 – Volume Fraction (0.46)	89.0	90.65	91.35	✓
Hollow 3 – Volume Fraction (0.23)	94.5	93.06	96.11	✓

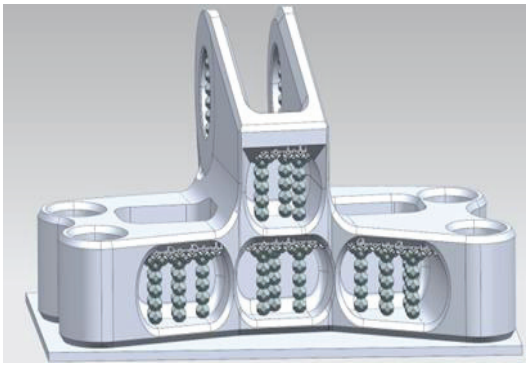


Figure 19. Support with Hollow unit cells at the Interface

Table 6. Hollow Rhombic Dodecahedron Support Results Support Results

Voxel Type	% Reduction - Total Support Volume	% Reduction - Total Support Contact Area	% Reduction - Total Sintering Area	FEA Validation for Support
Solid – Volume Fraction (1)	75.26	76.1	77.42	✓
Hollow 1 – Volume Fraction (0.72)	82.12	79.01	83.76	✓
Hollow 2 – Volume Fraction (0.48)	88.01	82.36	89.48	✓
Hollow 3 - Volume Fraction (0.25)	93.81	86.98	95	✓

4.3 Test Case 2

In this second test case, a turbine part (Safeen, 2015)(Figure 20) is considered. The results for the turbine are presented in Tables 8 and 9. It is observed that there is a substantial reduction in part volume, support contact area and sintering area when compared to solid support structures. FEA results for the

Table 7. Support with Hollow Interface

Voxel Type	% Reduction - Total Support Volume	% Reduction - Total Support Contact Area	% Reduction - Total Sintering Area
Solid	65.82	88.58	71.31
Solid with hollow voxels (volume fraction 0.64) at Interface	76.8	90.65	80.8

Table 8. Rhombic Dodecahedron support Turbine

Voxel Type	% Reduction - Total Support Volume	% Reduction - Total Support Contact Area	% Reduction - Total Sintering Area	FEA Validation for Support
Solid – Volume Fraction (1)	70	82.35	71.3	✓
Hollow 1 – Volume Fraction (0.72)	78.3	84.44	79.3	✓
Hollow 2– Volume Fraction (0.48)	85.45	87	86.6	✓
Hollow 3– Volume Fraction (0.25)	92.5	90.3	93.3	✓

Turbine are shown in Table 3. The stress analysis validates that the structure can support the component under self-loading.

Table 9. Truncated Octahedron support - Turbine

Voxel Type	% Reduction - Total Support Volume	% Reduction - Total Support Contact Area	% Reduction - Total Sintering Area	FEA Validation for Support
Solid – Volume Fraction (1)	40.26	87.5	47.4	✓
Hollow – Volume Fraction (0.64)	73.3	89.8	78.4	✓
Hollow 2 – Volume Fraction (0.46)	80.8	89.8	84.2	✓
Hollow 3 – Volume Fraction (0.23)	90.3	92.44	95	✓

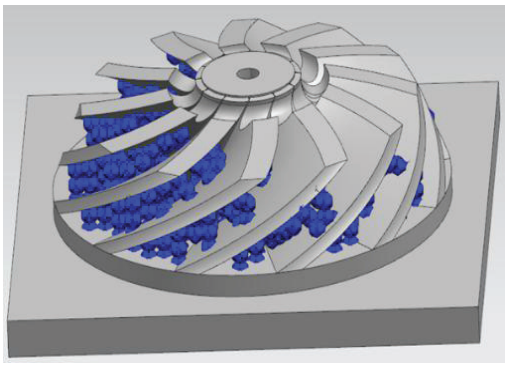


Figure 20. Solid Truncated Octahedron Support for Turbine

4.4 Test Case 3

In this third test case, the results for the support generation with accessibility constraint are presented. The part chosen is the same as the one described in section 3.6. Figure 21 shows the support structures without any accessibility constraint while Figure 22 shows the support structure with an accessibility constraint. The inaccessible voxels in the middle section of Figure 21 are eliminated and accessible support voxels are built. It can be observed in Figure 22 that there is no support structure behind the middle pillar of the part. The orthogonal accessibility for the support structures increases by 11 % from 70 % for Figure 21 to 81% for Figure 22. FEA results for this kind of a support are shown in Table 3 and show that the support structure can support the part under self-loading.

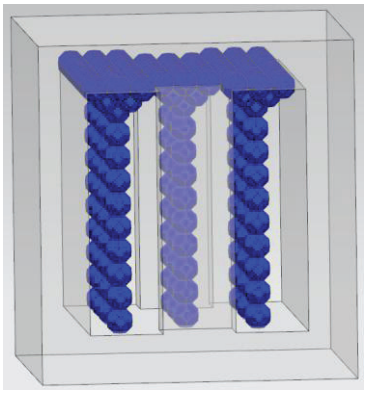


Figure 21. Support Structure without accessibility constraint

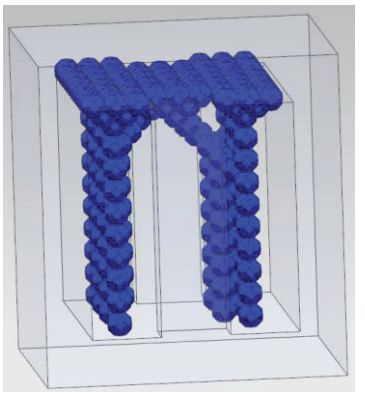


Figure 22. Support Structure with Accessibility constraint

5 Conclusion

This paper presents a new approach for support generation technique for additively manufactured parts using space filling solid and hollow cellular structures to build support for a given part. The method applies the Dijkstra's shortest path algorithm to generate minimal cellular support for the part. Stress analysis performed using ANSYS proves that these structures are capable of handling the load of the part or the feature that they support. It is observed that there is a substantial decrease in support volume, sintered area and support contact area when compared with a fully solid support. The paper also presents a method for building optimal supports while taking into consideration accessibility of supports for post processing.

Possible future work in this area will include the investigation of variable volume/density cellular and lattice structures for support generation. Structures having more than 14 faces (truncated octahedron) or 12 faces (rhombic dodecahedron) may be considered. Space filling polyhedrons such as truncated cuboctahedron or rhombi-cuboctahedron each of which has 26 faces can also be used. These structures will provide greater flexibility in support generation but may also lead to more complex algorithms.

In order to validate the buildability and robustness of the proposed support structures, experimental validation is currently being performed. Based on the experimental results the support generation algorithm can be modified to include solid supports or increase volume of cellular supports to ensure that the supports provide adequate thermal and mechanical constraints to compensate for deformation of the part.

References

Adam, A., 2010. *Matlab Central.* [Online] Available at: http://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation [Accessed 3 June 2015].

Allen, S. & Dutta, D., 1994. *On computation of part orientaion using support struvtures in layered manufactuirng.* Austin, Solid Freeform Fabrication.

ANSYS Inc, 2015. *ANSYS Workbench,* Canonsburg: ANSYS Inc.

Babaee, S. et al., 2012. Mechanical properties of open-cell rhombic dodecahedron cellular structures. *Acta Materialia,* 60(6-7), pp. 2873-2885.

Calignano, F., 2014. Design optimization of supports for overhanging structures in aluminum. *Materials & Design,* Volume 42, pp. 203-213.

Chalasani, K., Jones, L. & Roscoe, L., 1995. *Support Generation for Fused Deposition Modeling.* Austin, Solid Freeform Fabrication.

Chen, L. L. & Woo, T. C., 1989. Computational geometry on the sphere with application to automated machining. *Journal of Mechanical Design ,* 114(2), pp. 288-295.

Chua, C. K., Leong, K. F., Cheah, C. M. & Chua, S. W., 2003. Development of a Tissue Engineering Scaffold Structure Library for Rapid Prototyping. Part 1: Investigation and Classification. *The International Journal of Advanced Manufacturing Technology,* 21(4), pp. 291-301.

Chua, C. K., Leong, K. F., Cheah, C. M. & Chua, S. W., 2003. Development of a Tissue Engineering Scaffold Structure Library for Rapid Prototyping. Part 2: Parametric Library and Assembly Program. *The International Journal of Advanced Manufacturing Technology,* 21(4), pp. 302-312.

Cloots, M., Spierings, A. & Wegener, K., 2013. *SLM, Assessing new support minimizing strategies for the additive manufacturing technology.* Austin, Solid Freeform Fabrication.

Das, P., Chandran, R., Samant, R. & Anand, S., 2015. *Optimum Part Build Orientation in Additive Manufacturing for Minimizing Part Errors and Support Structures.* Charlotte, North American Manufacturing Research Conference.

Dijkstra, E., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik,* 1(1), pp. 269-271.

Dutta, D. & Kulkarni, P., 2000. A review of process planning techniques in layered manufacturing. *Rapid Prototyping Journal,* 6(1), pp. 18-35.

Huang, X. et al., 2009. Sloping wall structure support generation for fused deposition modeling. *The International Journal of Advanced Manufacturing Technology,* 42(11), pp. 1074-1081.

Hussein, A. et al., 2013. Advanced lattice support structures for metal additive manufacturing. *Journal of Materials Processing Technology,* 213(7), pp. 1019-1026.

Kang, J.-K. & Suh, S.-H., 1997. Machinability and Set-up Orientation for Five-axis Numerically. *The International Journal of Advacnce Manufacturing Technology,* 13(5), pp. 311-325.

Kweon, S. & D.J., M., 1998. Part orientations for CMM inspection using dimensioned visibility maps. *Computer-Aided Design,* 30(9), pp. 741-749.

Leary, M. et al., 2014. Optimal topology for additive manufacture: A method for enabling additive manufacture of support-free optimal structures. *Materials & Design,* Volume Materials and Design, pp. 678-690.

Majhi, J. et al., 1999. Minimizing support structures and trapped area in two-dimensional layered manufacturing. *Computational Geometry,* 12(3-4), pp. 241-267.

Mathworks Inc, 2015. *Matlab,* Natick: Mathworks Inc.

Paul, R. & Anand, S., 2011. Optimal Part Orientation in Rapid Manufacturing Process for Achieving Geometric Tolerances. 30(4), pp. 214-222.

Paul, R. & Anand, S., 2012. Process energy analysis and optimization in selective laser sintering. *Journal of Manufacturing Systems,* 31(4), pp. 429-437.

Paul, R. & Anand, S., 2014. Optimization of layered manufacturing process for reducing formerrors with minimal support structures. *Journal of Manufacturing Systems,* Volume 36, pp. 231-243.

Roberts, A. & Garboczi, E., 2002. Elastic properties of model random three-dimensional open-cell solids. *Journal of the Mechanics and Physics of Solids,* 50(1), pp. 33-35.

Safeen, e., 2015. *Grabcad.* [Online] Available at: https://grabcad.com/library/turbine-rotor-21 [Accessed 3 October 2015].

Samant, R., 2015. *Support Structure Accessibility and Removal in Additive Manufacturing using Octree Data Structure,* Cincinnati: University of Cincinnati.

Siemens PLM, 2015. *NX for Design,* Plano: s.n.

Siemens PLM, 2015. *NX Open,* Plano: s.n.

Strano, G., Hao, L., Everson, R. M. & 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,* 66(9), pp. 1247-1254.

Sudarmadji, N. et al., 2010. Selective laser sintering adaptation tools for cost effective fabrication of biomedical prototypes. *Rapid Prototyping Journal,* 16(2), pp. 90-99.

Thomas, D., 2009. *The Development of Design Rules for Selective Laser Melting,* Wales: University of Wales.

Tomek, F., 2013. *GrabCad.* [Online] Available at: https://grabcad.com/library/bracket-ge-ver5-1 [Accessed 16 July 2015].

Yang, Y., Fuh, J., Loh, H. & Wong, Y., 2003. Multi-orientational deposition to minimize support in the layered manufacturing process. *Journal of Manufacturing Systems,* 22(2), pp. 116-129.