Journal of Manufacturing Systems 48 (2018) 131–139

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Part decomposition and 2D batch placement in single-machine additive manufacturing systems

Yosep Oh[a], Chi Zhou[a], Sara Behdad[b,*]

[a] Industrial and Systems Engineering, University at Buffalo, The State University of New York, Buffalo, NY, 14260, United States

[b] Mechanical and Aerospace Engineering, Industrial and Systems Engineering, University at Buffalo, The State University of New York, Buffalo, NY, 14260, United States

ARTICLE INFO

Keywords:
Batch placement
Design for additive manufacturing
Genetic algorithm
Part decomposition
Process planning for additive manufacturing
2D packing

ABSTRACT

To produce a large object within a limited workspace of an Additive Manufacturing (AM) machine, this study proposes a two-phase method: (1) part decomposition to separate a part into several pieces; and (2) 2D batch placement to place the decomposed parts onto multiple batches. In Phase 1, the large object is re-designed into small pieces by a *Binary Space Partitioning (BSP)* with a hyperplane, where parts are decomposed recursively until no parts are oversize the limited size of the workspace. In Phase 2, the decomposed parts are grouped as batches to go through serial build processes using a single AM machine. Within a batch, the decomposed parts are placed based on a 2D packing method in which parts are not placed over other parts to avoid potential surface damage caused by support structure between parts. A *genetic algorithm (GA)* for the 2D batch placement is applied to find near-optimal solutions for build orientations, placement positions, and batch number for each part. As an objective function, the total process time including build time and post-processing time is minimized. This research provides some insights into the relation between part decomposition and 2D batch placement. It shows that minimizing the number of decomposed parts could be more critical than minimizing the size of decomposed parts for reducing the overall process time in serial batch processes.

1. Introduction

Researchers and practitioners have considered *Additive Manufacturing (AM)* as a supplement to the traditional manufacturing (subtractive and formative) [1,2]. However, the AM technology still has several practical limitations such as the finite workspace size of an AM machine [3]. In some cases like houses [4,5] and automobiles [6] that the size of a product is non-scalable and larger than the buildable size, a sufficiently large AM machine might be one solution. However, developing large-scale AM machines does not seem practical, since it requires a huge investment and causes other limitations such as less flexibility of storages and difficulty of transportation. Another solution is to re-design an initial model into assemblies to fit in smaller-scale workspaces. For decades, researchers have worked on methods to decompose an object, *Part Decomposition for AM* [7,8], and methods to pack multiple parts into the limited space, *Part Packing or Placement for AM* [9]. These two issues have been addressed independently and sometimes simultaneously, *Decomposition-and-Packing (DAP) problems for AM* [10].

This paper provides three main research contributions. First, it expands the research boundary of DAP by applying multiple batches rather than a single batch. This is a practical need as an AM machine has a limited workspace and multiple batches are often required to print the whole product. Second, this study presents the relation between the part decomposition and the multiple batches. It discusses that the number of decomposed parts could be more critical than minimizing the size of decomposed parts in terms of reducing the overall process time of serial batch processes. Third, it shows that the 2D packing could be preferred to 3D packing for multiple batches in terms of minimizing the support amount. It validates the claim by Zhang et al. [9] that 2D packing is effective in terms of improving the surface quality by avoiding overlapping parts [9].

In this study, an original model is decomposed into several pieces to fit in the limited space of an AM machine, and then the decomposed parts are placed in multiple batches with 2D packing that is named as 2D batch placement. Fig. 1 illustrates the overall procedure of the proposed approach for a rabbit model. First, an initial solid model goes through the part decomposition algorithm and is decomposed into seven pieces. Then, the pieces enter into the *genetic algorithm (GA)* for the 2D batch placement. In this example, the decomposed parts are placed over three batches as shown in Fig. 1-(b).

The proposed method in this study is suitable for large-size and non-scalable objects since it includes a part decomposition method to fit in

* Corresponding author.
E-mail addresses: yosepoh@buffalo.edu (Y. Oh), chizhou@buffalo.edu (C. Zhou), sarabehd@buffalo.edu (S. Behdad).

https://doi.org/10.1016/j.jmsy.2018.07.006
Received 26 November 2017; Received in revised form 1 July 2018; Accepted 23 July 2018

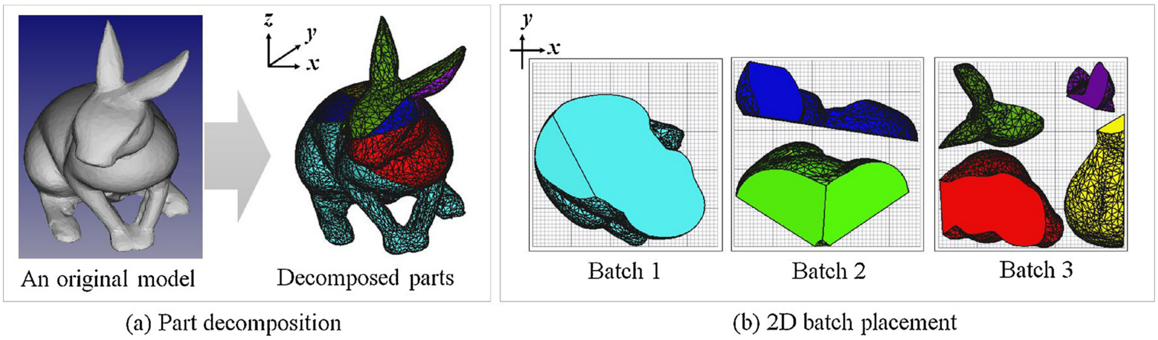


Fig. 1. A Two-Phase approach: (a) part decomposition and (b) 2D batch placement.

the limited workspace. In addition, since the batch placement is based on 2D packing, this study can be effective for the AM technologies with support structure issues, such as *Stereolithography (SLA)* and *Fused Deposition Modeling (FDM)* [11].

The rest of this paper is organized as follows. Section 2 reviews the related literature on part decomposition and packing issues for AM. The proposed methods and algorithms are introduced in Section 3. A numerical example is described in Section 4. Finally, Section 5 concludes the study and suggests the future research directions.

2. Literature review

To clearly categorize the literature, we define two groups of multiple parts based on component relations: *independent parts* and *dependent parts*. The independent parts are literally not related to each other for assembly. For instance, the relation of a rabbit model and a cat model. On the other hand, the dependent parts are sub-assemblies needed to complete a final product such as table legs and a table board. Section 2.1 reviews AM studies on how dependent parts are generated from an initial object and how they are packed in a limited workspace. The literature for packing issue of independent parts is covered in Section 2.2.

2.1. Part decomposition and packing of dependent parts (assemblies)

2.1.1. Part decomposition methods to fit an object into a limited work size

Part decomposition has been studied for several purposes: to fit a large object into the limited workspace of an AM machine [12]; to minimize process time [13]; to remove support structure [14]; to improve surface quality [15]; to have interchangeability among parts [16]; and for artistic purpose [17]. The current paper focuses on the first purpose, known as *printability*.

To fit an object into the limited workspace of an AM machine, several decomposition methods have been developed. For example, Chan and Tan [18] proposed a decomposition method [18], in which a solid model is cut with split tool surfaces, a hyperplane or a curved surface, to fit in a rectangular or cylindrical chamber. Medellin et al. (2006) suggested a decomposition algorithm to generate octants [19]. They developed a recursive decomposition process that divides a cube into two spaces for the three axes (x, y, and z) by a hyperplane, and finally, an octree structure is generated in which each parent node has eight child nodes. The octants are cubes of leaf nodes in the octree structure. Hao et al. [20] presented a curvature-based partitioning method to fit a large complex model to the buildable space [20]. In their algorithm, the best-fit loop is selected and then cut with a hyperplane. Luo et al. [7] suggested a framework for decomposing a large solid 3D model into smaller pieces to fit into the working volume of the 3D printer, known as *the Chopper*. They adopted a BSP and cut an initial model with a hyperplane [7]. A binary tree represents decomposition processes and the leaf nodes are final decomposed parts. In their algorithm, cutting is recursively conducted until the part volume is less than a certain threshold parameter. However, the focus of the above-mentioned studies was only on the part decomposition not packing issues.

2.1.2. Considering both decomposition and packing issues

Some studies have addressed both decomposition and packing topics known as *PackMerger* [21], *decompose-and-pack* [10], *partitioning and packing* [22] or *split-and-pack* [23]. Vanek et al. [21] were the first group who expanded the object decomposition issue to packing problems for AM, which affected later studies such as Chen et al. [10]. In PackMerger, an initial model is decomposed into several parts using a bottom-up approach in which several starting seeds are getting merged with adjacent cells. Then, build orientations and packing of resulting parts are optimized sequentially [21]. Later, other studies have optimized both orientation and packing issues simultaneously [10,22,23]. For example, Chen et al. [10] adopted a pyramidal shape [24] to solve both part decomposition and 3D packing issues, known as DAP problems [10]. They proposed a global optimization algorithm for solving DAP problems, named as Dapper. The Dapper algorithm aims to minimize support material, build time and assembly cost, and considers several constraints including the bounding container, and the assembly thresholds such as cut area and part thickness. Yao et al. [22] developed the decomposition and packing system based on level-set methods [22]. The level-set method is used to refine segmentation boundary between parts with free forms such as curved seams. The authors showed a locking issue that prevents decomposed parts from being assembled back into the original shape. However, the above-mentioned decomposition and packing studies only consider 3D packing assuming the full placement of all parts. This still leaves the subset placement issue that all parts cannot be placed on one AM machine.

2.2. Packing optimization of independent parts

The packing problem deals with how to optimally place independent multiple parts (with same or different shapes) into a limited build space (3D packing) or onto the build tray (2D packing) with respect to user-defined objectives [9].

2.2.1. 3D packing

To name several studies that have been focused on 3D packing, Ikonen et al. [25] developed a GA for packing 3D non-convex parts with cavities and holes into the build cylinder of a *Selective Laser Sintering (SLS)* machine [25]. Parts are randomly selected from a specified group to form a subset of parts in which each part had 24 pre-defined alternative orientations (45 degrees of increment in three directions). The parts are placed one by one with finite relative positions constrained by a pre-set including five attachment points for each part. Hur et al. [26] proposed a part placement optimization strategy for SLS to maximize the utilization of workspace and reduce the total build time [26]. Before

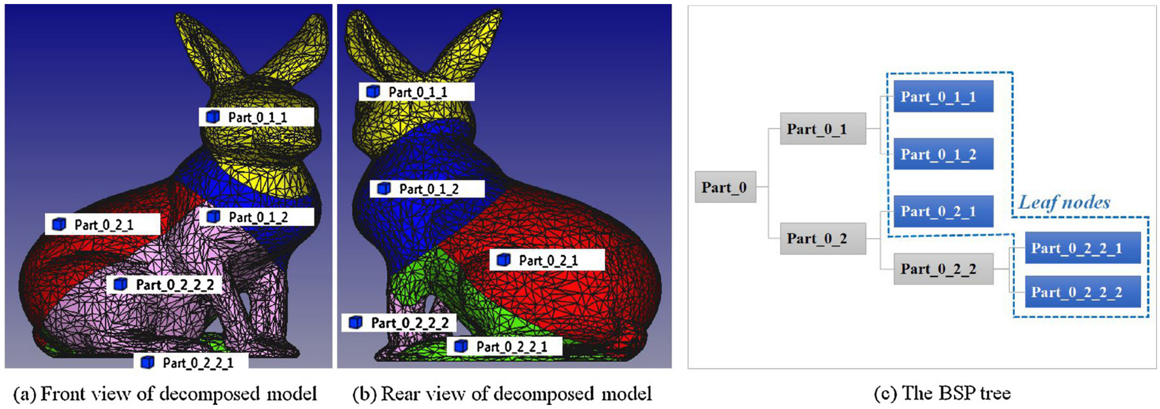


Fig. 2. The part decomposition of a rabbit model and its binary space partitioning (BSP) tree.

packing, the build orientations of parts are optimized by considering the height of parts and surface quality. Then, a modified *Bottom-Left (BL)* approach implemented by GA is used to search an optimal packing solution within a cylindrical build chamber. However, the BL packing requires order information as it is a serial packing and reduces the original solution space. Zhang et al. [27] developed a layout optimization strategy implemented with a *simulated annealing (SA)* for *solid ground curing (SGC)* process [27]. In their method, parts are represented by bounding box and rotated under six pre-set orientations. As parallel packing method, an overlap between parts is considered and could be removed by applying a compensating algorithm. Gogate and Pande [28] developed a 3D layout planning system for optimizing the multiple parts placement in AM [28]. In their study, build orientations of parts are optimized according to user-concerned criteria. For each part, a finite set of acceptable orientations are generated. A BL approach is used to solve the 3D packing problem. Parts are represented by voxel models and could be rotated around the build axis with an increment of 45 degrees. Lutters et al. [29] suggested an algorithm for 3D packing based on a non-deterministic approach in which high frequency vibrating motions called *Brazil Nut Effect* is used [29]. Wu et al. [30] proposed an improved BL method to solve the 3D packing problem in AM [30]. In their method, orientations are pre-set and fixed during packing. Multi-objective functions are used in optimization to generate a Pareto-frontier for user's further decision-making.

2.2.2. 2D packing

While most of the previous studies have focused on 3D packing to save wasted space [28,30], Canellidis et al. [31] proposed a 2D packing method for the SLA process. [31]. Before the 2D packing, the build orientation for each part is determined individually by considering one or several criteria. The packing rule is an improved BL packing method, called *Left-Border-Down-Border (LB-DB)*. This study is improved by Canellidis et al. [32]. They adopted *No-Fit-Polygon (NFP)* as an additional placing rule based on the former LB-DB to avoid overlap of projections. Zhang et al. [9] pointed out the surface damage issue caused by support structures in 3D packing [9]. They suggested the 2D placement optimization to solve the problem. The method goes through two optimization processes sequentially: AM feature-based orientation optimization, which decides on each part's build orientation to guarantee the production quality, and parallel packing optimization, which aims to maximize the compactness of placements by using the projection profiles of parts so as to decrease the total build time and cost.

The above-mentioned studies for 3D packing and 2D packing only considered independent parts, meaning that the part decomposition has not been included in their research scope. In this article, we have developed a method that integrates both part decomposition and packing problem.

3. The two-phase approach

This section describes the two phases of the proposed approach as highlighted in Fig. 1. In Phase 1, an initial object is decomposed using a *Binary Space Partitioning (BSP)* with a hyperplane. The part decomposition is repeated until no parts are oversize the limited size of a workspace. In Phase 2, a GA is employed to find a near optimal solution for 2D batch placement. In the GA, parent chromosomes are selected based on the tournament approach. Additionally, decomposed parts are placed in serial order with the *Left-Bottom (LB)* approach for 2D packing. The major assumptions in this study are as follows:

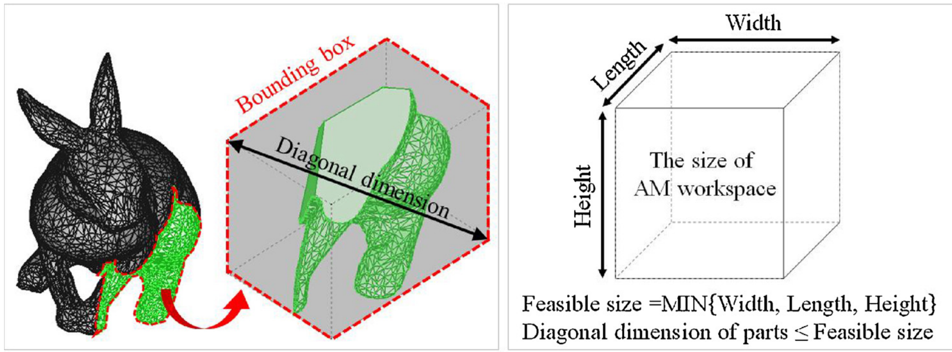
- Parts are built by a single AM machine;
- The buildable space of workspace (build chamber) is rectangular; and
- The object is larger than the buildable space, meaning the diagonal of the bounding box of the object is larger than the width, length, or height of the buildable space.

3.1. Phase 1: part decomposition

In this paper, the major motivation for decomposing an object is its *printability* within the limited size of the workspace. The smaller is the size of workspace, the lager is the number of decomposed parts. Therefore, the size of the workspace is a key factor in determining the number of assemblies.

The use of a BSP with a hyperplane for the part decomposition results in the higher efficiency by giving up the flexibility of arbitrary cuts [7,33]. In this study, the cutting direction with a hyperplane is randomly determined based on the center of mass. Since, in the BSP method, a part is decomposed into two pieces, the number of parts increases one by one. The data structure of BSP is represented by a binary tree. In the binary tree, each parent node includes two child nodes indicating the resulting decomposed parts. The root node of the tree is an initial object and the leaf nodes are the final pieces after the completion of part decomposition. Fig. 2 shows an example of the part decomposition for a rabbit model, *Part_0*. Five parts are generated as shown in Fig. 2-(a) and (b) corresponding to the leaf nodes in Fig. 2-(c). In this example, part decomposition is conducted four times.

Part decomposition is repeated until the diagonal dimension of all parts is smaller than the feasible size. As shown in Fig. 3, the diagonal dimension is the diagonal size of the bounding box of a part and the feasible size is the minimum dimension of width, length, and height of an AM workspace. When parts are placed in the workspace, they are rotated according to x-, y-, and z-axis. Therefore, the diagonal dimension, theoretically the largest size, should be smaller than the feasible size to fit in the workspace.



(a) The bounding box of *Part_0_2_2_1* (b) The feasible size

Fig. 3. (a) The bounding box of a part and (b) the feasible size.

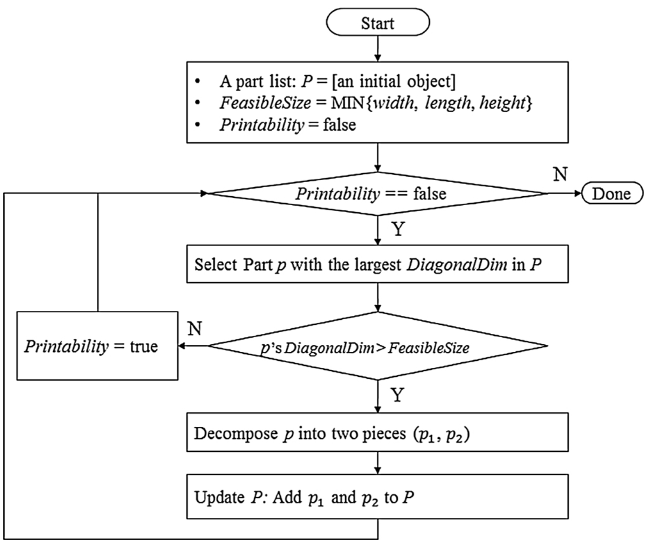


Fig. 4. The flowchart of part decomposition.

Fig. 4 represents the flowchart of the proposed part decomposition method. First, variables and setting parameters are initialized. The part list, *P*, includes an initial object as an input part. *FeasibleSize* is the minimum among the width, length, and height of an AM workspace. As a stopping condition of the algorithm, *Printability* is set to false. When the diagonal dimension, *DiagonalDim*, of all parts is smaller than *FeasibleSize*, *Printability* becomes true and the loop is terminated. When *P* includes multiple parts, a part, *p*, with the largest *DiagonalDim* is selected and then is decomposed into two pieces, p_1 and p_2. The degrees of a hyperplane for decomposition are determined arbitrarily. After part decomposition, *P* is updated by adding two new parts.

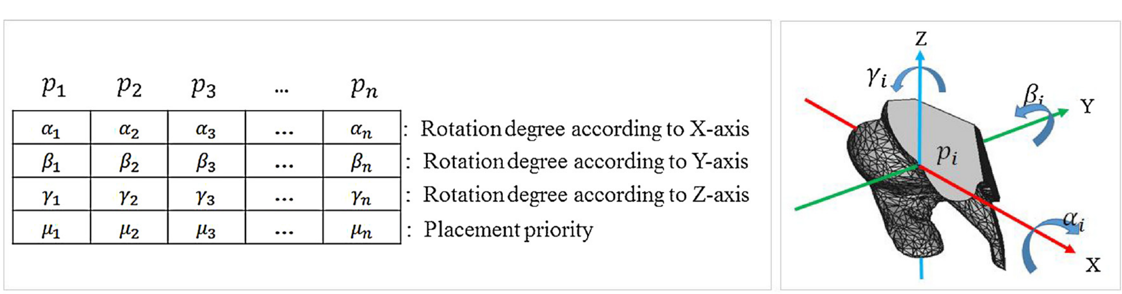
3.2. Phase 2: genetic algorithm for 2D batch placement

In Phase 1, the geometry and number of parts are decided by part decomposition. The second phase is to determine how to place the decomposed parts over multiple batches. As discussed before, this study considers the 2D packing to avoid one part placed over other parts. In the placement process, each part is represented by a bounding box to simplify the computation procedure. The bounding boxes of parts are placed one by one, as serial placement orders. To determine the placement order for each part, placement priority is assigned to a corresponding part. As such, for batch placement, a part with the higher priority goes first. If there is no space to put the bounding box of a part on a certain batch, it is placed in a new batch. The bounding boxes of parts are placed to the left and bottom corner in the X–Y plane, the LB approach for placement orders. In this approach, the x-position of a bounding box is determined first before the y-position.

In the proposed GA, a chromosome consists of genes that correspond to parts. As such, the length of genes in a chromosome is identical to the number of decomposed parts. Fig. 5-(a) details the structure of a tetraploid chromosome including *n* parts where each part (p_i) has four factors influencing batch placement: rotation degrees according to X- (α_i), Y- (β_i), and Z-axis (γ_i); and placement priority (μ_i). With the center of mass, a part is rotated according to the three axes as shown in Fig. 5-(b). This changes the size of a bounding box of a part.

Fig. 6 shows an example of the 2D batch placement. In this example, there are six parts with fixed rotation degrees. Placement orders are determined by placement priority. Therefore, the bounding box of Part p_2 with the highest priority, 6, is placed first in the left-bottom corner of Batch 1. Then, p_3 and p_5 are sequentially placed after p_2. When p_1 is placed, there is no more space in Batch 1. Therefore, p_1 is placed in a new batch, Batch 2. This is the same as the last part, p_4. It is placed in a new batch, Batch 3, since p_4 is too big to place in Batch 2.

Fig. 7 represents the flowchart of the GA for batch placement. First, a population (*U*), a list of chromosomes, is initialized. For the initialization, rotation degrees and placement priority in a chromosome



(a) A tetraploid chromosome (b) Part rotations for each axis

Fig. 5. (a) The structure of a chromosome and (b) part rotation for the three axes (x, y, and z).

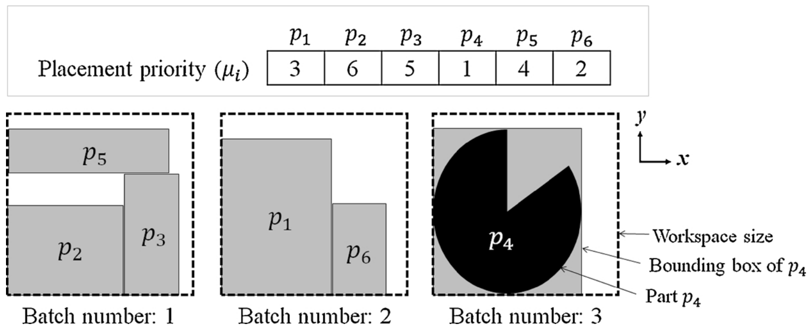


Fig. 6. An example of the 2D batch placement.

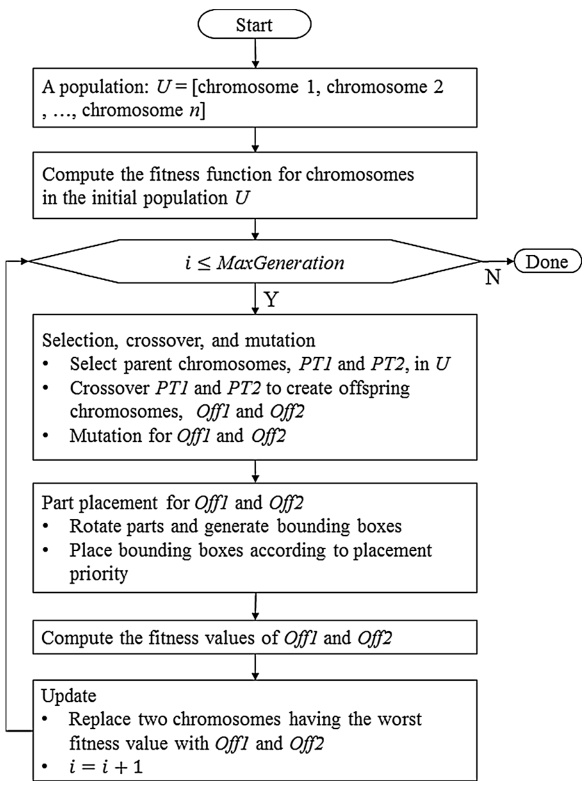


Fig. 7. A flowchart of the genetic algorithm (GA) for batch placement.

are randomly assigned for each part. Then, each chromosome is evaluated based on the fitness function shown in Eq. (1). The loop in the algorithm is conducted until the number of generations, i, reaches to *MaxGeneration*. In the loop, two parent chromosomes, *PT1* and *PT2*, are selected in U where the selection process is tournament-based. Two selected chromosomes are going through the crossover process to create two offspring chromosomes, *Off1* and *Off2*. Herein, crossover for rotation degrees and placement priority is different. In the crossover for rotation degrees, two chromosomes interchange genes that are below the randomly selected position, which is a one-point crossover. On the other hand, for placement priority, *partially matched crossover (PMX)* is used since all parts should have different priorities. Then, the mutation process is conducted. Mutation genes are randomly chosen in a chromosome. The rotation degrees of chosen genes are randomly assigned and the placement priority of two chosen genes are exchanged. In the part placement, parts are rotated according to gene data. Then, the corresponding bounding boxes are generated. Next, the bounding boxes are placed according to placement priority. Based on the batch placement of all parts, the fitness functions of *Off1* and *Off2* are calculated. Finally, the two chromosomes with the worst fitness value are replaced with the offspring chromosomes.

Eq. (1) represents the fitness function that calculates the total process time. The objective is to minimize the fitness value with the aim of finding the minimum total process time, T, including both the total build time, T_{bld}, and the total post-processing time, T_{post}.

$$Min:\ T = T_{\text{bld}} + T_{\text{post}} \tag{1}$$

where T_{bld} and T_{post} denote the total build time and the total post-processing time when all parts in List P are placed over several batches. As shown in Fig. 8, T_{bld} is the sum of the build time for all batches and T_{post} is the sum of assembly time and all setup time.

It is challenging to calculate an accurate build time since a variety of process parameters should be considered, depending on the type of AM processes [34] and mechanical properties such as time to heat material [35]. Moreover, considering process parameters increases computation time and cost. In particular, it is critical for the GA since the build time is calculated for each generation. Therefore, this paper simplifies build time model based on part geometry including volume and height. The model for SLA is modified from [36] and [37] and roughly estimates build time as follows:

$$T_{\text{bld}} = \sum_{i=1}^{n} \{t_{\text{scan}}^{(i)} + t_{\text{trn}}^{(i)}\} \tag{2}$$

where $t_{\text{scan}}^{(i)}$, $t_{\text{trn}}^{(i)}$, and n denote scan time to draw parts or support, transition time between layers, and the number of batches, respectively.

$t_{\text{scan}}^{(i)}$ and $t_{\text{trn}}^{(i)}$ are calculated as follows:

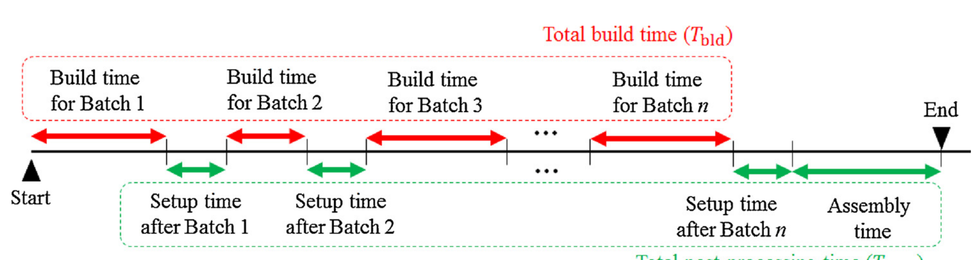


Fig. 8. The total build time and the total post-processing time for batch processes.

$$t_{\text{scan}}^{(i)} = \frac{V_p^{(i)}}{lD_pS} + \frac{V_s^{(i)}}{lD_sS} \tag{3}$$

$$t_{\text{trn}}^{(i)} = t_{\text{rec}}L^{(i)} \tag{4}$$

where $V_p^{(i)}$, $V_s^{(i)}$, D_p, D_s, S, and l are the volume of parts in the i-th batch, the volume of the support structure in the i-th batch, scan distance of parts, hatch distance of support structure, scan speed, and the layer thickness. Herein, support amount is indirectly estimated by summarizing the volume made by the downward (overhanging) facets generating the supports [36]. Additionally, t_{rec} and $L^{(i)}$ denote recoat time for a layer and the number of layers in the i-th batch. $L^{(i)}$ is computed by dividing the height of highest part in the batch to the layer thickness, l.

As shown in Eq. (5), T_{post} consists of two parts: setup time and assembly time.

$$T_{\text{post}} = t_{\text{set}}n + \tau(|P|-1) \tag{5}$$

where t_{set}, τ, and $|P|$ denote the setup time after a batch, the unit assembly time, and the total number of assemblies. The assembly time is estimated by multiplying τ to the number of connections ($|P|-1$). τ means the time to handle parts and use glue for connecting the parts. In this study, we assume that all parts are simply connected with glue. In practice, t_{set} could include various factors such as time to detach parts from the build tray, support removal time, and machine setup time. However, for simplicity, it is roughly set as a consistent parameter. Note that τ for all connections are also consistent.

4. Case study

For case studies, the rabbit model shown in Fig. 1-(a) is used as an initial model. The size of the model is 140.34 × 137.25 × 156.18 (mm) with the diagonal size of 250.85 (mm). The part decomposition and batch placement algorithms are coded using Python 2.7.8 and Macro files in the CAD platform of FreeCAD 0.16 [38].

4.1. The impact of feasible size on the number and size of decomposed parts

The feasible size of the workspace affects the number of decomposed parts. By limiting the diagonal size of parts, a smaller feasible size results in a higher number of parts decomposed. To demonstrate this point, part decomposition is conducted 20 times for different feasible sizes ranging from 110 to 250 (mm). Fig. 9-(a) and (b) show the average number of decomposed parts and the average diagonal size of decomposed parts based on the feasible size. While the average diagonal size of parts linearly decreases by getting confined the part size, the number of parts exponentially increases by decreasing the feasible size. This means that the too small size of workspace leads to a significant number of decomposed parts.

In practice, the maximum size of the workspace, the feasible size, is fixed when only one AM machine is used. However, although the feasible size is fixed, the result of part decomposition can vary in terms of the size and number of parts as shown in Fig. 9-(c). Therefore, the trade-off between the number and size of decomposed parts should be considered in batch placement to minimize the total process time.

4.2. The relation between part decomposition and 2D batch placement

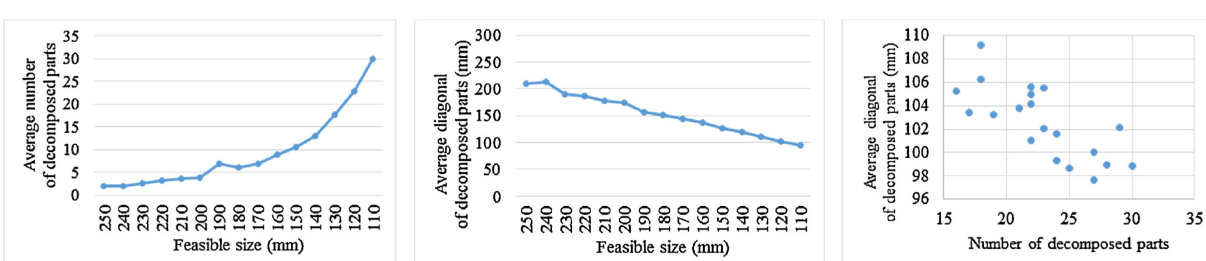
Section 4.1 represents that, even if the same initial part is used, the random-based part decomposition generates different results in terms of the number and size of parts. As shown in Fig. 9-(c), running the part decomposition algorithm for 20 times results in 20 different assembly cases. The result of part decomposition for each case is used as an input for the 2D batch placement algorithm to identify the relation between part decomposition and 2D batch placement. The results of the 2D batch placement for each case are shown in Table 2.

In this case study, we consider the size of the workspace as 120 × 120 × 120 (mm), so the feasible size is 120 (mm). To run the GA for the batch placement, the population size is set to 10 and the maximum number of generations, *MaxGeneration*, is set to 300. In the tournament-based selection, the selection pressure to choose dominant parents is 0.9. To calculate the fitness function based on Eq. (1), the process parameters are given in Table 1.

In Table 2, the best and worst cases in terms of minimizing the total process time are Cases 8 and 4. In this experiment, the best case with 71,038 (s) decreases the total process time by 26.96% compared with the worst case, 97,252 (s). This means that the results of the 2D batch placement significantly vary depending on part decomposition even if the same initial part is used.

The trade-off between the number and size of decomposed parts should be considered to minimize the total process time. If an object is decomposed into many pieces, the size of pieces would be small mostly. This has two possibilities for the total process time. First, the total process time might increase since many parts cause many batches which results in more build processes. Second, the total process time might decrease since the small size of parts creates small build height in 2D packing, which results in short build time.

Figs. 10 and 11 represent that the first possibility is more likely to happen. The data in Table 2 are plotted in Figs. 10 and 11 to show the relations between input factors (the number and size of decomposed parts) and the total process time of the 2D batch placement for the 20 cases. Fig. 10-(a) presents that the number of decomposed parts and the number of batches are positively correlated with the correlation coefficient of 0.85. In addition, Fig. 10-(b) shows that the number of batches and the total process time are also considerably related with the correlation coefficient of 0.86. This means that the high number of decomposed parts causes the large number of batches, which results in a long total process time. On the contrary, Fig. 11-(a) and (b) present that the size of parts is not considerably related to the total process time, which is against the second possibility. Therefore, to minimize the total



(a) The average number of parts based on the feasible size

(b) The average diagonal of parts based on the feasible size

(c) The number and size of decomposed parts (feasible size = 120 mm)

Fig. 9. The average number of parts and the average diagonal of parts according to the feasible size.

Table 1
Summary of parameters used in the case study [3,36].

Parameter	Value	Parameter	Value
Layer thickness (l)	0.05 mm	Scan distance of part (D_p)	0.1 mm
Scan speed (S)	10,000 mm/s	Hatch distance of support (D_s)	0.7 mm
Recoat time for a layer (t_{rec})	6 s	Setup time (t_{set})	1260 s
Assembly time (τ)	30 s		

process time, it is preferred to have a fewer number of decomposed parts even if the size of each part is large.

4.3. Comparison of 2D and 3D batch placement

To compare with the result of 2D batch placement, the GA for the 3D batch placement is conducted for the decomposed parts resulted from Section 4.1. All experiment conditions and process parameters are the same as Section 4.2. For 3D packing, the *Left-Bottom/Down-Bottom (LBDB)* approach is used with serial placement [32]. In Table 2, the 2D and 3D batch placement methods are compared in terms of four factors: the number of batches, the average height of batches, the support amount and the total process time. In this study, batch height is the maximum height of parts within the build tray of a batch and the average height of batches is the average of batch height for all batches.

When the average values of the 20 cases are compared, the total process time of 2D and 3D batch placement is 83,215 (s) and 82,481 (s) respectively. This means that the results of two methods are not that much different in terms of the total process time. The total process time is affected by three other factors. For the 2D batch placement, the number of batches, 7.50, is larger than the value, 5.80, of 3D batch placement. The larger number of batches has a negative influence on minimizing the total process time since more build and setup processes are required. However, the smaller values for the average height of batches, 65.06 (mm), and support amount, 441,746 (mm3), have a positive effect on minimizing the build time for each batch and compensate for the loss of a large number of batches. Herein, the point is the gap between two methods in terms of the support amount. The 3D batch placement requires 61.45% more support structure than the 2D batch placement since parts are stacked over others and the space between parts is filled with support. It has negative influences in terms of material cost and product quality since more material for support structure is needed and more risk is involved to damage part surface for removing support. These are major reasons to apply 2D batch placement instead of 3D batch placement [9].

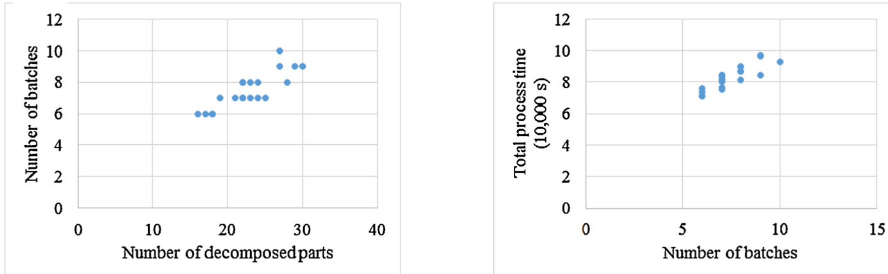
5. Conclusion and future research directions

The objective of this study is to answer the following two main research questions: (1) how to cut a large object into smaller pieces to fit in the limited workspace of an AM machine; and (2) how to place and group the decomposed parts into several batches to minimize the total process time. To answer these questions, this study proposes a two-phase approach including a part decomposition method and a 2D batch placement algorithm. In Phase 1, an object is decomposed into smaller pieces using a BSP method with a hyperplane. Then, the decomposed parts are placed over several 2D batches using the GA. The result of case studies presents two major findings. First, minimizing the number of decomposed parts is more critical than minimizing the size of decomposed parts in terms of reducing overall process time of serial batch processes. Second, 2D batch placement could be preferred than 3D batch placement in terms of minimizing the support amount.

This study can be extended in several ways. First, generating assemblies by part decomposition could be controversial in *Design for AM (DfAM)*, since it may not be lucrative by causing additional processes such as assembly. If part decomposition generates too many assemblies, the assembly cost would increase. However, if an original model is not sufficiently decomposed, parts would be too big to fit in the workspace of an AM machine. As such, determining the proper number and geometry of assemblies by part decomposition can be considered as future work. Second, since generating assemblies by part decomposition causes assembly processes, this research can be extended to include the *Design for Assembly (DFA)* issues considering the ease of assembly and other connection types such as fasteners and interlocking. Third, the random-based part decomposition method can be elaborated by

Table 2
The 20 cases for part decomposition and batch placement (feasible size = 120 mm).

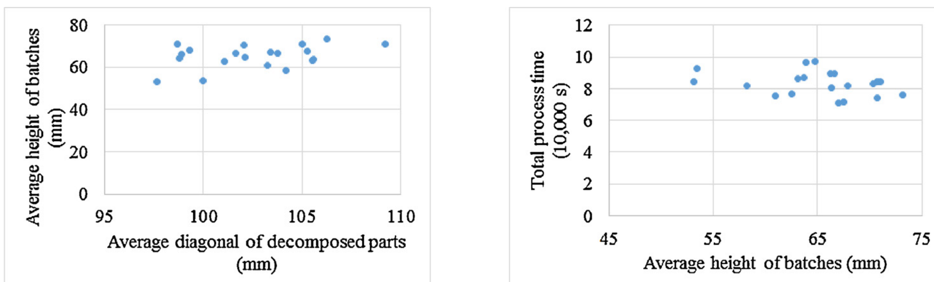
Case	Phase 1: Part decomposition		Phase 2: Genetic algorithm (GA)							
			2D batch placement				3D batch placement			
	Number of decomposed parts	The average diagonal of decomposed parts (mm)	Number of batches	The average height of batches (mm)	Support Amount (mm3)	Total process time (s)	Number of batches	The average height of batches (mm)	Support Amount (mm3)	Total process time (s)
1	30	98.81	9.00	63.87	430,974	96,328	7.00	91.16	690,473	102,147
2	22	104.17	8.00	58.23	369,451	81,575	6.00	83.05	553,018	83,477
3	27	100.00	10.00	53.42	442,489	92,660	6.00	93.02	771,180	91,425
4	29	102.13	9.00	64.76	427,407	97,252	7.00	82.98	663,478	95,165
5	23	105.52	8.00	63.06	490,204	86,584	6.00	85.72	806,099	86,152
6	22	101.06	7.00	62.53	378,467	76,965	6.00	73.08	429,944	75,942
7	21	103.77	7.00	66.37	446,849	80,356	6.00	83.91	629,944	84,280
8	17	103.40	6.00	67.02	293,352	71,038	5.00	77.56	431,794	68,455
9	25	98.68	7.00	70.69	521,698	84,318	5.00	95.86	1009,901	81,332
10	24	99.29	7.00	67.88	394,305	81,561	5.00	88.43	839,183	76,352
11	28	98.90	8.00	66.24	501,676	89,821	7.00	83.70	686,828	95,809
12	23	102.03	7.00	70.30	342,693	83,423	6.00	80.99	674,567	82,369
13	24	101.63	8.00	66.58	401,559	89,741	5.00	94.44	792,303	79,824
14	16	105.28	6.00	67.50	302,379	71,383	5.00	79.48	403,084	69,499
15	27	97.65	9.00	53.10	386,719	84,487	6.00	83.69	818,657	84,845
16	19	103.27	7.00	60.91	460,940	75,747	5.00	83.74	519,149	72,472
17	18	106.25	6.00	73.19	458,798	75,986	6.00	73.32	687,177	76,732
18	22	104.99	7.00	71.04	428,457	84,253	6.00	84.73	474,975	84,462
19	22	105.58	8.00	63.72	380,403	86,876	6.00	80.66	709,990	82,203
20	18	109.20	6.00	70.70	376,110	73,954	5.00	89.93	704,071	76,686
Average			7.50	65.06	411,746	83,215	5.80	84.47	664,791	82,481



(a) Correlation between the number of batches and the number of decomposed parts (correlation coefficient = 0.85)

(b) Correlation between the total process time and the number of batches (correlation coefficient = 0.86)

Fig. 10. The relation between the number of decomposed parts and the total process time of 2D batch placement.



(a) Correlation between the average height of batches and the average diagonal of decomposed parts (correlation coefficient = 0.38)

(b) Correlation between the total process time and the number of batches (correlation coefficient = -0.30)

Fig. 11. The relation between the size of decomposed parts and the total process time of 2D batch placement.

considering mechanical properties such as durability. Moreover, the objective function can consider other criteria such as process cost and surface quality and the estimation model can be elaborated to obtain more accurate time. Additionally, the 2D batch placement with a single AM machine can be extended to multiple AM machines. In this case, it is combined with the scheduling problem for AM in which multiple parts are assigned to multiple AM machines. Lastly, to represent parts for 2D placement, other methods instead of bounding boxes can be considered to save wasted space.

Acknowledgments

This material is based upon work supported by the National Science Foundation – USA under grant # CMMI- 1727190. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Ambriz S, et al. Material handling and registration for an additive manufacturing-based hybrid system. Int J Ind Manuf Syst Eng 2017;45(Supplement C):17–27. October.
[2] Le VT, Paris H, Mandil G. Process planning for combined additive and subtractive manufacturing technologies in a remanufacturing context. Int J Ind Manuf Syst Eng 2017;44(Part 1):243–54. July.
[3] Gibson I, Rosen D, Stucker B. Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing. Springer; 2014.
[4] Hager I, Golonka A, Putanowicz R. 3D printing of buildings and building components as the future of sustainable construction? Procedia Eng 2016;151(Supplement C):292–9. January.
[5] Lim S, Buswell RA, Le TT, Austin SA, Gibb AGF, Thorpe T. Developments in construction-scale additive manufacturing processes. Autom. Constr. 2012;21(Supplement C):262–8. Jan.
[6] Giffi CA, Gangula B, Illinda P. 3D opportunity in the automotive industry: additive manufacturing hits the road. Deloitte University Press; 2014. May.
[7] Luo L, Baran I, Rusinkiewicz S, Matusik W. Chopper: partitioning models into 3D-printable parts. ACM Trans Graph 2012;31(November (6)). pp. 129:1–129:9.
[8] Oh Y, Zhou C, Behdad S. Part decomposition and assembly-based (Re) design for additive manufacturing: a review. Addit Manuf. 2018;22(August):230–42.
[9] Zhang Y, Gupta RK, Bernard A. Two-dimensional placement optimization for multi-parts production in additive manufacturing. Robot Comput.-Integr Manuf 2016;38(April):102–17.
[10] Chen X, et al. Dapper: decompose-and-pack for 3D printing. ACM Trans Graph 2015;34(October (6)). pp. 213:1–213:12.
[11] Bikas H, Stavropoulos P, Chryssolouris G. Additive manufacturing methods and modelling approaches: a critical review. Int J Adv Manuf Technol 2016;83(March (1–4)):389–405.
[12] Dimitrov D, Schreve K, Taylor A, Vincent B. Rapid prototyping driven design and realisation of large components. Rapid Prototyp J 2007;13(April (2)):85–91.
[13] Oh Y, Behdad S, Zhou C. Part separation methods for assembly-based design in additive manufacturing. Presented at the the ASME 2017 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2017). 2017.
[14] Ren L, Sparks T, Ruan J, Liou F. Process planning strategies for solid freeform fabrication of metal parts. Int J Ind Manuf Syst Eng 2008;27(October (4)):158–65.
[15] Wang WM, Zanni C, Kobbelt L. Improved surface quality in 3D printing by optimizing the printing direction. Comput Graph Forum 2016;35(May (2)):59–70.
[16] Duncan N, Yu L-F, Yeung S-K. Interchangeable components for hands-on assembly based modelling. ACM Trans Graph 2016;35(November (6)). pp. 234:1–234:14.
[17] Yao J, Kaufman DM, Gingold Y, Agrawala M. Interactive design and stability analysis of decorative joinery for furniture. ACM Trans Graph 2017;36(March (2)). pp. 20:1–20:16.
[18] Chan CK, Tan ST. Volume decomposition of CAD models for rapid prototyping technology. Rapid Prototyp J 2005;11(September (4)):221–34.
[19] Medellín H, Corney J, Davies JBC, Lim T, Ritchie JM. Algorithms for the physical rendering and assembly of octree models. Comput-Aided Des 2006;38(January (1)):69–85.
[20] Hao J, Fang L, Williams RE. An efficient curvature-based partitioning of large-scale STL models. Rapid Prototyp J 2011;17(March (2)):116–27.
[21] Vanek J, et al. PackMerger: a 3D print volume optimizer. Comput Graph Forum 2014;33(September (6)):322–32.
[22] Yao M, Chen Z, Luo L, Wang R, Wang H. Level-set-based partitioning and packing optimization of a printable model. ACM Trans Graph 2015;34(October (6)). pp. 214:1–214:11.
[23] Attene M. Shapes in a Box: disassembling 3D objects for efficient packing and fabrication. Comput Graph Forum 2015;34(December (8)):64–76.
[24] Hu R, Li H, Zhang H, Cohen-Or D. Approximate pyramidal shape decomposition. ACM Trans Graph 2014;33(November (6)). 213:1–213:12.
[25] Ikonen I, Biles W, Kumar A, Ragade RK, Wissel JC. A genetic algorithm for packing three-dimensional non-convex objects having cavities and holes. Proc 7th Int Conf Genet Algorithms 1997:591–8.

[26] Hur S-M, Choi K-H, Lee S-H, Chang P-K. Determination of fabricating orientation and packing in SLS process. J Mater Process Technol 2001;112(May (2–3)):236–43.
[27] Zhang X, Zhou B, Zeng Y, Gu P. Model layout optimization for solid ground curing rapid prototyping processes. Robot Comput-Integr Manuf 2002;18(February (1)):41–51.
[28] Gogate AS, Pande SS. Intelligent layout planning for rapid prototyping,". Int J Prod Res 2008;46(October (20)):5607–31.
[29] Lutters E, ten Dam D, Faneker T. 3D nesting of complex shapes. Procedia Cirp 2012;3(January):26–31.
[30] Wu S, Kay M, King R, Vila-Parrish A, Warsing D. Multi-objective optimization of 3D packing problem in additive manufacturing. IIE annual conference. Proceedings. 2014. p. 1485–94.
[31] Canellidis V, Dedoussis V, Mantzouratos N, Sofianopoulou S. Pre-processing methodology for optimizing stereolithography apparatus build performance. Comput Ind 2006;57(June (5)):424–36.
[32] Canellidis V, Giannatsis J, Dedoussis V. Efficient parts nesting schemes for improving stereolithography utilization. Comput-Aided Des 2013;45(May (5)):875–86.
[33] Fuchs H, Kedem ZM, Naylor BF. On visible surface generation by a priori tree structures. Proceedings of the 7th annual conference on computer graphics and interactive techniques. 1980. p. 124–33.
[34] Stavropoulos P, Foteinopoulos P. Modelling of additive manufacturing processes: a review and classification. Manuf Rev 2018;5:2.
[35] Foteinopoulos P, Papacharalampopoulos A, Stavropoulos P. On thermal modeling of Additive Manufacturing processes. CIRP J Manuf Sci Technol 2018;20(January):66–83.
[36] Byun H-S, Lee KH. Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making. Robot Comput-Integr Manuf 2006;22(February (1)):69–80.
[37] Byun HS, Lee KH. Determination of the optimal part orientation in layered manufacturing using a genetic algorithm. Int J Prod Res 2005;43(July (13)):2709–24.
[38] "FreeCAD: An open-source parametric 3D CAD modeler." [Online]. Available: https://www.freecadweb.org/. [Accessed: 22-Oct-2017].