Proceedings of the ASME 2018 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2018
August 26-29, 2018, Quebec City, Quebec, Canada

DETC2018-85662

OAPS: AN OPTIMIZATION ALGORITHM FOR PART SEPARATION IN ASSEMBLY DESIGN FOR ADDITIVE MANUFACTURING

Angshuman Deka
Graduate Research Assistant,
Department of Mechanical and Aerospace Engineering
University at Buffalo - SUNY
Buffalo, NY - 14260
angshuma@buffalo.edu

Sara Behdad*
Assistant Professor,
Department of Mechanical and Aerospace Engineering
Industrial and Systems Engineering Department
University at Buffalo – SUNY
Buffalo, NY - 14260
sarabehd@buffalo.edu

ABSTRACT
Additive Manufacturing (AM) provides the advantage of producing complex shapes that are not possible through traditional cutting processes. Along with this line, assembly-based part design in AM creates some opportunities for productivity improvement. This paper proposes an improved optimization algorithm for part separation (OAPS) in assembly-based part design in additive manufacturing. For a given object, previous studies often provide the optimal number of parts resulting from cutting processes and their corresponding orientation to obtain the minimum processing time. During part separation, the cutting plane direction to generate subparts for assembly was often selected randomly in previous studies. The current work addresses the use of random cutting planes for part separation and instead uses the hill climbing optimization technique to generate the cutting planes to separate the parts. The OAPS provides the optimal number of assemblies and the build orientation of the parts for the minimum processing time. Two examples are provided to demonstrate the application of OAPS algorithm.

Keywords: Assembly Design for Additive Manufacturing, Part Decomposition, Cutting Plane

1. INTRODUCTION
The basic guideline for Design for Assembly (DFA) is to reduce the number of parts and facilitate the assembly process [1]. Additive manufacturing (AM) helps in reducing the number of parts in coherence with the DFA guideline. Using AM technology, manufacturers can produce a single part regardless of shape complexity which may not be possible through traditional manufacturing processes [2]. However, recently it has been discussed that in additive manufacturing a consolidated product will not necessarily result in better productivity compared to a part formed by multi-part assembly [4].

Assembly based design has several advantages. It provides a pathway for product innovation in AM [3]. It also can improve AM productivity. In addition, it helps with achieving mechanical functionality for specific purposes and using a pre-defined build space to fit a given large product and manufacturing large-scale products [4].

The focus of design for additive manufacturing field has been mainly on combining multiple parts into one part to produce a product with desired components or functionality. Part separation techniques for assembly design in AM is quite new in the literature. Oh et al. [4] investigated this issue and proposed an algorithm which provides the optimal number of parts that a given object should be separated to and their corresponding orientations with the aim of minimizing processing time, including assembly time and build time. In their study, the processing time has been calculated based on the separated parts and the part separation was done using random cutting planes. This paper proposes an improved version of the previous algorithm by eliminating the random cutting criterion. The new algorithm named OAPS focuses on increasing the productivity by minimizing the processing time. Instead of random cutting planes, the OAPS uses the hill climbing optimization technique to generate cutting planes by using the cutting plane information of the most recent cut.

Two examples are provided to compare the results of both algorithms. In addition to decreasing the minimum processing time for an object using OAPS, it was seen that the results from the OAPS are more consistent. When both algorithms are run for the same duration, for an optimized number of parts, the OAPS results are more consistent than the previous algorithm

The rest of this paper has the following structure. Section 2 reviews the available literature. Section 3 discusses the algorithm proposed in this paper. The computational results and comparisons of both algorithms are provided in Section 4. Finally, Section 5 discusses the conclusions and provides directions for future research.

2. BACKGROUND AND RELATED WORK

The literature related to this work can be classified into three major groups: part separation techniques, assembly of multiple parts in additive manufacturing, and build orientation determination. This section discusses these three groups and describes the problem under study.

2.1 Techniques for part separation

Part separation has been the point of attention in additive manufacturing literature. To name several studies, Luo et al. [5] developed a framework named Chopper, where they used planar cuts for developing a binary space partitioning. They decomposed 3D models with the purpose of fitting all the parts in the limited printable volume of an AM machine. Chopper was validated by parts printed using Fused Deposition Modeling (FDM) and PolyJet technology. Hildebrand et al. [6] developed mutually intersecting planar cut-outs of the cardboard 3D models. They proposed an extended binary space partitioning tree to represent the cardboard models so that they can quickly evaluate the feasibility of newly added planar elements to the model. Hu et al. [7] separated a 3D object into multiple approximately pyramidal parts to minimize the time and cost when printing on an FDM printer. Song et al. [8] proposed the voxelization approach to separate parts from a 3D model into 3D interlocking parts. They validated their approach using three different types of printing technologies – FDM, Selective Laser Sintering, and Stereolithography. Chen et al. [9] proposed an algorithm named Dapper for addressing decompose-and-pack problems. The algorithm was proposed for powder and FDM based 3D printers with the aim of decomposing any input shape into smaller parts and packing them efficiently. The input parts in this process are initially partitioned into pyramidal parts to voxelize them into pyramidal polycubes. Oh et al. [10] showed how part separation for assembly design can play a role in minimizing processing time, cost, and surface roughness. They used Digital Light Processing AM technology to assess the assembly part design quantitatively.

The advantages of part separation and its impact on improving productivity have been discussed in the above-mentioned literature. In the current work, planar cuts have been used to separate the 3D object. A given part is initially separated into two parts and then the part with the largest surface area is separated further using the proposed algorithm. The process is continued, and the technique is applied to the parts until the given 3D model has been separated to the optimized number of parts that minimize the total processing time.

2.2 Assembly of multiple parts in additive manufacturing

There is not much literature available about assembly in additive manufacturing. Ahmad et al. [11] proposed guidelines for facilitating the use of rapid prototyping in training for assembly and validated the guidelines using a case study. Hallgrena [12] discussed redesigning for metal additive manufacturing by redesigning multiple parts of a product. Crane et al. [13] discussed how self-assembly, which is defined as the positioning and bonding of components by random interactions, can be integrated with additive manufacturing. Yagnik [14] discussed how 2500+ parts produced using FDM was assembled for a jet engine prototype. The prototype helped the working teams to understand how the parts will fit during actual manufacturing of the engine. On the other hand, Cali et al. [15] have proposed a method where they converted CAD models into functional non-assembly models. Won et al. [16] also developed a fabricated prototype of a three-legged, six degree-of-freedom parallel manipulator which did not require assembly.

It can be seen in the literature that the focus has been mainly on design for part consolidation or for designing an object with multiple parts with a final goal to assemble them. Part consolidation can improve productivity by decreasing the need for assembly time. The current work also aims to improve productivity by decreasing processing time. However, this work focuses on how to separates a given object into multiple parts rather than part consolidation.

2.3 Build orientation of parts

An important decision for additive manufacturing is the build orientation of the parts. Many key characteristics which drive the quality and cost of a part depends on the build orientation [17]. In deciding the build orientation, there are two major steps that need to be addressed: 1) determining the feasible orientations from the infinite number of orientations possible by rotating the part in any angle in the three axes, 2) conducting single criterion or multi-criteria optimization based on factors affected by the orientation to determine the optimal solution [18].

To name a few studies that have addressed build orientation problem, Alexander et al. [17] decided build orientation based on surface accuracy and cost. They also proposed cost models and showed the interdependency of orientation and process cost. Frank et al. [19] developed the build orientation considering three factors as the objective: surface quality, build time, and support structures. Cheng et al. [20] proposed another multi-objective approach for determining the optimal build orientation. They considered part accuracy and build time as their objectives. Masood et al. [21] considered minimizing the volume error, described as the difference in the amount of material used by the printer and the amount suggested by the CAD model, as their objective for obtaining the optimal build orientation. Goyal et al. [22] presented a method that obtained the optimized build orientation from pre-selected orientations based on the minimum number of adaptive slices. West et al. [23] proposed a process planning method that implemented an optimization to minimize the aggregate measure of deviation from accuracy, finish, and build time targets. Build orientation was one of the variables used for orientation, and the other two being layer thicknesses and SLA process variables. Lan et. al [24] determined build orientation for SLA considering surface quality, build time, and complexity of the support structures.

As discussed above, the importance of the build orientation has been highlighted in the literature. In the current work, the orientation played an important role as the height of the parts were used for calculating the processing time. In addition, an assumption was made that any build orientation proposed by this algorithm can be printed by the printer.

A heuristic algorithm was previously proposed to minimize the total processing time of an object, when separated and printed into different parts and finally assembled to get the desired product [4]. The algorithm gave the optimal number of parts an object should be separated to and their corresponding orientations to minimize processing time. Random cutting planes were used for part separation, which does not ensure that the minimum possible processing time was obtained. Therefore, there is a high chance that the solution does not lie near any optimum points. The current work addresses the shortcoming of random cutting planes and uses an optimization technique to generate cutting planes which give improved processing time. In the proposed OAPS algorithm, a cutting plane is generated during the part separation process by using the plane information from the most recent cut to obtain the processing time. When an optimization technique is applied to find an optimum cutting plane, it results in improved estimation for processing time. Table 1 summarizes the major differences between the previous algorithm and OAPS.

Table 1: Differences between the previous algorithm and OASP

Factors	The previous algorithm by Oh et al. [4]	OAPS
Cutting Plane	The cutting plane is randomly generated.	An optimized cutting plane is used.
Build Orientation	Build orientations are pre-defined	Build orientations are not pre-defined.

In random cutting planes method, the processing time can differ significantly when the same part is separated using different planes as the cutting planes are completely random. When an optimization technique is used instead, the cutting planes during the iterations are updated from the most recent cutting plane information while the solution converges to a minimum. In the hill climbing optimization technique used here, multiple starting points are considered for defining the cutting plane and an optimum processing time is reached in each case. The optimum results lie in a narrow range even if different starting points are chosen for cutting planes. Thus, in addition to providing an improved estimation of processing time, consistent results will be obtained by using an optimization technique.

3. OAPS ALGORITHM

In OAPS, part separation method drives the build orientation and assembly process that finally influence the productivity. The main advantages of OAPS compared to the previous algorithm are the improved processing time estimation and consistent results. The downside of the algorithm is that an iteration takes longer for the OAPS, but the consistent results compensate for the same.

The same equations have been used to calculate the processing time as prior work [4]. The flowchart in Figure 1 gives a general overview of the algorithm with detailed explanations in the subsequent sections. Section 3.1 briefly describes the previous algorithm. Section 3.2 discusses the function to be optimized. Section 3.3 explains the build time and assembly time. The optimization procedure is discussed in detail in Section 3.4.

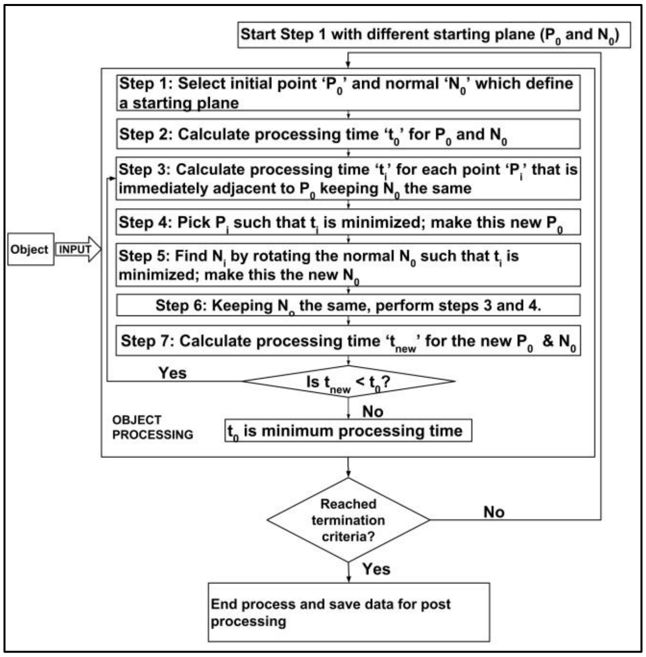

Figure 1: Flowchart giving an overview of the algorithm

3.1 Previous algorithm

In the previous algorithm, the processing time for the initial unseparated object is calculated first at its optimal orientation, which was the orientation with minimum height. Next, the maximum number of parts a given object should be separated was calculated, such that it does not exceed the initial time calculated above. The object is now separated using a plane whose cutting direction is generated randomly, and the processing time is calculated for the corresponding orientation of the parts. If the processing time is less than the initial time, the part with the larger volume is separated further and the processing time is calculated again. This process is repeated unless the processing time exceeds the time for the initial unseparated object.

3.2 Optimization function: total processing time

The goal of OAPS is to minimize the total processing time when a part is produced using AM. The processing time T, which is the sum of build time and assembly time is driven by the part shape, build orientation, and the total number of parts. As mentioned above, it is assumed that any build orientation generated by the algorithm can be printed. The objective function can thus be represented as:

$$Minimize\ total\ processing\ time, T = Build\ time\ (t_b) + Assembly\ time\ (t_a) \qquad (1)$$

3.3 Build time and assembly process and time

Build time:

OAPS has been developed considering a selective laser sintering (SLS) AM process. For SLS printing, build time can be represented by [25]:

$$t_b = Time\ for\ machine\ preparation\ (t_{mp}) + Time\ for\ layer\ drawing\ (t_{ls}) + Time\ for\ layer\ preparation\ (t_{lp}) + Time\ for\ ending\ operation\ (t_e) \qquad (2)$$

In the above-mentioned equation, the time for machine preparation and the time for ending operation are constant for a given SLS printer. Thus, the build time is derived from the time for layer drawing and the time for layer preparation as described below [25]:

$$Time\ for\ layer\ drawing\ (t_{ls}) = (V_o/t)\ /\ [N\ (d_l + h_d)\ v]\ +\ (A_o/t)/v \qquad (3)$$

where V_o, t, N, d_l, h_d, v, and A_o denote the volume of the object, layer thickness, the number of laser heads, laser diameter, hatching distance, laser scanning speed and the total surface area of parts, respectively.

Time for layer preparation (t_{lp}) is represented as [25]:

$$Number\ of\ layers\ X\ time\ per\ layer = (h_{max}/t)\ t_l \qquad (4)$$

where t_l, h_{max} and t denote the fixed unit time for preparing single layer, the maximum height of all the parts, and the layer thickness, respectively.

Assembly process and time:

Sodhi et al. [26] described different types of methods that can be used to assemble parts in product design. One of the methods that they discussed is the use of adhesives for assembly. In the current study, it is assumed that parts will be joined using adhesives. A standard unit time for joining two parts is estimated. Thus, the assembly time can be estimated as:

$$t_a = [Standard\ unit\ time\ X\ (Number\ of\ joints)] = [Standard\ unit\ time\ X\ (Number\ of\ parts - 1)] \qquad (5)$$

In the current work, the assembly time is considered as constant. Thus, the time is a linear function of the number of parts that a given object is separated to. A trade-off is needed between the number of parts an object should be separated while trying to minimize the processing time. In addition, it is not necessary that a standard constant time is always considered for assembly. The assembly time can be a function of different factors such as part geometries, the number of fasteners, and the type of fasteners.

3.4 Optimization procedure

A planar based separation method is used to divide the object into multiple parts. The aim of the current work is to replace the random cutting criterion used in previous studies by using optimized cutting planes. The optimization technique applies the information from the most recent cutting plane to generate an updated new cutting plane.

A local optimum search technique known as hill climbing algorithm, which uses an iterative algorithm consisting of a neighborhood and an objective function is used in this work [27]. First, a candidate solution in a neighborhood is initially determined. Then, a new candidate solution is determined from the possible solutions in the neighborhood of the initial solution. The value of the objective function is now assessed and if it is less than the initial value, it is accepted, otherwise rejected. The process is repeated multiple times unless there is no neighborhood solution which gives the objective function to be less than the current value. The iteration is stopped when a local minimum is obtained. The major drawback of this technique is that it converges to a local minimum and there is no guarantee that a global optimum is obtained.

The hill climbing optimization is an effective technique in the case of large spaces with many possible candidate solutions. The hill climbing optimization finds a local minimum and is similar to gradient descent algorithms. The difference in hill climbing method is that it uses a random local search to find the direction and step size instead of a gradient to determine the next step in the search [28]. In this method, an initial point is selected, and an attempt is made to improve the objective function by moving to a local point. The value of the objective function for the neighboring configuration is calculated and if there is an improvement, the point is updated. The process is repeated multiple times unless there is no neighboring point which can reduce the value of the objective function. The algorithm is summarized as follows:

Start with an initial point I_o

Repeat the following steps:

- Compute the objective function at multiple neighboring points I_1
- If the objective function at one of the $I_1 <$ Objective function at Io
- Update I_o with I_1

Return final point which minimizes the objective function.

This idea behind this technique is that multiple local optimum points are obtained by starting from different initial points, Io.

The time required for SLS printing depends on the time for layer drawing and the time for layer preparation. The time for layer drawing in Equation 3 depends on the total volume of the parts, V_o and the total surface area of the parts, A_o. All the other variables in the equation are printer dependent and is independent of the object to be separated. Similarly, for the time for layer preparation in Equation 4, the variables fixed unit time for preparing one layer and the layer thickness depend on the printer. Therefore, for any given part, the time for layer preparation is dependent on the maximum height h_{max} of the parts to be printed. Therefore, both Equations 3 and 4 can be simplified as follows:

$$t_{ls} = V_o \,/\, (Constant) + A_o \,/\, (Constant) \tag{6}$$

$$t_{lp} = (Constant) * h_{max} \tag{7}$$

where the above-mentioned constants depend on the printer.

Since, the time for machine preparation and the time for ending operation are constant, the processing time depending on the part to be separated can thus be calculated as:

$$t_{lp} + t_{ls} + t_a = [(Constant) * h_{max}] + [V_o \,/\, (Constant) + A_o \,/\, (Constant)] + [Standard\ unit\ time\ X\ (Number\ of\ parts - 1)] \tag{8}$$

where, the term $[V_o \,/\, (Constant)]$ will be a constant for a given object and is independent of the total number of resulting parts. The surface area of the objects, A_o will change depending on the cutting plane, and therefore it is a cut dependent factor. Also, the value of t_a is a linear function and will remain constant for a fixed number of parts.

A given 3D part is initially separated into two parts, using a cutting plane. Once the part is separated the total processing time is calculated. To do this, multiple orientations are checked and the orientation with optimum height is selected for each part. h_{max} is then calculated which is the maximum height of all the parts. As discussed above the maximum height of the parts, the total surface area of parts, the volume of the parts, and the number of parts contribute to the total processing time. In this algorithm, the initial cutting plane is defined using a direction vector and a point lying in the part as these two are the most fundamental elements to define a plane. Once the initial time has been calculated after cutting using the plane, the next step is to perform the optimization to update the plane to get a better processing time than the initial time.

The hill optimization technique is now implemented to find the updated cutting plane. The first step is to update the plane by updating the point. 6 different neighboring points are evaluated, keeping the direction vector constant to see if the value of the objective function decreases. Once a point that decreases the objective function is found, then the direction vector is updated in the neighborhood, and six new directions are evaluated to check if a minimum value can be found compared to the existing value. If a new direction vector gives a better value, then six new points in the neighborhood are evaluated again to confirm that the minimum objective function is obtained at that point. This process is repeated until there is no point or vector that can generate a plane to further minimize the value of the objective function. Again, the process is started using a random point and a direction vector. This whole process is repeated multiple times to tackle the problem that hill climbing optimization gives a local minimum. If a similar minimum value of the function is obtained multiple times even after starting from random starting points for the point and direction vector defining the plane, it can be concluded that the solutions are close to the global minimum.

Once the initial object has been separated into two parts, the part with a larger surface area is separated further using the same process described above. Both volume and surface area could have been used as criteria for part selection for further separation as both of them give a true representation of the object. On the other hand, it was observed that deciding on the basis of optimal height was not a good idea as it was observed that there were parts with large height but very thin surface, and thus not representing the correct object size. For example, there were parts with similar height but the surface area of one was nearly five times the other.

As discussed in Section 3.3, there is an upper limit to the number of parts into which a given object should be separated. A given part can be separated into infinite parts. But the algorithm will not be useful if the processing time for a solid unseparated object is less than the processing time of the separated parts. Thus, the number of parts a given object is separated is limited by the processing time for the original unseparated object.

To summarize, the advantages of the proposed algorithm are as follow:

1. The algorithm accepts any user input for the direction of normal and the point used to define the initial cutting plane.
2. The users can define how much a normal direction is changed when updating the normal direction during the search.
3. The users can define the step size while updating the point.
4. For a given object, a scale factor can be used to change the size. This eliminates the need to create different sizes of the same model to calculate the processing time.

These advantages can help users calculate the processing time faster. For example, a large step size for updating the point or normal can be used. This will result in faster convergence to an optimum compared to using small step sizes since the search space will be covered quicker. The downside to this is that the results may be less accurate than using a smaller step size.

4. CASE STUDY AND RESULTS

Two different case studies have been performed to compare the performance of OAPS with the previous algorithm. Both algorithms have been run for approximately the same duration. The number of iterations completed for both algorithms has been reported in their corresponding sections. For both examples, the number of iterations of OAPS completed during an hour was less than the previous algorithm. This was because the previous algorithm generated planes randomly and separated the parts to get the processing time. On the other hand, each optimum point of OAPS was obtained after performing multiple optimization iterations to reach that point. For both case studies, the standard unit time for assembly is assumed as 120 seconds. The printer parameters for the case studies are the same as the ones used by Zhang et al. [25] as summarized in Table 2.

Table 2: Printer parameters [25]

Machine	**EOSINTP385**
Material used	PA2200
Layer thickness (t)	0.15 mm
Hatching distance (h_d)	0.33 mm
Laser diameter (d_l)	0.6 mm
Laser scanning speed (v)	700 mm/sec
Laser head (N)	1
Layer preparation time (t_l)	6 seconds

In both algorithms, the initial object was separated into two parts, m1 and m2. The build time was calculated for all the optimization iterations completed during the time the algorithms were run. For the iteration with the lowest build time, the part out of m1 and m2 with the higher surface area was separated further. The separations were carried out further until it exceeded the processing time of the consolidated object which has not gone through any part separation. There were certain iterations where the parts were generated with very low height or surface area. Although the object was very large, parts with small surface area or height were generated when the cutting plane passed near a corner or edge, separating the parts into two, with one part very small. Those parts were not considered when finding the parts with lowest build time for further separation.

4.1 Stanford Bunny

The Stanford Bunny [29] is a part of the ‘The Stanford 3D Scanning Repository’ of the Stanford Computer Graphics Laboratory. The Stanford bunny, scanned by Greg Turk and Marc Levoy [30] has been used to test the proposed algorithm. One of the original ply files was used to test the algorithm which ensures that the original dimensions are maintained. The dimensions of the object are 194.5(mm) X 147(mm) X 190.5 (mm). Figure 2 [29] shows an image of the Stanford Bunny.

Figure 2: Stanford Bunny [29]

The processing time for the solid unseparated part is calculated as 21,144 seconds for the optimum orientation where it has an assembly time of zero seconds. Both algorithms were run for approximately one hour. Table 3 gives the number of iterations completed during that time.

Table 3: Total number of iterations completed in one hour

	Total number of parts			
	2	**3**	**4**	**5**
Number of iterations for OAPS	12	12	15	15
Number of iterations for the previous algorithm	2700	2700	2700	2700

Table 4 gives a comparison of the processing time of both algorithms when the initial part is separated into multiple parts. For the previous algorithm, the minimum processing time was 20,833 seconds when separated into 2 parts. For the OAPS, for the minimum processing time of 20,649 seconds, the optimal number of parts was 3. In both algorithms, anything more than 4 parts was not a feasible solution as the processing time exceeds the build time for the solid unseparated part, 21,144 seconds. Figure 3 shows the isometric view of the optimal orientation when the bunny model is separated into 3 parts using OAPS. Figure 4 shows the same parts from a different angle for better understanding.

Table 4: Processing time comparison for Stanford bunny

	Total number of parts			
	2	3	4	**5**
Process time for OAPS	20771	20649	21160	22216
Process time for the previous algorithm	20833	20989	21139	21311

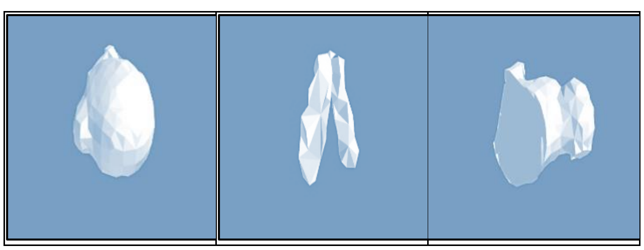

Figure 3: Isometric view of four separated parts

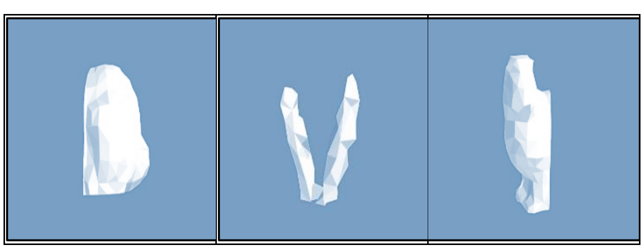

Figure 4: View from another angle

4.2 Bracket

For the second case, a simple L bracket as shown in Figure 5 was designed on SolidWorks with dimensions 25.4 (mm) X 38.1 (mm) X 76.2 (mm). The build time for the solid unseparated bracket at its best orientation was calculated to be 1356 seconds.

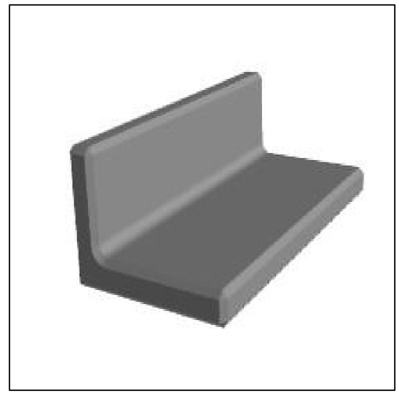

Figure 5: Bracket

Both algorithms were tested for this part for approximately 15 minutes. The minimum processing time obtained from OAPS was for 2 parts, and 813 seconds. The minimum processing time for the previous algorithm was 942 seconds for 2 parts as well. Table 5 reports the number of iterations completed for both algorithms and Table 6 reports the processing time. Figure 6 shows the resulting two parts with minimum processing time when using OAPS.

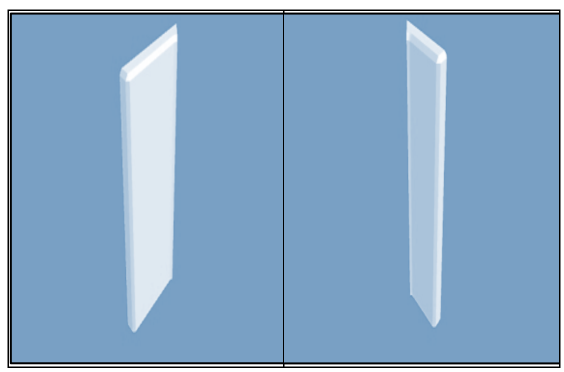

Figure 6: Isometric view of the two separated parts

Table 5: The total number of iterations completed in 15 mins

	Total number of parts					
	2	**3**	**4**	**5**	**6**	**7**
Number of iterations for OAPS	12	15	12	12	12	12
Number of iterations for the previous algorithm	1350	1350	1350	1350	1200	-

Table 6: Processing time comparison for bracket

	Total number of parts					
	2	3	4	5	6	7
Process time for OAPS	813	934	1056	1177	1300	1422
Process time for the previous algorithm	942	1007	1126	1243	1363	-

4.3 Consistency of results

Another performance criterion evaluated for both algorithms was the consistency of the results when both algorithms were run for the same duration. The performance of OAPS was more consistent compared to the previous algorithm, meaning that the OAPS consistently converged to minimum processing times which were close to each other. Tables 7 and 8 show how much the results varied from the minimum (in terms of percentage) for all the iterations for a particular number of parts in both algorithms.

Equation 9 shows how the percentage deviation was calculated:

$$Percentage\ deviation\ = [(Max\ processing\ time - Minimum\ processing\ time)/ Minimum\ processing\ time] * 100 \quad (9)$$

Table 7: Deviation comparison (in percentage) of processing time for both algorithms for Stanford Bunny

	Total number of parts		
	2	3	4
Deviation in results for OAPS	1.67	8.69	4.15
Deviation in results for the previous algorithm	19.45	30.58	30.27

For the previous algorithm, although the minimum processing time for part separation was less than the processing time for the unseparated objected, it was observed that most of the results were higher than the time for the unseparated object.

Table 8: Deviation comparison (in percentage) of processing time for both algorithms for bracket

	Total number of parts					
	2	3	4	5	6	7
Deviation in results for OAPS	4.36	2.73	0.89	1.15	1.10	0.93
Deviation in results for the previous algorithm	70.88	24.04	14.11	17.56	13.3	-

Figure 7 below shows the scatter plots for processing time when the Stanford Bunny is divided into an optimal number of parts. The red line in the plot is the processing time for the unseparated object. Similarly, Figure 8 shows the processing time scatter plot for the bracket. For both cases, it can be seen that most of the values are above the processing time for the unseparated object. This further shows that the processing time is not consistent when random cutting planes are used in the previous algorithm, and there is a high chance that the processing time will lie away from the minimum.

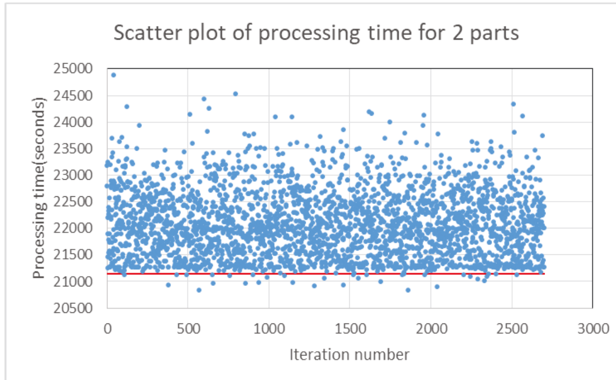

Figure 7: Scatter data of processing time for Stanford Bunny for 2 parts using the previous algorithm

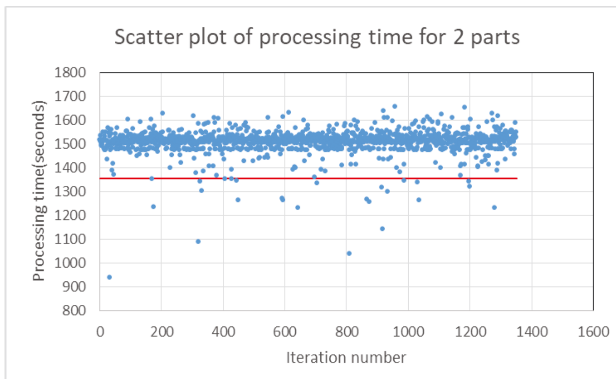

Figure 8: Scatter data of processing time for the bracket for 2 parts using the previous algorithm

4.4 Sensitivity analysis

A sensitivity analysis was performed for OAPS to see how changing the step size affects the performance of the algorithm. Two experiments were performed, and the Bunny was split once for both cases to get two parts. As mentioned above, a point and a cutting plane were used to form the plane. To test the sensitivity of the step size during searching for optimum, the step size during updating the point to form the cutting plane (Step 3 in Figure 1) was varied relative to the original step size used in the algorithm. The step size for searching the normal was not varied during this time. 12 iterations were performed for both cases and the results are reported in Table 9. The results prove that varying the step size during the search has an impact on the processing time.

Table 9: Minimum processing time variation with a step size

Change in step size	Minimum processing time
Increased 100 %	21076
Decreased 33%	20755

The results show that the OAPS provides improved estimation for processing time. In addition, as discussed previously when both algorithms are run for the same duration, the results from OAPS are more consistent and lie within a very short range.

5. CONCLUSION AND FUTURE WORK:

This work proposes an optimization algorithm named OAPS to find the optimal number of parts and their corresponding orientations for assembly-based design in AM. It improves an already-existing algorithm to achieve the minimum processing time during part separation. In the previous work, the cutting planes used for part separation were randomly generated. The current study offers an optimization technique to generate cutting places. Using random cutting planes will not necessarily result in an optimum processing time. In addition, the resulting processing times can lie within a wide range depending on the number of iterations used in the algorithm. To address these issues, the hill climbing optimization technique is applied to generate the cutting planes for part separation. It has shown that the OAPS results in better processing time. In addition, the results obtained from OAPS are more consistent when both algorithms are run for the same duration.

The proposed algorithm can be improved in several directions. First, the orientations generated by the algorithm that are not feasible for printing can be separated from printable ones. Second, the part-separation algorithm can be improved such that it does not generate parts with very low height or surface area. Third, the algorithm code can be optimized such that more iterations can be performed in less time. This will generate more optimum results at the same time, thereby increasing the dataset for comparison and further analysis of both algorithms. In the current algorithm, if an object is separated into more parts than desired, it cannot be detected and thus it gives undesired results. For example, when a plane passes through the bracket at a forty-five degrees angle relative to the base, it separates the bracket into three parts instead of two and thus the results obtained are not desirable. The algorithm can be improved to detect and discard such results.

In addition, this work is focused on only one AM technology. The work can be extended to other AM technologies as well. Another aspect of improvement will be to increase the number of case studies to find the pros and cons of the algorithm.

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] T.C. Kuo, S.H. Huang, H..C Zhang, “Design for manufacture and design for 'X': concepts, applications, and perspectives”, Journal of Computers and Industrial Engineering, 41, pp. 241-260, 2011.

[2] S.H. Huang, P. Liu, A. Mokasdar, and L. Hou, “Additive manufacturing and its societal impact: a literature review”, The International Journal of Advanced Manufacturing Technology, vol. 67, Issue 5-8, pp. 1191-1203, 2013.

[3] Floriane Laverne, Frederic Segonds, Nabil Anwer, and Marc Le Coq. “Assembly Based Methods to Support Product Innovation in Design for Additive Manufacturing: An Exploratory Case Study.” Journal of Mechanical Design, vol. 137, no. 12, pp. 121701-1–121701-8, Dec 2015.

[4] Yosep Oh, Sara Behdad, and Chi Zhou. “Part separation methods for assembly based design in additive manufaturing”, ASME 2017 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Cleveland, Ohio, USA, August 2017.

[5] Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. “Chopper: Partitioning Models into 3D-Printable Parts.” ACM Transactions on Graphics 31, no. 6, Article 129, November 2012.

[6] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. “Crdbrd: Shape Fabrication by Sliding Planar Slices.” Computer Graphics Forum, vol. 31, no. 2, pp. 583–592, 2012.

[7] Ruizhen Hu, Honghua Li, Hao Zhang, and Daniel Cohen-Or. 2014. Approximate pyramidal shape decomposition. ACM Trans. Graph. 33, 6, Article 213 (November 2014), 12 pages. DOI: https://doi.org/10.1145/2661229.2661244

[8] Peng Song, Zhongqi Fu, Ligang Liu, and Chi Wing Fu. “Printing 3D Objects with interlocking Parts.” Computer Aided Geometric Design, 35–36, pp.137–148, 2015.

[9] Xuelin Chen et al., Bedrich Benes, Ruizhen Hu, Daniel Cohen-or, and Baoquan Chen. “Dapper : Decompose-and-Pack for 3D Printing,” ACM Trans. Graph. Vol. 34, pp. 213:1–213:12, 2015.

[10] Yosep Oh and Sara Behdad, "Assembly Based Part Design to Improve the Additive Manufacturing Productivity: Process Time, Cost and Surface Roughness", Proceedings of the ASME 2016 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2016, August 21-24, 2016, Charlotte, NC, USA

[11] A. Ahmad, Saber Darmoul, Wadea Ameen, Mustufa H. Abidi, and Abdulrahman M. Al-Ahmari. "Rapid Prototyping for Assembly Training and Validation." IFAC-PapersOnLine 48-3, pp.412–17, 2015.

[12] Sebastian Hällgrena, Lars Pejrydb, and Jens Ekengrenb. "(Re)Design for Additive Manufacturing." Procedia 26th CIRP Design Conference, 50, pp. 246–51, 2016.

[13] N.B. Crane, J. Tuckerman, and G.N. Nielson. "Self-Assembly in Additive Manufacturing: Opportunities and Obstacles." Rapid Prototyping Journal, Vol. 17, Iss. 3, pp.211–217, 2011.

[14] Deepa Yagnik "Fused Deposition Modeling – A Rapid Prototyping Technique for Product Cycle Time Reduction Cost Effectively in Aerospace Applications." IOSR Journal of Mechanical and Civil Engineering, pp.62–68.

[15] Jacques Cali, Dan a. Calian, Cristina Amati, Rebecca Kleinberger, Anthony Steed, Jan Kautz, and Tim Weyrich. "3D-Printing of Non-Assembly, Articulated Models." ACM Transactions on Graphics, vol. 31, no. 6, Article no. 130, (2012): 1. doi:10.1145/2366145.2366149.

[16] Jey Won,, K DeLaurentis, and C Mavroidis. "Rapid Prototyping of Robotic Systems." Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065) 4, no. April (2000): 3077–82. doi:10.1109/ROBOT.2000.845136.

[17] Paul Alexander, Seth Allen, and Debasish Dutta. "Part Orientation and Build Cost Determination in Layered Manufacturing," Vol. 30, No. 5, pp. 343-358, 1998.

[18] Prashant Kulkarni, Anne Marsan, and Debasish Dutta. "A Review of Process Planning Techniques in Layered Manufacturing." Rapid prototyping Journal, Vol. 6, Issue 1, pp. 18-35, 2000.

[19] D. Frank and G. Fadel, "Expert system-based selection of the preferred direction of build for rapid prototyping processes", Journal of Intelligent Manufacturing, (1995) 6: 339. https://doi.org/10.1007/BF00124677

[20] W. Cheng, J.Y.H. Fuh, A.Y.C. Nee, Y.S. Wong, H.T. Loh, and T. Miyazawa. "Multi-Objective Optimization of Part- Building Orientation in Stereolithography," Rapid Prototyping Journal, Vol. 1 Issue: 4, pp.12-23, 1995.

[21] S.H. Massod, W. Rattanawong, P. Iovenitti, "Part build orientations based on volumetric error in fused deposition modelling", International Journal of Advanced Manufacturing Technology, 16, pp. 162–168, 2000.

[22] A. Goyal, A. Sudha, A. Kumar, P.V.M. Rao, Optimum part orientation for rapid prototyping based on estimated number of adaptive slices, Proceedings of International Conference on E-manufacturing, pp. 475 247–253, 2002

[23] A. P. West, S. P. Sambu, and D. W. Rosen. "A Process Planning Method for Improving Build Performance in Stereolithography." CAD Computer Aided Design, 33, no. 1, pp. 65–79, 2000.

[24] Po-Ting Lan, Shuo-Yan Chou, Lin-Lin Chen, and Douglas Gemmill. "Determining Fabrication Orientations for Rapid Prototyping with Stereolithography Apparatus." Computer-Aided Design, Vol. 29, no. 1, pp. 53–62, 1997.

[25] Yicha Zhang and Alain Bernard, "Generic Build Time Estimation Model for Parts Produced by SLS," High value manufacturing: Advanced research in virtual and rapid prototyping. Proceedings of the 6th International Conference on Advanced Research in Virtual and Rapid Prototyping, Portugal , pp. 43-48, 2013.

[26] R. Sodhi, M. Sonnenberg and S. Das, "Evaluating the unfastening effort in design for disassembly and serviceability" Journal of Engineering Design, vol. 15, no. 1 pp. 69-90, 2004.

[27] A.W. Johnson, S.H. Jacobson, "On the convergence of generalized hill climbing algorithms", Discrete Applied Mathematics, Volume 119, Issues 1–2, Pages 37-57, 2002.

[28] S. Chalup and F. Maire, "A study on hill climbing algorithms for neural network training," Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, 999, pp. 2021 Vol. 3. doi: 10.1109/CEC.1999.785522.

[29] http://graphics.stanford.edu/data/3Dscanrep/

[30] https://www.cc.gatech.edu/~turk/bunny/bunny.html