

A decomposition method for efficient manufacturing of complex parts

Brandon R. Massoni  and Matthew I. Campbell 

Oregon State University, USA

ABSTRACT

This paper presents a method to decompose three dimensional complex parts into readily available stock material to take advantage of advanced joining to build up a rigid assembly. The method generates many alternative assemblies by decomposing the solid geometry iteratively with cutting planes. Each assembly is then evaluated based on cost. The process continues until the developed search algorithm converges on a near optimal solution. Application of this method will reduce material waste, thus reducing per part processing time, energy consumption, and associated production costs. Example parts for a variety of metals show how the computational tool finds near optimal solutions for complex three dimensional solids.

KEYWORDS

Decomposition; assembly optimization; design automation; friction welding; advanced joining

1. Introduction

During the design process, it is often advantageous to consolidate multiple features into a single part to reduce assembly costs. However, feature consolidation can lead to complex parts that are expensive or wasteful to machine from a single piece of stock material (Fig. 1. left and middle). This is especially true for metal parts, where the material cost can be more than 50% of the production cost [1]. The best solution may be to build up complex parts from an assembly of simpler parts (Fig. 1 right). This approach is valuable as long as the assembly maintains similar mechanical properties and if the combined material and machining savings outweigh the additional assembly costs.

While other assembly operations, like bolting and gas welding, introduce stress concentrations and weaken the overall structure (as compared to machining from a single block), linear friction welding (LFW) and rotary friction welding (RFW) can join metals without melting them, such that the weld joint has mechanical properties just less than or surpassing the parent material [2][4]. If advanced joining methods, like LFW and RFW are applied correctly to the right parts, the combined material and machining savings can outweigh the additional assembly costs while maintaining similar mechanical performance.

Determining the best way to decompose a complex part into several subparts and evaluate the cost of production is time consuming for a designer. A human designer

would determine the best way to decompose the geometry using intuition and experience. The designer would then alter the geometry in CAD and store all the important geometric information. Lastly, one would request a quote for each production process or use a cost estimation tool. Based on the results, the designer would likely iterate over the process several times. While it may not take the designer long to determine the option shown in Fig. 1, this is not necessarily the optimal assembly option and it does not show the different possible stock materials or production processes. For a comprehensive search, this iterative design loop could take weeks to develop and evaluate a variety of design options, and the outcome would depend on the expertise and knowledge of the designer.

The method presented in this paper automatically generates alternative decompositions and compares their

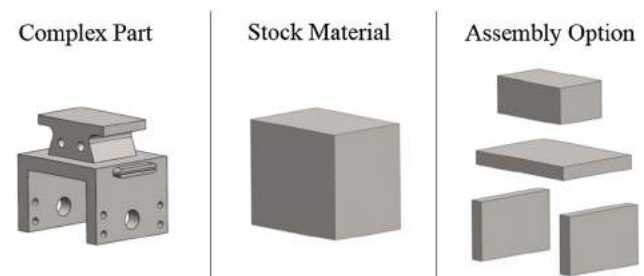


Figure 1. (left) Complex part, (middle) traditional stock material, (right) assembly option.

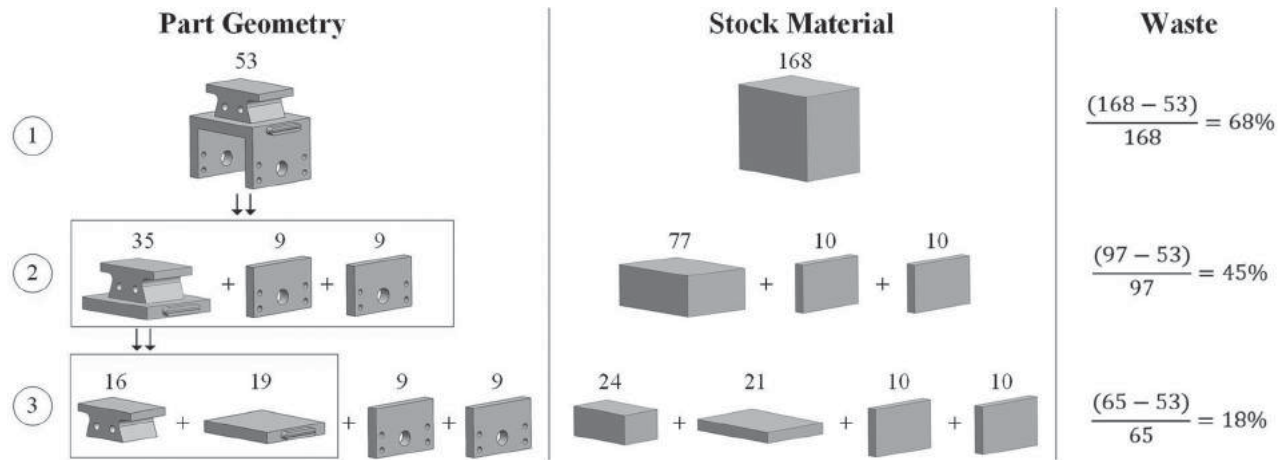


Figure 2. (left) Part geometry for three separate assembly options, (middle) stock material for each of those assemblies, (right) percentage of waste material.

relative costs. The resulting assemblies are assumed to be constructed from readily available stock material and joined together with LFW or RFW. For simplicity, this stock material is considered to be round and rectangular bar stock of every size. This method provides the best assembly quickly, with little effort from the design engineer. In addition, it considers many more alternatives than the designer could reasonably consider and has the potential to be integrated with a highly detailed cost model.

An example of this method is shown in Fig. 2. The top left corner shows the complex original part geometry. The top middle shows the stock material that the part is machined from. The top right shows the percentage of material waste (68%). If the complex part is decomposed into the three parts shown in the second row, in which each part is created from rectangular bar stock and then assembled and machined, the resulting material waste is less (45%). Lastly, if decomposed a second time into the part geometry in the bottom row, the resulting material waste is much less (18%). However, the further a complex part is decomposed, the more assembly operations are required. Keeping this in mind, which assembly is the least expensive to manufacture? The answer depends on the costs of the material, machining, and assembly operations. All of these cost components are used within the presented search to find the least expensive assembly to manufacture.

2. State of the art

There are a variety of approaches in the literature that focus on decomposing parts to improve design or manufacturability. While our method bears some similarity to literature in process planning, additive manufacturing,

and hybrid manufacturing, the application and goals are distinctly different. To our knowledge, this is the first attempt to automate decomposition of parts into assemblies to take advantage of advanced joining methods, like LFW. Since this is a unique problem not yet tackled in literature, this section discusses some of the innovation and background of similar work.

2.1. Process planning research

Process planning motivates the bulk of research on part decomposition and formed the starting point for the decomposition method presented in this paper. In 1991, Tang and Woo [18] developed the Alternating Sum of Volumes (ASV) approach, which uses convex hulls to decompose machined features. The machined features could then be represented by a list of convex hulls with alternating signs. This method, however, does not guarantee convergence [19]. Kim [6] identified the cause of non-convergence and corrected it in the Alternating Sum of Volumes and Partitioning (ASVP). Both methods are limited to fairly simple geometry, which makes them unsuitable for complex parts.

In 1994, Tseng and Joshi [20] took a different approach at volume decomposition that used a part's faces to split the negative (machined) space into small machinable volumes. These small volumes are then merged back together to form larger machinable features. This approach is limited in its ability to decompose all features (e.g., pocket with islands). Sakurai [15] developed a similar idea based on decomposition with convex edges, where maximal volumes are merged into machinable features. This approach expanded the possible volumes to include curved surfaces [16]. Since this can result in many feature interpretations, Woo and Sakurai [21] proposed

a “divide and conquer” approach in 2001 to optimize the machining feature selection. More recently, Fu et al., [3] developed a convex decomposition algorithm that uses concave edges to identify cutting planes. These cutting planes are then used to decompose the machined volume into feasible machining operations. The order of these operations is determined by heuristics. Other recent work has focused primarily on optimization methods to find the best order for machining features, which are defined at the onset of the search [7][8][17].

Many of the issues (e.g., combinatorial search space, complex features, and order of operations) faced in these process planning decomposition methods are similar to the issues faced in our approach, with two significant differences. First, we are decomposing the positive solid rather than the negative (machined) volume. Second, since our method is concerned with how to divide and rejoin subparts rather than machining, partial decompositions are feasible solutions. Consider the example in Fig. 2. The top, middle, and bottom rows are all feasible solutions. Depending on the material and assembly operations, it is likely one of the three decompositions in Fig. 2 is much nearer to optimal than a full decomposition (0% waste). Process planning, on the other hand, must fully decompose the negative geometry into machinable volumes to get a complete solution.

2.2. Other relevant research

This section discusses state of the art decomposition methods in closely related fields. These fields include additive manufacturing, hybrid manufacturing, and topology optimization.

Luo et al., [9] presented a decomposition algorithm based on additive manufacturing criteria (e.g. print volume, connector feasibility, aesthetics). Our search space for advanced joining is very similar and both methods implement a beam search. The primary difference and novelty of our work lies in the application of such methods to advanced joining and basing the objective function on production cost.

The hybrid manufacturing method presented by Kerbrat et al., [5] aims to decompose a part into additive and machined subparts. Their method splits the part wherever machining complexity is high. While machining complexity may indeed find good solutions and play a role in production cost, it is only an indicator for additive manufacturing. Our method is instead based on production cost, and is an open-ended hybrid manufacturing approach that could conceivably be expanded to consider many manufacturing processes and constraints.

Lastly, Saitou and Yetis developed a process to determine optimal assemblies for topology optimized

structural products [14]. Saitou’s continued work in the field has focused on topology optimization and assembly synthesis for 2D structural parts [13][22]. As work on this topic continues to advance to include 3D geometry, a merger between considerations in production cost from our model and structural integrity could provide even a more comprehensive design approach.

3. Approach

The method presented in this paper decomposes complex parts into assemblies with the goal of lowering overall production cost. To begin, the decomposition approach generates many unique assembly options by iterative applying cutting planes (Section 3.1). Next, an exhaustive list of manufacturing plans is created for each assembly option (Section 3.2). Then the cost of each of these assembly options is estimated (Section 3.3). Since there are many decomposition options and each option has a unique cost, a beam search optimization algorithm was applied (Section 3.4). The following sections describe each of these operations in further detail.

3.1. Decompose geometry

In this method, parts are decomposed by applying one cutting plane at a time, such that the resulting subparts can be assembled using advanced joining. To guarantee every part of an assembly can be accessed, the original part is decomposed with infinite flat planes. Cutting planes are applied one at a time, cutting the original part into alternative sets of subparts (Fig. 3 right 3 columns). Additional cutting planes are applied iteratively, adding one new plane in each iteration. This iterative approach allows each cutting planes to be mapped directly to an accessible assembly operation. In effect, each volume decomposition operation is the inverse of assembly operation(s). For example, plane C in Fig. 3 resulted in three subparts. For assembly, this maps to two joining operations, one for each of the legs.

The first task in decomposing a part is to select the possible cutting planes (Fig. 3 left). The ideal cutting planes reduce the waste material as much as possible. Some geometric features that indicate wasted material are protrusions and large holes. Such features, as well as less obvious options, can be found by selecting all the faces with adjacent concave edges. Using concave edges performs well in most cases, but Relevant Flats [11] is a more general approach. In this method, a cutting plane is added to every flat that will cut through the solid when that face is extended indefinitely. For the part in Fig. 3, this approach resulted in nineteen unique cutting planes. Relevant Flats was chosen because it is the fastest of the five

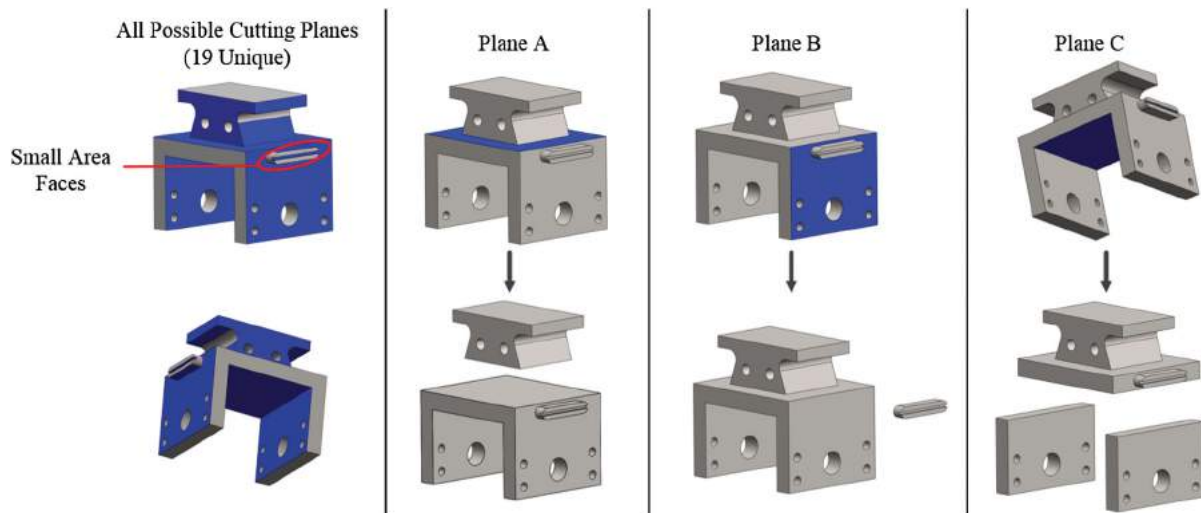


Figure 3. (left) All possible cutting planes, (right 3 columns) planes A, B, and C with resulting assembly option pictured below.

cutting plane identification methods presented in [11], however, for more complex parts Area Decomposition [11] and user feedback may be preferable.

In general, large faces that correspond to important part geometry produce better results than very small faces. For this reason, a cut off tolerance is applied as an area proportional to the total surface area of the original part. For the example on the left of Fig. 3, removing the small area faces from both sides of the part reduces the possible, unique cutting planes from nineteen to fifteen. While Relevant Flats is not comprehensive, it does provide a number of good cutting planes with little effort.

To identify these flats outside of a CAD platform, triangulated surfaces must be used. In this case, the input file is a .STL and face adjacency is added. To build up a flat surface, adjacent faces with the same normal are selected. This effectively turns the entire model into a collection of flat surfaces. The method then removes all surfaces that have an area less than the cut off tolerance.

Decomposing the part with cutting planes generates assembly options, such as the one shown in Fig. 4. The bidirectional arrows show references between parent and children subparts. If the subpart contains no children in the assembly, then it requires stock material. The boxed subparts are those that require stock material, while the unboxed subparts are intermediary steps. Each subpart contains geometric information including the tessellated solid, volume, and surface area for the subpart. The unidirectional arrows show assembly operations, where the arrow points toward the “base” subpart. The “base” subpart of each joining operation is the one with the larger area on the cutting plane, while the subpart(s) on the other side of the cutting plane are considered “joined” subpart(s).

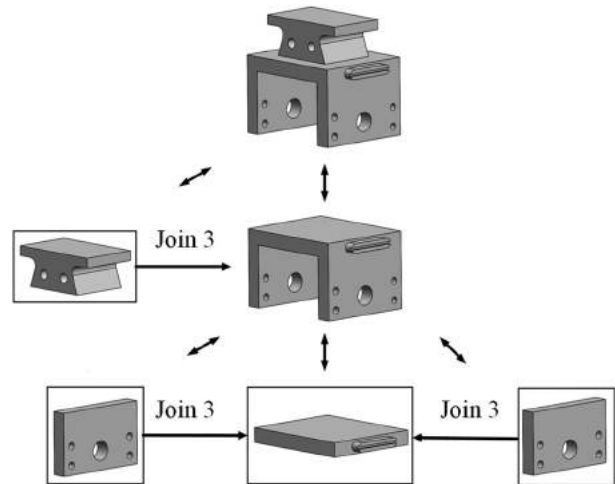


Figure 4. Assembly option generated by applying cutting plane C after cutting plane A from Fig. 3.

3.2. Generate manufacturing plans

To find the lowest cost manufacturing plan for an assembly option, every combination of stock materials and assembly operations is considered. Each set of “base” subpart, “joined” subpart, and joining operation is called a link. To allow a fast exhaustive search of all the available combinations, each link is assumed to be independent. This assumption is valid in most cases, since parts will need to be reset in the machines, but it does ignore some possible reductions in setup time.

Fig. 5 shows the six link options considered for a joining operation with two subparts that require stock material (Fig. 4 join 1). If a subpart is an intermediate step (subassembly), then there are fewer link options since it does not require new stock material. For example, in

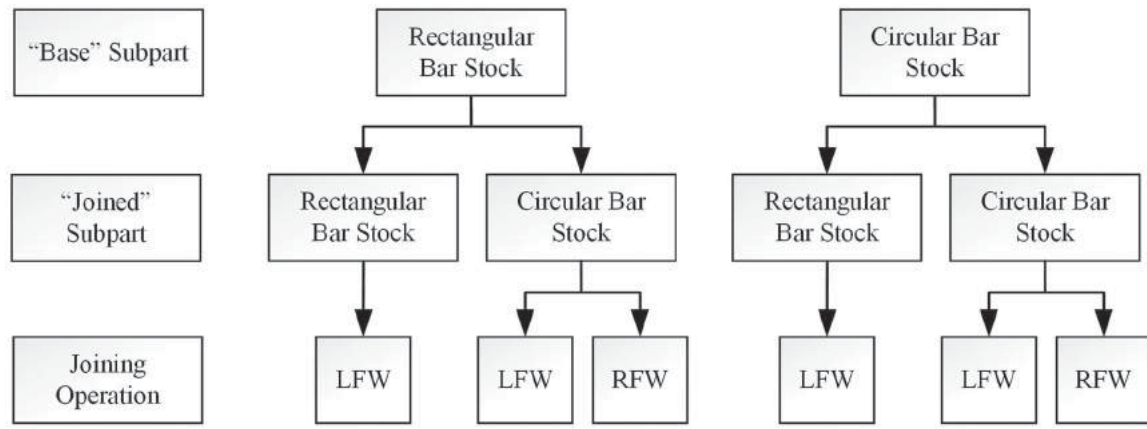


Figure 5. Exhaustive link options for a joining operation with two subparts requiring stock material.

Fig. 4 the resulting subassembly from join 1 is the “base” for join 2, so there are only three possible link options for join 2. The same goes for join 3. In the case where there is a join attaching two subassemblies, there are only two link options. Altogether, this means that 12 combinations of stock materials and joining operations ($6 + 3 + 3$) are considered for the simple example in Fig. 4. The lowest cost manufacturing plan will be the one with the lowest cost link for each joining operation (Eqn. 3.1 in Section 3.3).

3.3. Estimate cost

Production cost was chosen as the objective function because it is the primary driver in industry decisions and because most other factors (including energy usage and carbon emissions) can be monetized. However, the cost estimation model presented in this paper is just a framework for more detailed cost models, and therefore it only includes the three major assembly cost factors; material, machining, and joining cost. There are other high level factors that could be included, such as the cost of heat treatment, inspection, and material movement, but in general those costs are less than the three major factors. While these additional costs were not included presently, they could be included in the future. In addition, the cost estimation model in this method does not need to reflect the total cost of production costs, but rather the relative production costs to inform decisions between assembly options. For instance, heat treatment, boring, surfacing, and other post rough machining operations may be assumed to be the same for all assembly options, and therefore have no influence on the decision between options. Likewise, machining setup and unload cost is not included since machining is assumed to be after all the assembly operations. Focusing on cost differences allows the cost models to be simpler and require less geometric information.

The production cost assigned to an assembly is equal to the minimum manufacturing plan cost, which is equal to the sum of the minimum cost links for each joining operation (Eqn. 3.1). The cost of each link is the sum of the costs of its base subpart, joined subpart, and assembly operation (Eqn. 3.2). The cost of each subpart is the cost of its material plus machining (Eqn. 3.3), but if the subpart is an intermediate step (subassembly) then it has a cost of zero. Although the machining cost is included in the cost of each link, the machining operation is not performed till after the part is assembled. This helps choose the minimum cost link independent of the other links. It does not affect overall cost, since setup/unload time is not included in the machining cost.

$$\begin{aligned} Cost_{Assembly} &= MinCost_{Manufacturing\ Plan} \\ &= \sum_{i=1}^{Number\ of\ Joins} Min(Cost_{Link\ i}) \end{aligned} \quad (3.1)$$

$$\begin{aligned} Cost_{Link} &= Cost_{Base\ Subpart} + Cost_{Joined\ Subpart} \\ &\quad + Cost_{Assembly\ Operation} \end{aligned} \quad (3.2)$$

$$Cost_{Subpart} = Cost_{Material} + Cost_{Machining} \quad (3.3)$$

The method currently assumes that the stock material available is either circular or rectangular bar stock of any size, and that the stock material will be joined along a flat plane (e.g., a cutting plane). Because of these assumptions, the smallest bar stock volume can be determined using 2D geometric algorithms (minimum bounding circle and minimum bounding rectangle) and the depth. The depth is the length between the cutting plane and furthest vertex of that subpart. The top row in Fig. 6 shows the depth, area, and rectangular stock volume, while the bottom row shows the same information for the circular stock volume. The material cost is simply the price of material (USD/mass) times the mass of the part (Eqn. 3.4).

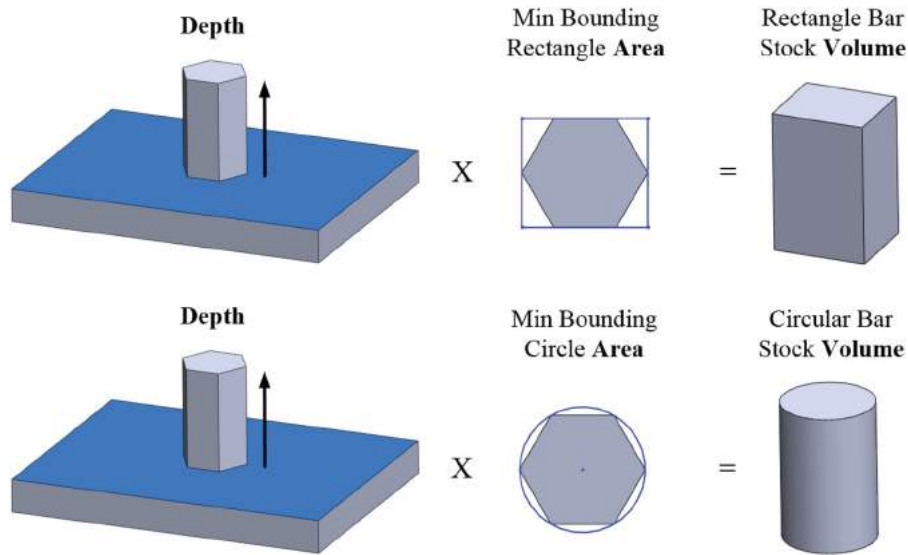


Figure 6. Subpart geometric information for rectangular and circular bar stock.

$$Cost_{Material} = Price_{Material} * Volume_{Stock} * Density_{Material} \quad (3.4)$$

The cost of machining is the cost rate to run a machine for a factory ($R_{Machine}$) multiplied by the time the machine is being used ($t_{Machining}$) (Eqn. 3.5). The cost rate aims includes the capital cost of the machine, operation costs, plant overhead, fixtures, tooling, and labor. The time of machining could also include other factors such as setup time, quantity, and non-operation time, but for simplicity, the time of machining is calculated here as the removed volume over the material removal rate (MRR) (Eqn. 3.6). The total volume being removed is the difference between the stock material and the final volume (Eqn. 3.7), and the MRR is assumed to be a given value for rough machining. Finish machining and drilling are not included because they are assumed to have little to no impact between competing assemblies.

$$Cost_{Machining} = R_{Machine} * t_{Machining} \quad (3.5)$$

$$t_{Machining} = Volume_{Removed} / MRR \quad (3.6)$$

$$Volume_{Removed} = Volume_{Stock} - Volume_{Final} \quad (3.7)$$

The cost of each assembly operation is highly dependent on the type of assembly processes that the user has decided to include. Gas welding, for example, is dependent on the material and the perimeter of the weld, whereas RFW is dependent on the mass and the area of the weld. Other assembly operations may require additional fasteners (e.g., bolts or rivets) or features on the parts (e.g., flanges or threads). Therefore, the cost of each assembly operations should be determined by its own unique cost model.

This paper focuses on LFW and RFW, which are both cost dependent on the area of the weld. The case studies presented in Section 4 use the simple assembly operation cost model shown in Eqn. (3.8), where $R_{Assembly}$ is the cost rate per area and $Fixed_{Assembly Cost}$ is a fixed minimum assembly cost. The area used in Eqn. (3.8) is the contact area on the cutting plane where the two subparts come together. The fixed cost is included to avoid assembly operations that join very small features, such as fillets.

$$Cost_{Assembly Operation} = R_{Assembly} * Area + Fixed_{Assembly Cost} \quad (3.8)$$

This equation does not include the capital cost of the machine or the tooling involved, those costs are often the barrier to entry for friction welding, but they do not greatly affect the cost difference between alternatives. Capital costs should be considered when comparing the savings of using friction welding to the current cost of production. This is a business level comparison and beyond the scope of this paper.

3.4. Optimization

3.4.1. Search space

Understanding the assembly option search space is crucial for determining an effective optimization algorithm. In general, the search space is non-monotonic, dependent on order of operations, and very large. This section discusses these characteristics in further detail. All of the findings in Section 3.4.1 are new contributions to defining this search space.

The search space is non-monotonic because the cost of each further decomposed assembly is not guaranteed to

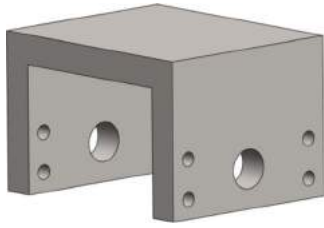


Figure 7. Part illustrating monotonicity.

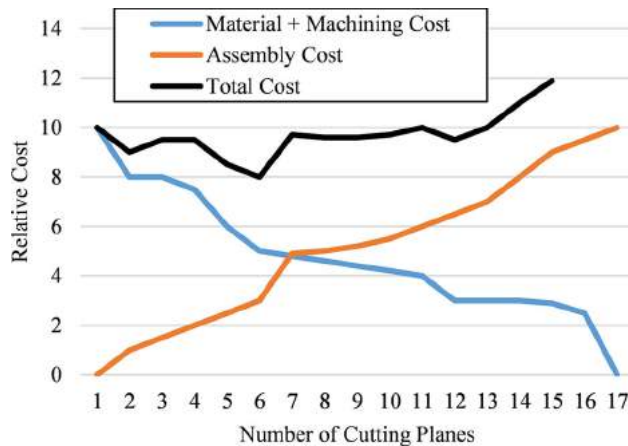


Figure 8. Graph showing an example of the monotonicity of cost terms.

be less expensive than its parent. It is only less expensive if the savings in material cost and machining outweigh the increase in assembly costs. Furthermore, even if an assembly is more expensive than its parent, there is no

guarantee that the assembly's children will also be more expensive. This can be seen when considering the faces on the inside of each of the legs of the part in Fig. 7. If one inside face on either side is chosen as a cutting plane, it will not reduce the overall bounding box until the other face is also chosen as a cutting plane.

In fact, the only monotonicity that can be guaranteed is that the cost of material and machining are strictly monotonically decreasing. For the vast majority of operations, the cost of assembly operations is weakly monotonically increasing. However, this is not guaranteed because the area of the weld may decrease as the subparts are further decomposed to closer net shapes. An example of the total cost, as shown by the black line in Fig. 8, may increase and decrease several times as a function of the number of cutting planes.

Additionally, the order that the cutting planes are applied is important for two reasons. First, two cutting planes applied in different order can result in geometrically different assemblies (Fig. 9 top row). This occurs whenever two cutting planes intersect inside the solid (Fig. 9 top left), but it can also occur when a cutting plane creates more than two subparts or when cutting on one plane makes the other invalid (Fig. 9 bottom row). Note how in the bottom right of Fig. 9, plane A will no longer cut the right subpart into smaller pieces or reduce the wasted volume of the left subpart. Second, since the resulting assembly order is different, the fixtures and machines required are different between even two geometrically equal assemblies. Since the order of

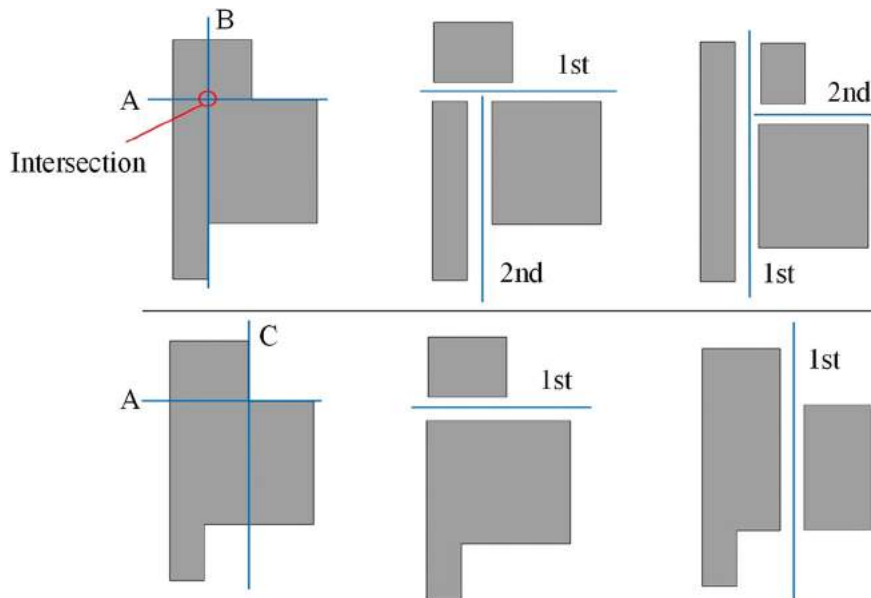


Figure 9. (top row) Ordering planes A and B differently results in different subparts (compare middle to right), (bottom row) when either plane A or C is chosen, the other becomes invalid.

cutting planes changes the assembly plan and overall production costs, order must be captured in the search space.

The theoretical maximum number of possible assembly options is based on the number of cutting planes (“s”). After one cutting plane cut is applied, there is the possibility for “s - 1” additional cutting planes and so on, making the maximum depth “s.” Fig. 10 shows the search space for a model with only three possible cutting planes (s = 3). Each circle represents a unique assembly option, resulting in a maximum of sixteen unique assembly options. Summing up the number of options, results in the geometric series shown in Eqn. (3.9). Now consider the complex part in Fig. 3, which has fifteen unique cutting planes after eliminating the small faces (s = 15). Eqn. (3.9) results in over 3.5 trillion (3.5×10^{12}) assembly options. However, the actual maximum number of possible assembly options is much less than the theoretical maximum because certain planes make other planes invalid (Fig. 9). Nevertheless, an AI search is required because an exhaustive search is impractical when considering the size of the search space.

$$\text{Maximum Possible Assemblies} = s!(1 + \sum_{k=1}^s \frac{1}{k!}) \quad (3.9)$$

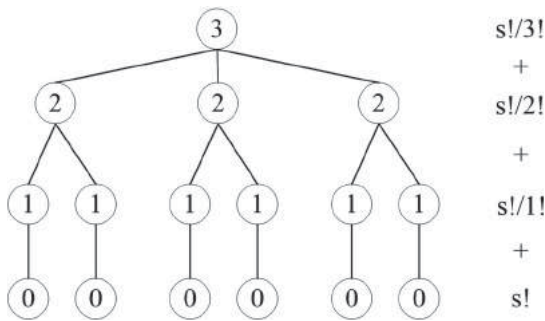


Figure 10. Search space for a model with three possible cutting planes. Each circle represents a unique assembly option. The number inside shows the number of possible cutting planes remaining.

3.4.2. Optimization algorithm

There were four primary factors in determining the best optimization algorithm to implement. First, the search space is organized as a decision tree (e.g., Fig. 10). Second, the design space is inherently non-monotonic. Third, every option in the design space is a potential solution. Fourth, the design space is very large, so large that it is not realistic to search every option.

A variety of optimization algorithms were considered, but ultimately beam search [12] was chosen as the best algorithm for these primary factors. Exhaustive search is impractical because of the size of search space. Greedy search is quick, but does not explore many options. A* could be reasonable if a fairly restrictive admissible heuristic could be found, however, such a heuristic has so far eluded the researchers. A genetic algorithms could be modified to include the order of cutting planes, but it would greatly complicate the search structure and assembly option representations. Lastly, beam search is easy to implement, adjust, and in the future, parallelize. Though it does not guarantee that an optimal solution will be found, no other method could make this guarantee in a practical timespan. Beam search is a powerful approach that intelligently limits time and memory while considering a broad range of decompositions, and it tends to converge on good results quickly.

Beam search works like a breadth first (iterative) search, except that it only keeps the best performing candidates from each iteration. All other candidates are deleted. The beam width is a user-defined input parameter that adjusts the number of top candidates that are kept. In this way, it limits the design space and memory usage. Generally, a large beam width is used for beam search, but for the two case studies shown in Section 4, a beam width of ten (b = 10) was implemented. Larger beam widths, up to one hundred, were also tested, but produced the same near optimal assembly as a width of ten.

During the search, every identified cutting plane is considered at each iteration, assuming it is still feasible and has not already been used. For example, in Fig. 11 solution B, two subparts are formed. During the second iteration (2 cutting planes), every valid cutting plane except for the one used already will be applied to solution B to generate new options. In most cases, those cutting planes will only be valid (cut through the subpart) for one of the two subparts. This repeats at each iteration, with the assembly option considering every cutting plane it is not already using. Although beam search is not exhaustively searching all possible combinations of cutting planes, it is searching many options and does not eliminate a cutting plane because of poor performance early in the search.

The beam search algorithm, applied in this research, is shown in Fig. 12. The input is the original part geometry, which acts as the first assembly option for the while loop. The output of this loop is a sorted list of decompositions and their assembly plans. The first and second **foreach** loops decompose every subpart of every current assembly with the **DecomposeVolume** function. This function performs volume decomposition for every valid cutting plane on that subpart, saving a set of children

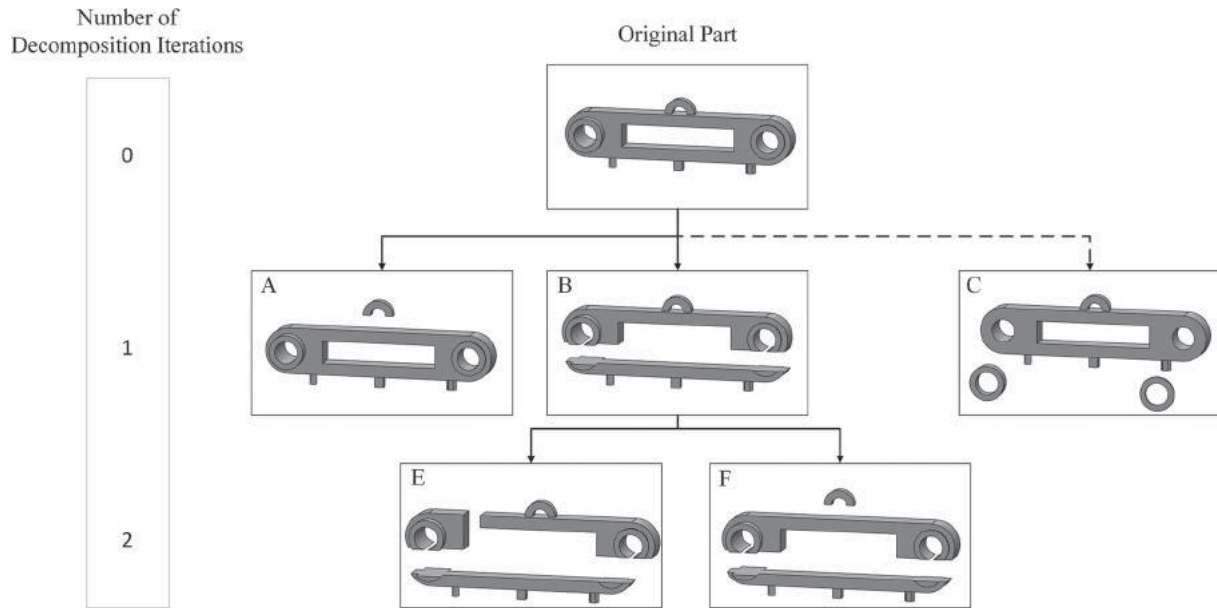


Figure 11. Example of a part being decomposed by iteratively applying cutting planes.

```

input: O is the original complex part, b is the beam width
output: Lowest Cost Assemblies

currentAssemblies += O
allAssemblies += O
minCost = EstimateCost(O)
while ContainsAny(currentAssemblies)
  foreach Assembly ( $\psi$ ) in currentAssemblies
    foreach StockComponent ( $\beta$ ) in Assembly
      if ( $\beta$  has no children)
        ChildSets = DecomposeVolume( $\beta$ )
        foreach ChildSet ( $\phi$ ) in ChildSets
          newAssembly( $\psi'$ ) = CreateNewAssembly( $\psi, \beta, \phi$ )
           $\psi'$ .Cost = EstimateCost( $\psi'$ )
          if ( $\psi'$ .Cost < minCost)
            minCost =  $\psi'$ .Cost
          if ( $\psi'$ .Cost -  $\psi'$ .Potential  $\leq$  minCost)
            newAssemblies +=  $\psi'$ 
        currentAssemblies = HighestRanked(newAssemblies, b)
  allAssemblies += currentAssemblies
return HighestRanked(allAssemblies)

function: ChildSets = DecomposeVolume( $\beta$ )
  CuttingPlanes = FindCuttingPlanes( $\beta$ )
  foreach CuttingPlane in CuttingPlanes
    ChildSet = CutAlongPlane( $\beta$ , CuttingPlane)
    ChildSets += ChildSet
  return ChildSets

```

Figure 12. Psuedo code for the applied beam search algorithm.

subparts (ChildSet) for each cutting plane. The third **foreach** loop creates a new assembly (ψ') based on each ChildSet, estimates its cost, and saves the results if it

has the potential to be better than the minimum cost assembly created thus far.

The maximum possible improvement in cost from the current state determines an assembly's potential savings (Eqn. 3.10). The maximum improvement to material cost is equal to the cost of the waste material. The maximum improvement to machining is that no machining is required. Since an additional decomposition will require a minimum of one assembly operation, the fixed cost is subtracted from the potential. This potential creates an admissible heuristic that creates a lower bound on the number of iterations, instead of using an artificial depth limit.

$$Potential = Cost_{Waste} + Cost_{Machining} + Fixed_{Assembly\ Cost} \quad (3.10)$$

After all the current assemblies have been decomposed, the candidates are ranked and trimmed to the top ten ($b = 10$). These top ranked assemblies are sent through the loop again to decompose them with another cutting plane. The outer **while** loop continues to run, until there is no further improvement, which occurs when no new assemblies are added to the newAssemblies list in that iteration.

4. Results

Two case studies were run through the method presented in this paper to demonstrate its value. In the first case study, a more complex part is used to show the value of this method on real world parts. The second case study considers four materials to show how the method

performs for different parameters. For both the case studies, the only stock materials considered are rectangular and circular bar stock of arbitrary size. Also, the only assembly operations considered are linear friction welding (LFW) and rotary friction welding (RFW), assuming Eqn. (3.8) for both.

The decomposition function, manufacturing plan generation, cost estimation, and beam search algorithm, all presented in Section 3, were developed as a standalone executable in C#. It is independent of any CAD software, and instead, the basic geometry functions (e.g., cut solid, minimum bounding rectangle) are implemented in a new geometry library. The tests were run using a computer with a 3.4 GHz processor and 12 GB of RAM.

Tab. 1 shows the input values used in this case study. The price and MRR for the various materials are based on industry values. These values will be different for different machines, cutting tools, and part sizes. As such, these values are meant to be used to show a comparison of different costs for materials, not an exact cost for a particular situation. It was assumed that the same machine can be used for the four materials and that the assembly costs are not based on the material. Therefore, the assembly and machine rates, along with the fixed cost of an assembly operation, were the same for all four materials. The rate of the machine and assembly costs are rough estimations, chosen so that they were an appropriate percentage of the overall production cost.

4.1. Case Study 1

The first case study part (Fig. 13) is a complex part intended to show the value of this method on real world parts. Its bounding dimensions are approximately

Table 1. Values for input variables used in case studies.

Material	Aluminum 6061	Stainless Steel 304	Ti6Al4V	Inconel 718
Density [g/cm ³]	2.70	8.0	4.43	8.19
Price [USD/kg]	4	6	18	60
MRR [cm ³ /min]	800	16	5	2
R_{machine} [USD/hr]	100	100	100	100
R_{Assembly} [USD/cm ²]	5	5	5	5
F_{Assembly} [USD]	25	25	25	25

93 cm × 23 cm × 15 cm. If machined from rectangular bar stock of that size, there is only 18% material utilization (82% of material is machined away). A common titanium (Ti6Al4V) is chosen as the material (see Tab. 1 for input values).

A cut off tolerance (0.125% × surface area) was used to identify large flat surfaces for cutting planes using the Relevant Flats method. This method identified 28 cutting planes, resulting in over 8 octillion (8×10^{29}) possible assembly options (Eqn. 3.9). An exhaustive search of this space is infeasible without additional restrictions, but even with cost (Eqn. 3.10) and plane interactions (Fig. 9) eliminating much of that space, an exhaustive search is prohibitively time consuming. So to begin assembly synthesis, the model is loaded into the beam search algorithm (Fig. 12).

Fig. 14 shows the cost comparison of the original “hog out” (machined from single piece of rectangular bar stock) production cost versus the assemblies that were created during decomposition. The dots represent the top ten assemblies that were decomposed further at each depth. The best solution is at a depth of three, but the beam search extends well past this solution. This is because the current approach continues decomposition as long as there is potential improvement (Eqn. 3.10). While this does seem extraneous, it does guarantee that the beam search is not stopped prematurely. Past depth 4, the assemblies are very similar because they have

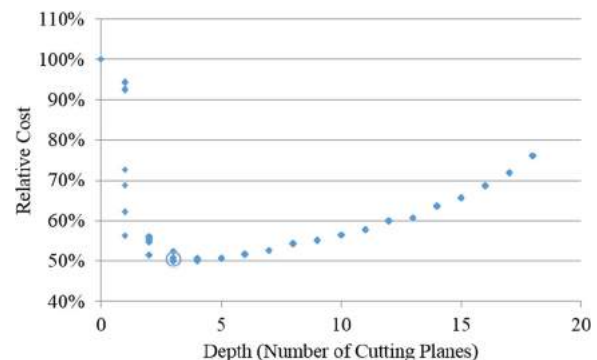


Figure 14. Case Study 1 comparison of “hog out” production cost to the assembly options.

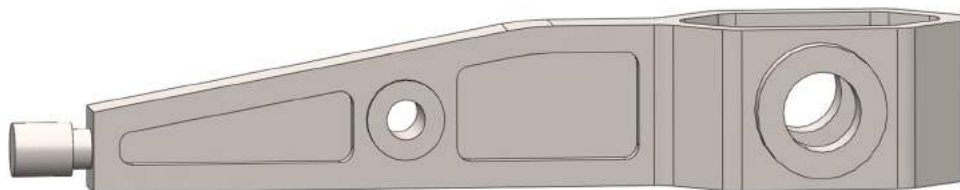


Figure 13. Structural support beam.

a similar lineage. Overall, 5,830 assembly options were created and evaluated in 78 minutes.

Fig. 15 shows the best assembly option for Case Study 1. The boxed subparts are the stock material, while unboxed subparts are intermediary steps. Each assembly operation is shown as a single arrow, pointing from the “joined” subpart to the “base” subpart. In this example, LFW was chosen for three assembly operations and

RFW was chosen for the other. The final operation is a machining operation.

Fig. 16 shows the production cost comparison of the original “hog out” versus the best assemblies for three depths. The best assembly solution is at depth 3, and it resulted in a 50% cost reduction and a 65% waste reduction. All three depth solutions show that the savings in material and machining cost more than make up for the

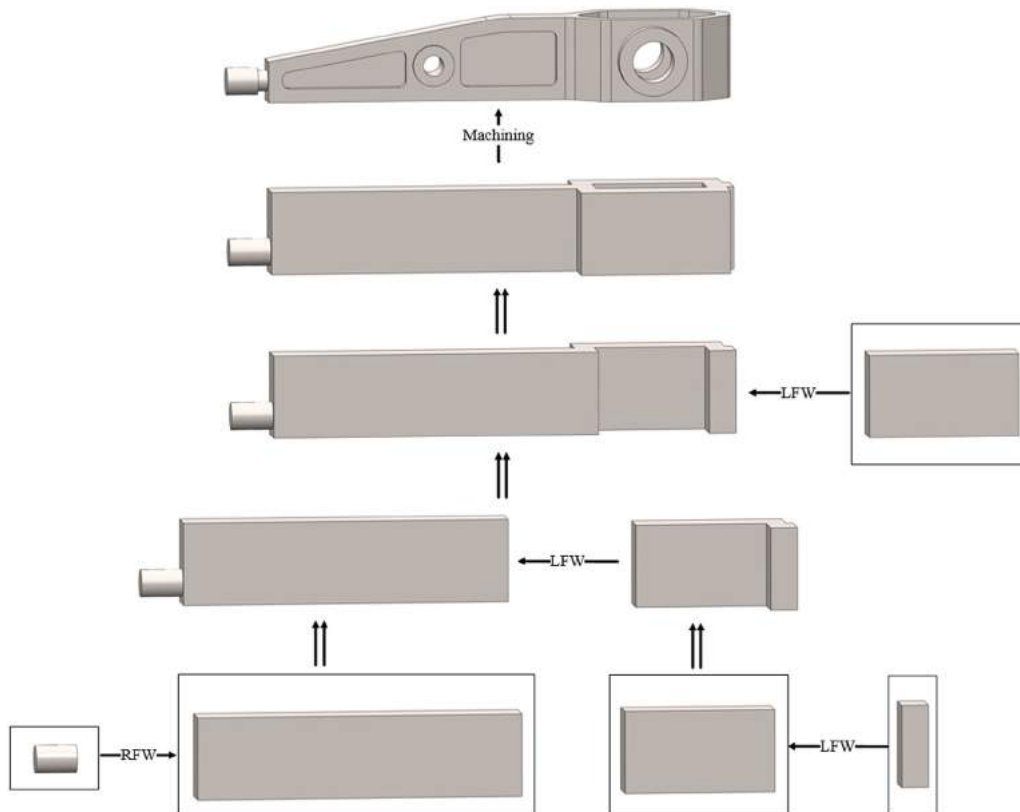


Figure 15. Case Study 1 best assembly option for Ti6Al4 V.

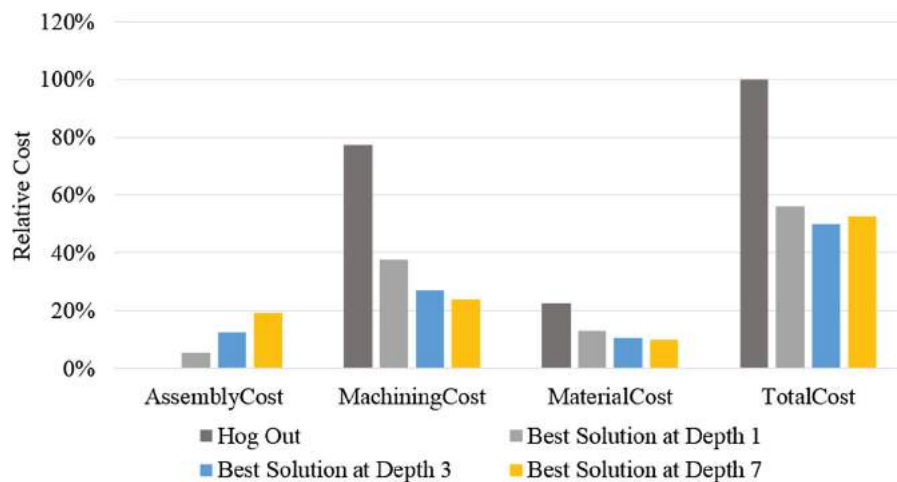


Figure 16. Case Study 1 production cost comparison of “hog out” and the best assemblies from depth 1, 3, and 7.

added assembly cost. Nevertheless, too many assembly operations eventually drive up the cost because there is less material and machining to save at each successive depth. Finding this balance in assembly vs. machining is one reason the automated method is so powerful.

4.2. Case Study 2

The second case study part (Fig. 17) is a smaller component intended to show the effect of material to the best assembly option. Its bounding dimensions are approximately 25 cm × 36 cm × 30 cm. If it is machined from rectangular bar stock of that size, there is only 32% material utilization. A common aluminum (6061), stainless steel (304), titanium (Ti6Al4V), and Inconel (718) were chosen for their variation in material cost and machinability (see Tab. 1 for input values).

A cut off tolerance ($0.125\% \times \text{surface area}$) was used to identify large flat surfaces for cutting planes using the Relevant Flats method. This method identified 15 cutting planes, resulting in 3.5 trillion maximum possible assembly options (Eqn. 3.9). An exhaustive search was attempted with both cost potential and plane interactions eliminating many options. For 8 cutting planes, an exhaustive search took 14 minutes, but with each added plane there is an exponential increase in time. For 15 cutting planes, it was estimated to take 60 days to finish the exhaustive search. The rest of the results for this case study are from loading the part into the beam search algorithm (Fig. 12).

Fig. 18 shows the cost comparison of the original “hog out” (machined from single piece of rectangular bar stock) production cost versus the minimum assembly cost at each depth. The best decomposition for each material is circled in Fig. 18. Notice the vast difference in assembly options for the different materials. Aluminum is a relatively inexpensive and machinable material, so the best solution is to machine it from a single block of stock material. Inconel and Titanium, on the other hand, are both expensive and difficult to machine thus resulting in multiple cutting planes. Stainless Steel fell in the middle.

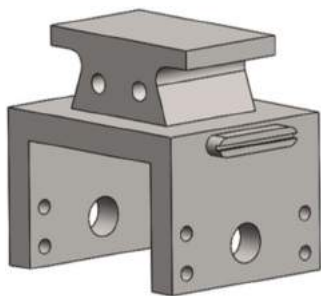


Figure 17. Square support bracket.

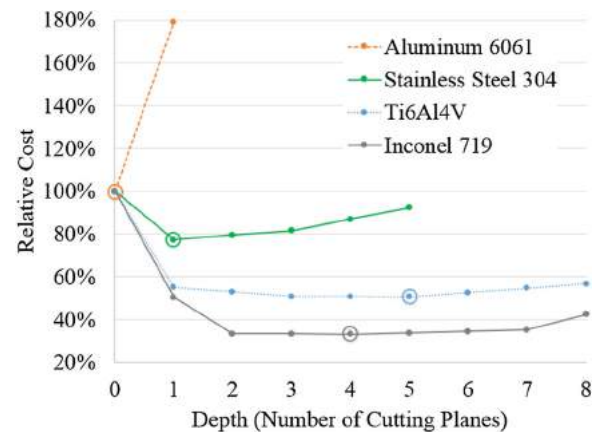


Figure 18. Case Study 2 comparison of “hog out” production cost to the assembly options considered for Ti6Al4V.

Fig. 19 shows the best assembly options for the Inconel and Titanium. The boxed subparts are the stock material, while unboxed subparts are intermediary steps. Each joining operation is shown as a unidirectional arrow, pointing from the “joined” subpart to the “base” subpart. In this example, LFW was chosen for every assembly operation. The final operation is a machining operation. Stainless Steel had a simple solution, where the two legs were joined onto the main body with LFW. This solution resulted in a 20% reduction in production cost and 67% reduction in waste material. Inconel, on the other hand, resulted in a 67% reduction in production cost and 92% reduction in waste material. Unfortunately, LFW is not as matured for Inconel as for other metals, but this result shows the need. Titanium resulted in a 49% reduction in production cost and 91% reduction in material waste. Aluminum did not benefit from advanced joining. These results show that decomposing complex parts into assemblies can significantly reduce the cost of production and material waste for a variety of metals.

It is important to note, that while the results from Case Study 2 show that the best solutions occur at depth 4 and depth 5 for Inconel and Titanium respectively, the solutions at shallower depths may be preferable to the designer for other factors (e.g., processing time, machine availability, and technology maturity). In making the software tool automated, the designer can look at all the output solutions and quickly identify which solution is most feasible and cost effective.

5. Limitations and future work

This section discusses the current state of research and the limitations of the approach presented in this paper.

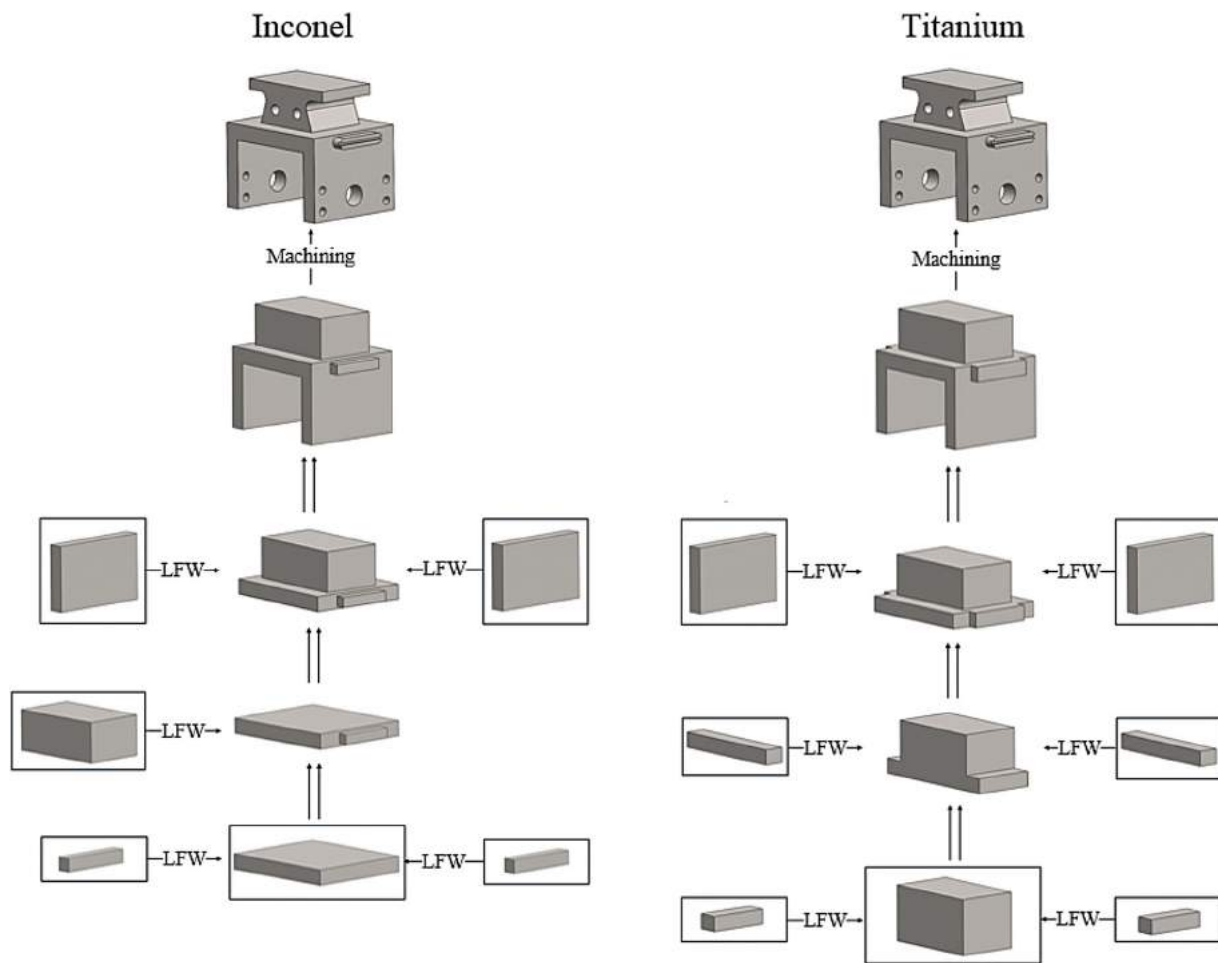


Figure 19. Case Study 2 best assembly options for the different materials.

The limitations can be split up into four primary categories; geometry, material, optimization, and manufacturing.

In regards to the geometry, there are two limitations to the cutting plane identification applied in this paper. First, the cutting planes are restricted to flat infinite planes to guarantee assembly operations will be possible. It may, however, be advantageous to consider curved or partial cutting planes. Second, while the cutting plane identification does find many good planes, it does not catch all the reasonable possibilities. This can especially be seen when considering fillets and curved features, which the current method ignores. Future work aims to include identification of ideal cutting planes on these complex curved shapes.

The method presented in this research does not consider loading conditions for two reasons. First, the material properties for friction welded joints have mechanical properties similar to their parent materials [2][4], so there is little effect on part performance with just a few welds. Second, including a FE analysis as a secondary constraint to cost is impractical for this already

expansive search process. Nevertheless, since friction welding allows joining of dissimilar materials, such as aluminum to titanium [10], including some information about loading could be valuable.

The major limitation to the optimization algorithm presented in this paper is that it can get trapped in locally optimal solutions. There are two ways in which the current method attempts to guard against this. First, it applies a reasonably sized beam width for this problem. This helps to explore a variety of options, but does not prevent the search from getting trapped in locally optimal solutions. For example, during the second iteration, the beam width could eliminate all child solutions from parents other than the local optimal parent, if the local optimal parent's children are best. Second, the potential cost savings (Eqn. 3.10) are used as the exit criteria rather than a depth limit. This prevents the search from ending at a premature depth. The primary issues effecting the optimization are the time and memory it takes to create and evaluate each feasible option. Beam search is an adjustable compromise of time, memory, and accuracy. While we believe it was the best optimization algorithm

to begin with, we encourage future work on alternate algorithms.

On the manufacturing front, there are four simplifications in this work. First, only rectangular and circular bar stock are considered. In reality, there are many alternatives. A minor adjustment is to include other shapes like tube and L-channel extrusions. Perhaps a more thorough approach would be to include water jetting, additive manufacturing, forging, casting, and other forming processes. These additional forming processes can be included by sending the cost model additional information and implementing a separate optimization search on the cost model. Second, only linear and rotary friction welding were included as assembly options. Additive manufacturing, friction stir welding, and traditional assembly operations may be advantageous to add. Third, the current method does not include tolerances to account for joining or machining. Fourth, a more detailed cost model could be implemented to provide more accurate results. These simplifications have been considered, and as such, the current method was developed to accommodate future work.

6. Conclusion

This research lays the ground work for future exploration of assembly options with advanced joining based on production cost. First, a new method to decompose complex parts into assemblies using advanced joining is presented. The decomposition approach (Section 3.1) and manufacturing plan generation (Section 3.2) are the first of their kind to consider advanced joining with LFW and RFW. Second, the search space is explored and defined (Section 3.4.1). This contribution is significant, since it dictates which optimization methods are reasonable. Lastly, an optimization method based on beam search is presented for this search space. While the beam search algorithm (Section 3.4.2) is not novel and has been applied in similar decomposition searches [9], this paper is the first to consider the size and characteristics of this particular assembly option space. Altogether, this method automatically generates alternative decompositions and compares their relative costs with little effort from the design engineer. In addition, it considers many more alternatives than the designer could reasonably consider and has the potential to be integrated with a highly detailed cost model.

Lastly, the two case studies show this method applied to two complex parts. The first case study shows an example of the beam search algorithm results and the production cost components in detail. The second case study, which considered four different metals, shows that this method is more beneficial for expensive materials,

but could also be used to economically reduce waste for less expensive materials. Both case studies clearly show the advantage of beam search, when compared to an exhaustive search. This advantage is expected to exponentially increase as the parts become more complex and the objective function evaluations are improved.

Acknowledgements

The research work presented in this paper was supported by the Boeing Company and the Oregon Metal Initiative (OMI). Views expressed in this paper are those of the authors and do not necessarily reflect the opinions of these supporters. The authors would like to acknowledge the larger project team, consisting of Prof. Karl Haapala, Prof. David Kim, Steven Lindberg, and Harsha Malshe, who were invaluable in the development of this concept and its implementation. In addition, the authors would like to thank Matt Carter, Stefanie Meier, Kevin Slattery, Eric Eide, Ryan Hanks, and Chris Carpenter for their guidance.

ORCID

Brandon R. Massoni  <http://orcid.org/0000-0002-0946-5205>

Matthew I. Campbell  <http://orcid.org/0000-0003-1296-6542>

References

- [1] Boothroyd, G.; Radovanovic, P.: Estimating the Cost of Machined Components During the Conceptual Design of a Product, *CIRP Ann. - Manuf. Technol.*, 38(1), 1989, 157–160. [http://dx.doi.org/10.1016/S0007-8506\(07\)62674-2](http://dx.doi.org/10.1016/S0007-8506(07)62674-2)
- [2] Bhamji, I.; Preuss, M.; Threadgill, P. L.; Addison, A. C.: Solid state joining of metals by linear friction welding: A literature review, *Material Science Technology*, 27(1), 2011, 2–12. <http://dx.doi.org/10.1179/026708310X520510>
- [3] Fu W.; Eftekharian, A. A.; Campbell, M. I.: Automated Manufacturing Planning Approach Based on Volume Decomposition and Graph-Grammars, *Journal of Computing and Information Science in Engineering*, 13(2), 2013. <http://dx.doi.org/10.1115/1.4023860>
- [4] Grujicic, M.; Arakere, G.; Pandurangan, B.; Yen, C.-F.; Cheeseman, B. A.: Process Modeling of Ti-6Al-4 V Linear Friction Welding (LFW), *Journal of Materials Engineering and Performance*, 21(10), 2012, 2011–2023. <http://dx.doi.org/10.1007/s11665-011-0097-8>
- [5] Kerbrat, O.; Mognol, P.; Hascoët, J.-Y.: A new DFM approach to combine machining and additive manufacturing, *Computers in Industry*, 62(7), 2011, 684–692. <http://dx.doi.org/10.1016/j.compind.2011.04.003>
- [6] Kim, Y. S.: Recognition of form features using convex decomposition, *Computer-Aided Design*, 24(9), 1992, 461–476. [http://dx.doi.org/10.1016/0010-4485\(92\)90027-8](http://dx.doi.org/10.1016/0010-4485(92)90027-8)
- [7] Li, X.; Gao, L.; Wen, X.: Application of an efficient modified particle swarm optimization algorithm for process planning, *International Journal of Advanced Manufacturing Technology*, 67(5), 2013, 1355–1369. <http://dx.doi.org/10.1007/s00170-012-4572-7>

- [8] Liu, X.-J.; Yi, H.; Ni, Z.-H.: Application of ant colony optimization algorithm in process planning optimization, *Journal of Intelligent Manufacturing*, 24(1), 2013, 1–13. <http://dx.doi.org/10.1007/s10845-010-0407-2>
- [9] Luo, L.; Baran, I.; Rusinkiewicz, S.; Matusik, W.: Chopper: Partitioning Models into 3D-Printable Parts, *ACM Transactions on Graphics*, 31(6), 2012, 1–9. <http://dx.doi.org/10.1145/2366145.2366148>
- [10] Maalekian, M.: Friction welding – critical assessment of literature, *Science and Technology of Welding and Joining*, 12(8), 2007, 738–759. <http://dx.doi.org/10.1179/174329307X249333>
- [11] Massoni, B; Campbell, M.I.: Optimizing Cutting Planes for Advanced Joining and Additive Manufacturing, *ASME 2016 International Mechanical Engineering Congress and Exposition*, 2, 2016. <http://dx.doi.org/10.1115/IMECE2016-67495>.
- [12] Norvig, P.: *Paradigms of Artificial Intelligence Programming: Case Studies in Common LISP*. Morgan Kaufmann, San Francisco, California, 1992.
- [13] Saitou, K.; Lyu, N.: Topology Optimization of Multi-component Beam Structure via Decomposition-Based Assembly Synthesis, *Journal of Mechanical Design*, 127(2), 2005, 170–183. <http://dx.doi.org/10.1115/1.1814671>
- [14] Saitou, K.; Yetis, A.: Decomposition-Based Assembly Synthesis Based on Structural Considerations, *Journal of Mechanical Design*, 124(4), 2002, 593–601. <http://dx.doi.org/10.1115/1.1519276>
- [15] Sakurai, H.: Volume decomposition and feature recognition: Part 1—polyhedral objects, *Computer-Aided Design*, 27(11), no. 11, 1995, 833–843. [http://dx.doi.org/10.1016/0010-4485\(95\)00007-0](http://dx.doi.org/10.1016/0010-4485(95)00007-0)
- [16] Sakurai, H.; Parag, D.: Volume decomposition and feature recognition: Part II: curved objects, *Computer-Aided Design*, 28(6), 1996, 519–537. [http://dx.doi.org/10.1016/0010-4485\(95\)00067-4](http://dx.doi.org/10.1016/0010-4485(95)00067-4)
- [17] Salehi, M.; Tavakkoli-Moghaddam, R.: Application of genetic algorithm to computer-aided process planning in preliminary and detailed planning, *Engineering Applications of Artificial Intelligence*, 22(8), 2009, 1179–1187. <http://dx.doi.org/10.1016/j.engappai.2009.04.005>
- [18] Tang, K.; Woo, T.: Algorithmic aspects of alternating sum of volumes. Part 1: Data structure and difference operation, *Computer-Aided Design*, 23(5), 1991, 357–366. [http://dx.doi.org/10.1016/0010-4485\(91\)90029-v](http://dx.doi.org/10.1016/0010-4485(91)90029-v)
- [19] Tang, K.; Woo, T.: Algorithmic aspects of alternating sum of volumes. Part 2: Nonconvergence and its remedy, *Computer-Aided Design*, 23(6), 1991, 435–443. [http://dx.doi.org/10.1016/0010-4485\(91\)90011-k](http://dx.doi.org/10.1016/0010-4485(91)90011-k)
- [20] Tseng, Y.-J.; Joshi, S. B.: Recognizing multiple interpretations of interacting machining features, *Computer-Aided Design*, 26(9), 1994, 667–688. [http://dx.doi.org/10.1016/0010-4485\(94\)90018-3](http://dx.doi.org/10.1016/0010-4485(94)90018-3)
- [21] Woo, Y.; Sakurai, H.: Recognition of maximal features by volume decomposition, *Computer-Aided Design*, 34(3), 2002, 195–207. [http://dx.doi.org/10.1016/s0010-4485\(01\)00080-x](http://dx.doi.org/10.1016/s0010-4485(01)00080-x)
- [22] Yildiz, A. R.; Saitou, K.: Topology Synthesis of Multicomponent Structural Assemblies in Continuum Domains, *Journal of Mechanical Design*, 133(1), 2011. <http://dx.doi.org/10.1115/1.4003038>