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

DETC2018-85646

GRAPH PARTITIONING TECHNIQUE TO IDENTIFY PHYSICALLY INTEGRATED DESIGN CONCEPTS

Praveen Kumare Gopalakrishnan
Graduate Research Assistant
Mechanical and Aerospace Engineering
University at Buffalo, The State University of New York
Buffalo, NY 14260
pgopalak@buffalo.edu

Helen Kain
Graduate Research Assistant
School of Mathematical Sciences
Rochester Institute of Technology
Rochester, NY 14623
hmk6315@g.rit.edu

Sogol Jahanbekam
Assistant Professor
School of Mathematical Sciences
Rochester Institute of Technology
Rochester, NY 14623
sxjsma@rit.edu

Sara Behdad*
Assistant Professor
Mechanical and Aerospace Engineering
Industrial and Systems Engineering
University at Buffalo, The State University of New York
Buffalo, NY 14260
sarabehd@buffalo.edu

ABSTRACT
This study proposes a graph partitioning method to facilitate the idea of physical integration proposed in Axiomatic Design. According to the physical integration concept, the design features should be integrated into a single physical part or a few parts with the aim of reducing the information content, given that the independence of functional requirements is still satisfied. However, no specific method is suggested in the literature for determining the optimal degree of physical integration of a design artifact. This is particularly important with the current advancement in Additive Manufacturing technologies. Since additive manufacturing allows physical elements to be integrated, new methods are needed to help designers evaluate the impact of the physical integration on the design success. The objective of this paper is to develop a framework for determining the best way that functional requirements can be assigned to different parts of a product.

Keywords: Physical Integration, Axiomatic Design, Additive Manufacturing, Graph Partitioning

1. BACKGROUND
Axiomatic Design (AD) was developed by MIT mechanical engineering professor Num P. Suh in 1976 and was the first to coin the idea of independence of functional requirements. The primary focus of AD is on mapping the problem into several domains (e.g. customer domain, functional domain, physical domain, and process domain), to enable designers to check the axioms and select the best design solution [1]. The first step in designing a system is to define a set of Functional requirements (FRs). The minimum set of independent requirements that the design should satisfy is considered the set of FRs. The next step is to map the set of FRs into the physical domain, or a set of Design Parameters (DPs). Once DPs are determined based on design embodiment principles, designers consider the process domain and identify the Process Variables (PVs). PVs often act as constraints in the system, since designers are not free to change the existing manufacturing processes [2].

Based on the philosophy that good designs share the same characteristics regardless of their physical nature or their domain of application, Suh attempted to root the engineering design

*Corresponding Author

process in two main axioms- (1) Independence Axiom and (2) Information Axiom. According to the independence axiom, FRs (which represent the goals of a design) must remain independent. To satisfy FRs, a set of DPs is chosen. According to the Independence axiom, DPs must be chosen such that the independence of FRs is maintained [3]. Based on the independence axiom, if one of DPs failed, not all functional requirements would be affected. The Independence axiom is based on the concept of changing multi-input/multi-output systems into a set of one-input/one-output systems to maintain the independence of FRs. The aim of information axiom is to minimize the complexity of the system or the information content [3].

What is important in AD is that the design derived from the mapping process must satisfy the Independence Axiom, meaning that the FRs should be satisfied independently with a set of DPs. AD uses design matrices to relate FRs to DPs and represents the design using a set of equations. What makes Axiomatic Design powerful is that it provides a quantitative approach to the formation of normative theories of design [4]. The relationship between the FRs and the DPs is characterized as follows:

$$\{FR\}=[A]\{DP\} \qquad (1)$$

Where each element of Matrix A, Aij, connects a component of the FR vector to a component of the DP vector [5]. The characteristics of design matrix A determine the degree in which the proposed design satisfies the Independence Axiom (Figure 1). For example, a diagonal matrix is an ideal matrix, where each FR is independently satisfied by one corresponding DP (uncoupled design). In the case of a full matrix (coupled design), the design violates the Independence Axiom since the change of any DP has an impact on all FRs. The independence axiom is particularly useful in the case of multi-objective optimization problems, due to the fact that each FR is independently satisfied by a set of design variables [6].

	Uncoupled design	Decoupled design	Coupled design
Design Matrix	$\begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$	$\begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}$	$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$

Figure 1. Three different forms of design matrices

Since its origination in the late 1970s, Axiomatic Design has been the point of attention in the academic research, has been used widely across many disciplines, and has been taught internationally as part of engineering curricula [7][8]. In fact, Axiomatic design is known one of the most important engineering developments of the last century [9]. So far, 10 international conferences on Axiomatic Design have been held in countries around the world, with the last one in Xi'an, China, September 21-24, 2016. In addition to the field of engineering design, AD has impacted a wide range of practices in other disciplines including but not limited to: healthcare delivery systems [10], software design [11], production scheduling [12], manufacturing system design [13], supplier selection [14], interactive art [15], decision science [16], and additive manufacturing [17]. There are, however, several flaws in Axiomatic Design, including the point that there is no structured method available for generating design matrices based on the axioms and the two axioms do not sufficiently capture all that is needed in the design (e.g., human aspects of design [18], consumer preference, market demand [19], manufacturing considerations, and the potential to force a preference structure on designers [20].

However, one main challenge about AD is that the concept of coupled design is very confusing to practitioners. Often designers believe that a simple design is a good design. From this belief, we may conclude that a coupled design in which one DP satisfies multiple FRs is preferred [5]. However, the Independence Axiom does not mean that the DPs must be independent nor that each DP must correspond to a separate physical part. For example, a bottle-can opener is designed to satisfy two FRs and has more than 10 DPs, but has only one piece (Figure 2). It should be noted that the concept of physical integration is completely different from modular design. Module is defined as a part or a group of parts that can be dismantled from the product in a non-destructive way as a unit [21][22]. Ishii et al. [23] have referred to modular design as minimizing the number of functions per part. According to Ulrich and Eppinger [24] the most modular design is one in which each function is implemented by exactly one module or subassembly and there are limited interactions between modules.

With the focus on physical integration of multiple design features integrated into a single part, researchers have come up with various complexity quantifying methods. Decomposition of FR-DP can result in concrete process variables, which is very much essential for practically applicable solutions. Further, this PVs can be integrated with CAs. But most of the complexities have been resorted between FR and DP [25–27]. There are several existing complexity quantifying methods. The concept of changeability and the use of Axiomatic Design when designing production equipment are first introduced. Second, design-solution-specific barriers to flexibility and changeability are described.[28,29]

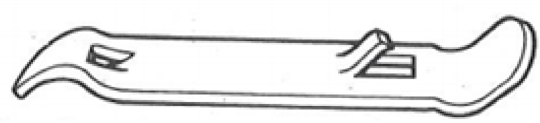


Figure 2. Bottle-can opener: An example of a physically integrated device that satisfies two functional requirements [30].

However, the idea behind *physical integration* or physical coupling is to integrate more than one FR in a single component, as long as FRs remain independent. Therefore, physical integration reduces the design complexity (at least in the physical domain). *While designers are in favor of physical integration, there is no normative approach on how to achieve physical integration using scientific engineering design techniques.*

Graph theory algorithms are widely used in making design decisions [31–33]. Buluc et.al discussed the way that graph partitioning is effective in analyzing complex networks. Division of graphs into small partitions is the primary step for making algorithmic operations more efficient. One of the important sub-steps for complexity reduction or parallelization is graph partitioning. Large graphs are first partitioned into small ones and then they are analyzed. This is highly helpful in simulations, social networks or road networks [34].

Different graph partitioning techniques may be used but the approaches are based on some certain basic algorithms [35]. With the current evolving technology, multiple graph partitions can be run in parallel and complex systems can be analyzed [36,37]. To name a few studies and different applications of graph partitioning methods, Li et al. used the graph partitioning techniques to extract reusable 3D CAD models to improve design reusability [38]. Borisovsky et al. [39] worked on a machining line design problem which has sequences of workstations equipped with processing modules called blocks each of which performs specific operations. They used a graph partitioning technique to integrate machines to perform different sets of operations. Biologists have used the graph partitioning technique that we have adopted to study the RNA structures. They analyzed the best possible RNA configurations that are stable using the Laplacian Eigen values and vectors [40]. We have used this graph partitioning approach using a proposed ranking system for functional requirements to bring in the concept of physical integration in design.

Integrating functional requirements facilitates fewer assembly parts, greater flexibility, and less logical efforts. EOS, a Manufacturer in Germany has used additive manufacturing technology for physically integrating the parts. They wanted to achieve integration of a maximum number of FRs with a minimum number of parts. EOS printed their centrifuge washing rotor using this concept. The traditionally manufactured parts with 32 assembly parts were reduced to 3 assembly parts (2 of them were printed). The structure intermeshing largely helped in reducing complexity and assembly time [41].

The objective of this paper is to provide some background information about the concept of physical integration and open a new venue for determining the best level of physical integration, particularly for additively manufactured parts that have less manufacturing constraints in terms of geometry and shape. The graph theory helps us find the best pair of FRs that can be combined to achieve a more feasible design.

2. PROPOSED METHOD: GRAPH PARTITIONING APPROACH

A graph G is an ordered pair G = (V, E) consisting of a set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V. Graphs can be used to provide algorithmic solutions to some types of real-world problems that can be modeled by graphs.

We say $P = (V_1,.,V_k)$ is a partition of the vertex set of G, if $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup^K_{i=1} V_i = V(G)$.A proper coloring of the graph G is a coloring on the vertices of G in such a way that adjacent vertices receive different colors. We say G is k-colorable, if it has a proper coloring using k colors. Equivalently is k-colorable if we can partition the vertex set of G into k parts in such a way that the vertices in each part are independent, i.e. there is no edge between them. The chromatic number of a graph Written as □(G), is the smallest integer k such that G is k-colorable.

Let u and υ be two vertices of G. We define G.uv to be the graph obtained from G after identifying vertices u and υ in G. If moreover uυ ⊂ E(G), we say we have contracted the edge uυ in G. We define G + uυ and G - uυ to be the graph obtained from G after adding uυ and after removing uυ from G, respectively. For $k \geq 3$ it is NP-complete to decide if a given graph is k-colorable[42] . In fact, there are algorithms with complexity $2^n n^{O(1)}$ to determine if a graph is k-colorable [43]. Determining the chromatic number of a graph is a harder task than checking its k-colorability for a fixed k. As such, it is also NP-complete to compute the chromatic number of a graph. The chromatic number of a graph G with n vertices can be determined in $O(2.2461^n)$ time [43].

The aim of this project is to design a product with a certain number of parts, say k parts, in such a way that the overall cost is as small as possible. We can correspond each of the functional requirements with a vertex. Two functional requirements are adjacent in the graph if they have common design parameters. Each edge between two functional requirements can be labeled by a number, where the value of the number is the cost of having the corresponding functional requirements in the same parts. Equivalently each edge label measures the desire to have the corresponding functional requirements in different parts.

Let G be the resulting graph. We present the labels on the edges by a function ω: E(G) →R. For a partition P of the vertex set of G.

Let Penalty (G; P) = P e is an edge inside a part $\omega(e)$. In the graph-theoretic view the aim then is to find a partition $P = (V_1,..,V_k)$ of the vertex set of the graph in such a way that the sum of the labels of the edges with endpoints both in the same part (i.e. Penalty (G; P)) is as small as possible. In an optimal answer, the functional requirements that correspond to vertices in the same part are recommended to be in the same part of the design.

If this graph G is k-colorable, then there exists an optimal answer with k colors having penalty 0, which is perfect. But it could be the case that the graph is not k-colorable, and in that case, the optimal answer has a positive penalty. As we see here, this problem is more complex than the problem of determining if a graph is k-colorable. Therefore, this problem is also NP-complete. Here we present an algorithm with complexity $O(2^n)$ that leads us to the best optimal answer.

Suppose $V(G) = \{\upsilon_1, \ldots\ldots, \upsilon n\}$ and let $\omega: E(G) \rightarrow R$ be the weighting on E(G). If $\upsilon_1\upsilon_2 \notin E(G)$, define $\omega(\upsilon_i\upsilon_j) = 0$.COk(G) is the minimum Penalty(G,P) among all partitions P of G into k parts. Therefore, our aim here is to find COk(G).

We start by the pair (υ_1, υ_2) of the vertices of G. In any optimal answer to the problem, either υ_1 and υ_2 belong to the same part or they belong to different parts. If υ_1 and υ_2 belong to the same part, then the optimal coloring of G corresponds to an optimal coloring of $G.\upsilon_{1,}\upsilon_2$ and COk(G) =COk(G. $\upsilon_1\upsilon_2$) + w($\upsilon_1\upsilon_2$). Let's call the vertex in $G.\upsilon_{1,}\upsilon_2$ that is obtained after identifying υ_1 and υ_2 in G simply υ1. Note that whenever two vertices get identified in a graph, it might result in some edge identifications as well. If this happens, we choose the weight of the resulting edge to be the sum of the weights of the corresponding edges before identification.

If in an optimal answer υ_1 and υ_2 belong to different parts, then add $\upsilon_1\upsilon_2$ to G (if it is not in G) and replace $\omega(\upsilon_1\upsilon_2)$ by a very big number that is larger than the sum of the initial weights of the edges of G, say 1000. We call this new graph G $*\upsilon_1\upsilon_2$. In this case, we have COk(G) = COk(G $*\upsilon_1\upsilon_2$).

Therefore combining the two cases together implies COk(G) = min {COk(G. $\upsilon_1\upsilon_2$) + $\omega(\upsilon_1\upsilon_2)$,COk(G $*\upsilon_1\upsilon_2$)}.We continue the above process on each of the graphs G. $\upsilon_1\upsilon_2$ and G$*\upsilon_1\upsilon_2$ by choosing another pair of vertices of G. At each step we choose a pair of vertices of the graph in such a way that the sum of their indices is the smallest value. If we obtain a graph of at most k vertices, or if we obtain a complete subgraph of order k + 1 all whose edges have weight at least 1000, then we can determine COk of the corresponding graph. Also note that after any vertex identification, the number of edges can reduce by at most a half. Thus if we repeat the above process in $n\text{-}k+1+2\binom{k+1}{2}$ steps, the resulting graphs are either a graph with at most k vertices or contain a complete subgraph of size k + 1 all whose edges have weight at least 1000. Each of these graphs can be handled easily. A graph with at most k vertices

is simply k-colorable, and any k-partition of a graph containing a complete subgraph of size k + 1 with all edges of weight at least 1000 contains at least one edges of size at least 1000 such that both of its endpoints belong to the same part. But this makes the penalty of this partition extremely big. As a result, this graph does not have anything to do with the optimal answer and can be ignored. Therefore in order to determine the optimal answer, we need to apply the above algorithm at most $n\text{-}k+1+2\binom{k+1}{2}$ steps. Since at each step two graphs are created, the complexity of this algorithm is $O(2^n)$.

Remark 1. This problem might look similar to graph clustering problems, but in fact, they are rather opposite. In graph clustering problems the aim is to cluster (partition) the graph in such a way that the parts in the partition form clusters, i.e. the graph is dense within the parts and it is sparse between the parts. This is exactly opposite of what we are trying to do. Here in this research, we are looking for a partition that is sparse inside parts and as a result denser in between parts. For that reason, the algorithms to solve these two problems are different and have different complexities.

To do this, we first randomly partition V (G) into k parts, say $V_1, \ldots, V_k$. Now we apply the following algorithm until it stops. Let $V(G) = \{\upsilon_1, \ldots., \upsilon n\}$. Start from vertex υ_1. At each step look at the vertex υ_i. Let $Sj(\upsilon_i)$ be the summation of the weights of the edges that join vi to the part Vj . Suppose

minj $\{Sj(\upsilon_i)\} = Sk(\upsilon_i)$. If the vertex vi does not belong to V_k, relocated υ_i to part V_k and start the algorithm again. Otherwise, continue to vertex υ_{i+1}. The algorithm stops when υ_n belongs to the part with minimum $Sj(\upsilon_n)$.

Figure 3 shows the procedure of the proposed analysis. We can repeat the process from Step 2, defining the number of products and then fix the edges according to our needs and a set of design constraints, and check the feasibility of the proposed design. This step can be repeated till we get the least number of parts with consideration on the FRs to be independent. This is further discussed with a numerical example of how the graph partition works and it is implemented on a simple example to show its application.

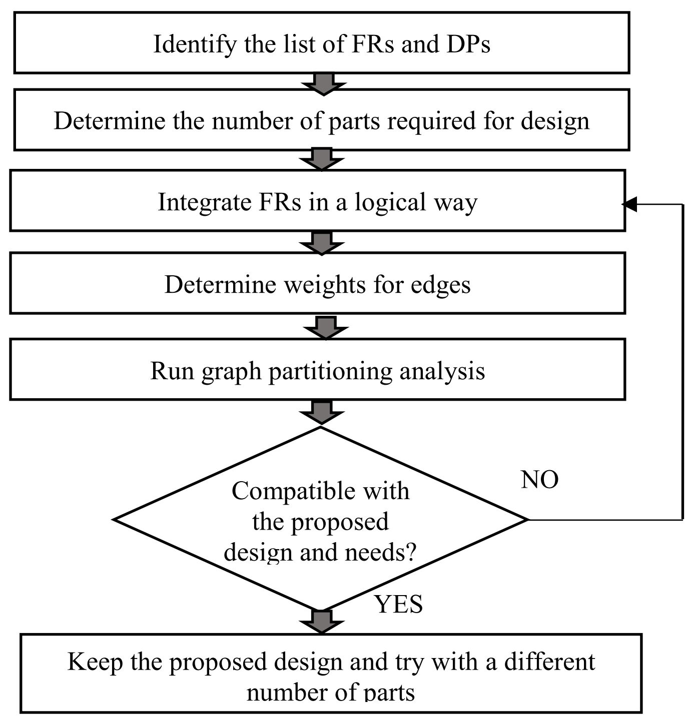
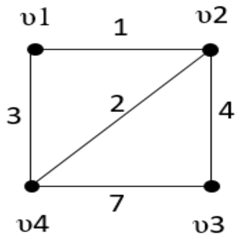


Figure 3: The overall procedure of the analysis

3. NUMERICAL EXAMPLE

In this section, first, we provide a numerical example to show the way a design matrix or its graph equivalent is partitioned to determine the assignment of functional requirements (edges or nodes of the graph) to a fixed number of parts. Figure 4 provides a numerical example in which we attempt to optimally partition four functional requirements into three parts.

The graph shown in Figure 4 is an illustration of the theory. The purpose is to partition four functional requirements represented as the edges of the graph into υ_1, υ_2, υ_3 & υ_4. The $w(v_iv_j)$ defined several constraints or preferences with respect to these four edges.



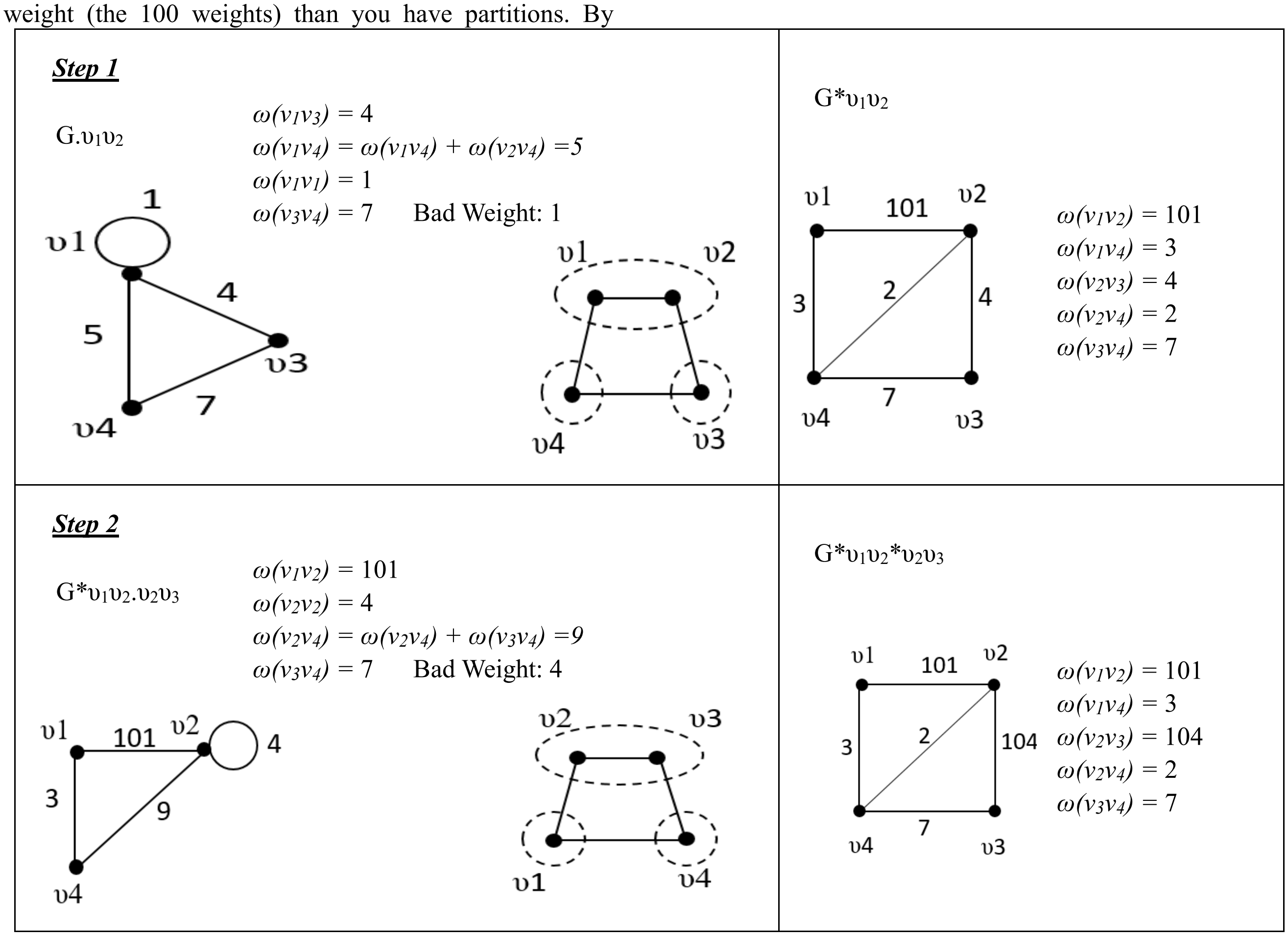
$\omega(v_1v_2) = 1$
$\omega(v_1v_4) = 3$
$\omega(v_2v_3) = 4$
$\omega(v_2v_4) = 2$
$\omega(v_3v_4) = 7$

Figure 4. Initial graph G (the equivalent of design matrix) and the corresponding weight of each arc

Figure 5 shows the steps involved in the optimal portioning of the FRs among parts. The working of the algorithm is clearly illustrated. The process works like a binary tree. You break the graph up two ways: either contract an edge to show the vertices are in the same partition or you make the weight between the vertices extremely high (add 100 in this case) to show they are in different parts. You repeat this until you have a complete graph (for edge contraction) or until you have more edges of very high weight (the 100 weights) than you have partitions. By following the algorithm backward, you can see which vertices are in the same part and which vertices are in different parts.

The first step shows the optimal coloring of G corresponding to an optimal coloring of G.υ_1,υ_2 and $\omega(v_1v_4)$ *is defined as* $\omega(v_1v_4) + \omega(v_2v_4) = 5$. If we look at the possibility of 1 & 2 combining together the sum of weights appears to be 101, a number larger than the sum of initial weights. The next step, we try to combine FRs 2 & 3, FR 1 & 4 and so on to make the edges high weight, Finally, when we combine part 1 and 3 we see that they appear to be in a single part. The reason that Step 5 gives the best answer is that the minimum possible cost is 0 and at Step 5 we get bad weight 0, which is the smallest possible. If the bad weight on this step was not 0, we would need to continue the algorithm by the end.

The process works like a binary tree. You break the graph up two ways: either contract an edge to show the vertices are in the same partition or you make the weight between the vertices extremely high (add 100 in this case) to show they are in different parts. You repeat this until you have a complete graph (for edge contraction) or until you have more edges of very high weight (the 100 weights) than you have partitions. By following the algorithm backward, you can see which vertices are in the same part and which vertices are in different parts.



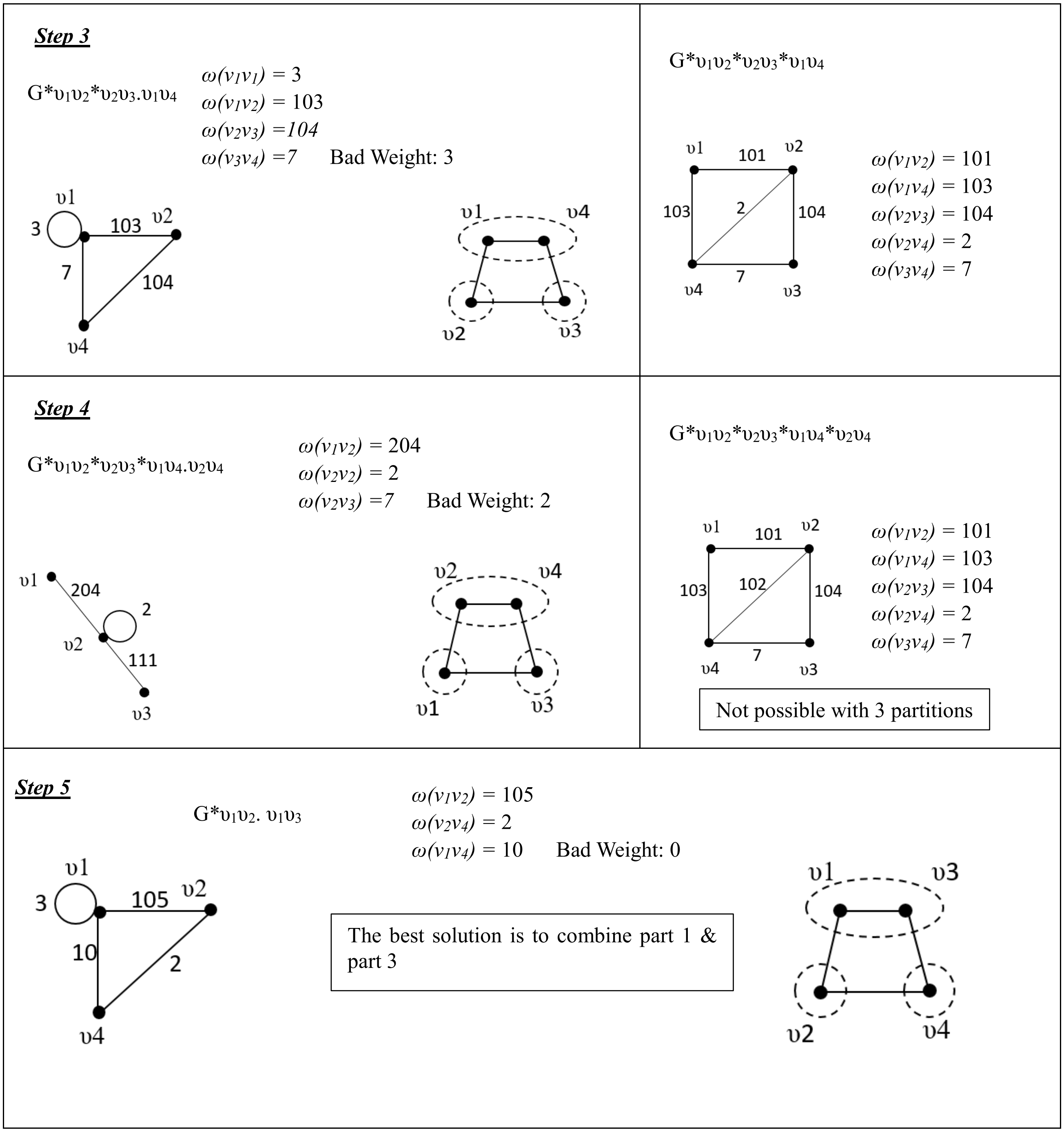


Figure 5. Steps involved in the optimal partitioning of the FRs among parts.

The question then arises as of how could we make sure that we always identify a feasible set of design parameters after combining the functional requirement. The algorithm examines all possible cases and always gives the best answer. If two design parameters are not compatible, when we model the problem into a graph we make an edge with a high weight between them. This way we can make sure that in the best answer, we never get those incompatible design parameters together.

In our numerical example, the best solution happens to be an answer that combines two non-adjacent vertices. But this is not true for the general graphs, because not always the best answer has penalty 0. In the way we do the modeling, the aim is to partition the vertex set of a graph into given number of parts in such a way that the cost of edges within parts is as small as possible. Therefore, the cost penalty is the sum of weights of edges within the parts. When it is zero it means that we have a perfect answer.

Designs which do not satisfy the Independence Axiom are called coupled. Designs which satisfy the Independence Axiom are called uncoupled or decoupled. The difference is that in an uncoupled design, the DPs are totally independent, while with a decoupled design, at least one DP affects two or more FRs. As a result, the order of adjusting the DPs in a decoupled design is important. However, the idea behind physical integration or physical coupling is to integrate more than one FR in a single component. The proposed algorithm in this study is focused on assigning FRs to a fewer number of parts to enhance physical integration. The algorithm is general and it does not necessarily require the independence of FRs, however since we would like to improve the concept of axiomatic design, we assign FRs into different parts in such a way that the FRs in each part are independent, i.e. there is no shared DPs between them.

Integrated design includes various functions in one part. Such integration results in reduced number of parts. Thus making parts more compact and it reduces the assembly time significantly. The proposed algorithm identifies the best combination to group the parts, thereby facilitating easy printing using a 3D printer. It is expected that the integration of parts reduces the assembly time.

Sometimes it may happen that the grouped part may have one function, but because of manufacturing constraints, they are produced as separate parts. Therefore, physical integration does not necessarily mean that all parts should be manufactured at the same time.

4. EXAMPLE OF PENCIL DESIGN

This section explains an example of a mechanical pencil. First, the FRs and the DPs of the pencil are defined. DSM matrix is formed for the proposed design.

Figure 6 illustrates the different parts of a mechanical pencil. It mainly consists of a body, lead reservoir tube, eraser, and lead sleeve [44]. It has 8 parts assembled together to get the product.

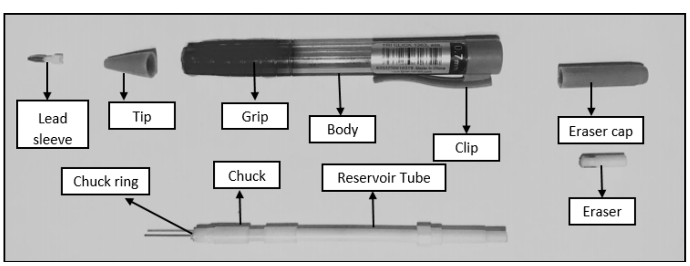


Figure 6. An example of a currently used design for a Mechanical Pencil

The following functional requirements are defined for the mechanical pencil:

FR1- Erasing
FR2- Storage for lead
FR3 – Storage for the eraser
FR4- Advance lead
FR5- Support lead while using
FR6- Position lead in place (lead sleeve)
FR7- Grip for comfort.
FR8- Holding clip.
FR9- Accommodating body for all other parts.

Next, in order to explore the design concept, the design parameters need to be defined.

DP1- Eraser
DP2- Opening for the eraser
DP3- Cylinder with stopper
DP4- Spring lead advancer for the lead movement (spring)
DP5- Chuck to hold lead
DP6- External grip
DP7- Chuck opening to accommodate lead of different sizes (chuck ring)
DP8- Clip design (integrated to push button)
DP9- Body geometry

The design matrix [A] in [FR]=[A][DP] shows the relations between the given set of functional requirements and the design parameters. The design matrix for the mechanical pencil is defined in Equation 2.

$$\begin{pmatrix} FR1 \\ FR2 \\ FR3 \\ FR4 \\ FR5 \\ FR6 \\ FR7 \\ FR8 \\ FR9 \end{pmatrix} = \begin{pmatrix} X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & X & X & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & X & X & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & 0 & X & 0 & 0 \\ 0 & 0 & 0 & X & X & 0 & X & 0 & 0 \\ 0 & 0 & 0 & 0 & X & 0 & X & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & X & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & X & 0 \\ X & X & 0 & 0 & 0 & X & 0 & X & X \end{pmatrix} \begin{pmatrix} DP1 \\ DP2 \\ DP3 \\ DP4 \\ DP5 \\ DP6 \\ DP7 \\ DP8 \\ DP9 \end{pmatrix} \tag{2}$$

If we look at FR4, advancement of the lead, the spring (DP4), chuck (DP5), and chuck ring (DP7) work together when we push the button, the lead is transferred from the lead reservoir tube through the lead sleeve. When the push button is pressed, the chuck goes past the chuck ring and the lead falls through, and when the button is released the jaws close to hold the lead. They retract back into the chuck ring thereby holding the lead in place. Similarly, FR6, the position of lead is dependent on chuck design (DP5) and chuck ring (DP7).

The current design does not satisfy independence axiom; each individual functional requirement is not satisfied by fully independent physical components or subsystems. A decoupled or uncoupled design for the mechanical pencil is essentially difficult to achieve, as many of the design parameters are reused for multiple functions. So make it satisfy the independence axiom, we have to integrate design features in a single physical part if FRs can be independently satisfied in the proposed solution [45].

Per the concept of physical integration, the objective is to satisfy the list of functional requirements with a minimum number of parts. We will look at two cases in this section. Case 1 in which FRs are assigned to 5 parts and Case 2 in which FRs are assigned to 4 parts.

4.1. Case 1: 5-part Design

In this case, we are integrating the design to satisfy two or more FRs with a single physical part. So this design is reduced to 5 parts by the concept of physical integration by integrating the functional requirements together.

We have a strong desire to have FR1 (eraser), FR6 (sleeve), and FR8 (clip) in separate parts in our design. Therefore, when modeling the corresponding graph, we make vertices FR1, FR6, and FR8 adjacent to all other vertices of the graph.

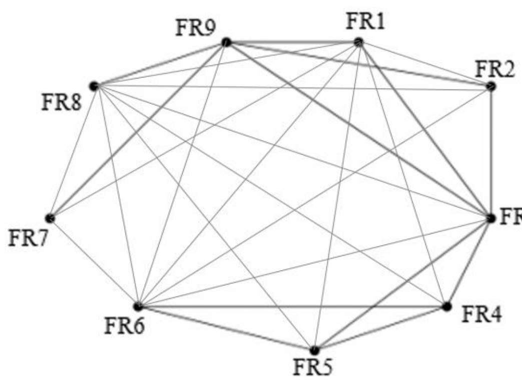


Figure 7: The initial graph of the design matrix

Let us assume the initial parts be {FR1}, {FR6}, {FR8}, {FR2, FR3, FR4}, {FR5, FR7, FR9}.

Table 1: the weight of each arc considered for the graph of Figure 7 based on technical constraints and manufacturability requirements

w(FR1-FR3) = 10	*w*(FR1-FR5) = 10
w(FR1-FR9) = 10	*w*(FR1-FR6) = 10
w(FR2-FR3) = 2	*w*(FR1-FR7) = 10
w(FR2-FR9) = 5	*w*(FR1-FR8) = 10
w(FR3-FR4) = 1	*w*(FR6-FR2) = 10
w(FR3-FR5) = 2	*w*(FR6-FR3) = 10
w(FR3-FR9) = 10	*w*(FR6-FR7) = 10
w(FR4-FR5) = 2	*w*(FR6-FR8) = 10
w(FR4-FR6) = 10	*w*(FR6-FR9) = 10
w(FR5-FR6) = 10	*w*(FR8-FR2) = 10
w(FR7-FR9) = 5	*w*(FR8-FR3) = 10
w(FR8-FR9) = 10	*w*(FR8-FR4) = 10
w(FR1-FR2) = 10	*w*(FR8-FR5) = 10
w(FR1-FR4) = 10	*w*(FR8-FR7) = 10

In Table 1, we have labeled all edge incident to FR1, FR6, FR8 by 10. This represents the technical constraints that designer often has. Now let's apply the algorithm. We randomly partition the vertices of the graph into 5 parts.

After applying the algorithm, the final answer obtained {FR1}, {FR6}, {FR8}, {FR2, FR3, FR4, FR7}, {FR5, FR7, FR9}. In this case, the penalty is not zero like the numerical example. The minimum penalty is 2+1=3, because FR2, and FR3 are in the same parts and FR3 and FR4 are in the same parts. The algorithm suggested FR7 to be integrated either with FR2, FR3 & FR4 or with FR5 & FR9.

Based on the algorithm results, it seems that several FRs are grouped and can be put in same parts. One example of such proposed 5-part design is illustrated in Figure 8. The FRs and DPs are also integrated to form a design matrix with 5 parameters. The Chuck, chuck ring, spring and the lead reservoir tube are integrated into a single part. This satisfies FR2 (lead storage), FR4 (advance lead), FR5 (support lead while using). The body design is integrated together as one, it has the opening to accommodate eraser, it integrates the clip and the grip with the body thereby satisfying FR3 (eraser storage), FR7 (grip for comfort), FR8 (clip to hold). The other two parts are tip with lead sleeve and eraser.

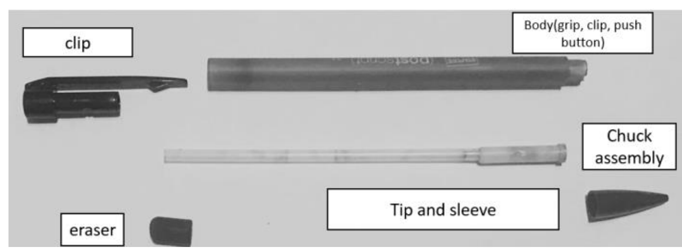


Figure 8: A 5-part design available in the market

The functional requirement for this particular mechanical pencil can be summarized as follow:

FR1- Erasing
FR2- Lead chuck
  FR21- Storage for lead
  FR22-Advance lead
  FR23- Support lead while using
FR3 – Body
  FR31- Grip for comfort
  FR32- Storage for the eraser
FR4- Position lead in place (lead sleeve)
FR5- Clip to hold.

Further to explore the design concept the design parameters need to be defined.
DP1- Eraser
DP2- Body design
  DP21- Opening for the eraser

DP22- External grip
DP3-Chuck design
DP31-Cylinder with stopper
DP32- Spring lead advancer for the lead movement (spring)
DP33-Chuck opening to accommodate lead of different sizes (chuck ring)
DP4-Lead sleeve design
DP5-Clip design

The new design matrix is as follow:

$$\begin{pmatrix} FR1 \\ FR2 \\ FR3 \\ FR4 \\ FR5 \end{pmatrix} = \begin{pmatrix} X & 0 & 0 & 0 & 0 \\ 0 & 0 & X & X & 0 \\ 0 & X & 0 & 0 & 0 \\ 0 & 0 & X & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \begin{pmatrix} DP1 \\ DP2 \\ DP3 \\ DP4 \\ DP5 \end{pmatrix} \quad (3)$$

We should note that we do not necessarily need to redefine the design matrix and the original design matrix with 9 FRs can be used in the analysis.

4.2. Case 2: 4-part Design

Now, consider a different set of requirements. Suppose that designers are interested in designing a four-part product in which FR1 must be in a separate part in our design due to some technical constraints (e.g. material selection, shared design parameters). Therefore, when modeling the corresponding graph, we make vertex FR1 adjacent to all other vertices of the graph. Figure 9 explains all conditions that we want to consider for this case. Since we have the strong desire that FR1 be in a separate part, we label all edge incident to FR1 by 10. Our aim here is to design this product in such a way that it has 4 parts. Now let's apply the algorithm. We randomly partition the vertices of the graph into 4 parts.

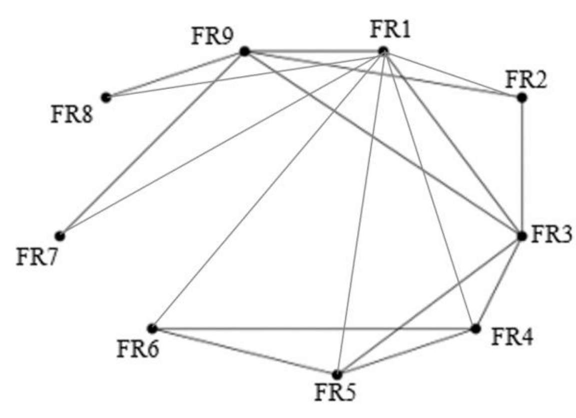


Figure 9: Initial graph for the proposed design

So let the initial parts be {FR1}, {FR2, FR5}, {FR3, FR4}, {FR6, FR7, FR8, FR9}. This design consideration was proposed from an additive manufacturing perspective. These factors were adopted from a logical perspective to reduce the number of parts. The idea of this consideration was based on a design proposed by Grunewald [46] who successfully printed a 3D pencil with four moving parts. His design can be printed using white polypropylene material with a water soluble gel-like support structure. He also suggested the pencil body must have a geometry with a lot of holes to facilitate easy removal of material from the support structure in a 3D printer.

After applying the algorithm, the final answer will be {FR1}, {FR3, FR6, FR7, FR8}, {FR2, FR5}, and {FR4, FR9}. we can integrate multiple FRs and can be printed. So this design considers the FR1 (erasing) as a separate part and rest can be integrated in any way.

Table 2: weight of each arc considered for the graph in Figure 8

w(FR1-FR3) = 10	w(FR5-FR6) = 2
w(FR1-FR9) = 10	w(FR7-FR9) = 5
w(FR2-FR3) = 2	w(FR8-FR9) = 2
w(FR2-FR9) = 5	w(FR1-FR2) = 10
w(FR3-FR4) = 1	w(FR1-FR4) = 10
w(FR3-FR5) = 2	w(FR1-FR5) = 10
w(FR3-FR9) = 10	w(FR1-FR6) = 10
w(FR4-FR5) = 2	w(FR1-FR7) = 10
w(FR4-FR6) = 1	w(FR1-FR8) = 10

In Table 2, we have labeled all edge incident to FR1 by 10 since we want to include FR 1 as a separate part. In this particular example, FR 1 can be excluded from analysis as well, however, we have included in as inputs to the algorithm to show the application of the model for the cases in which such constraints exist. Our aim here is to design this product in such a way that it has 4 parts. We have included the FR1 also in the algorithm to show, the independent FRs does not affect the result of the proposed algorithm. The FR is assigned to the separate part and the algorithm runs to integrate other FRs among the parts.

After applying the algorithm, the final answer obtained is {FR1}, {FR3, FR6, FR7, FR8}, {FR2, FR5}, and {FR4, FR9}. So this design is devoid of any penalty. It appears that we can satisfy the list of our FRs with only four parts. One example of such designs in when the body design is modified to accommodate the lead positioning, clip and grip [46] (as shown in Figure 10). This design is inspired from screw mechanism, on rotation the back enters inside the hollow body of the pencil and facilitates the movement of lead. The detents in the screw piece clicks into place every turn, on rotation clockwise the lead moves forward and vice-versa. The lead sleeve helps us in holding the lead (prevents sliding).

The functional requirement for this particular mechanical pencil can be summarized as follows:

FR1- Erasing
FR2- Lead chuck
FR21- Storage for lead
FR22-Advance lead

FR23- Support lead while using
FR3- Back press button
FR31- Storage for the eraser
FR32-Stopper for lead
FR4– Body
FR41- Holding clip
FR42- Grip for comfort

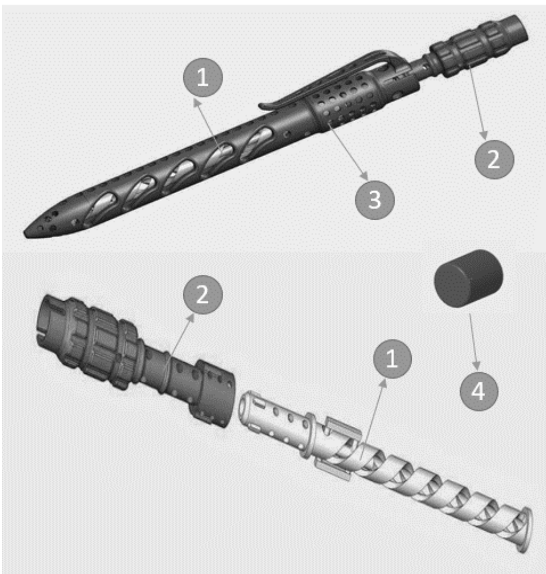


Figure 10: An example of a 4-part design, gotten from [46]
(1) Screw piece (2) Back of pencil (3) Pencil Body (4) Eraser

Further to explore the design concept the design parameters need to be defined.

DP1- Eraser
DP2- Screw piece
DP21- Lead advancer
DP22- Cylinder with stopper.
DP3- Back of pencil
DP31- Opening for the eraser.
DP32- Accomdate screw motion
DP4- Body design
DP41- Clip design
DP42- Grip design

The design matrix for the new design is:

$$\begin{pmatrix} FR1 \\ FR2 \\ FR3 \\ FR4 \end{pmatrix} = \begin{pmatrix} X & X & 0 & 0 \\ 0 & X & X & 0 \\ 0 & 0 & X & 0 \\ 0 & 0 & 0 & X \end{pmatrix} \begin{pmatrix} DP1 \\ DP2 \\ DP3 \\ DP4 \end{pmatrix} \quad (4)$$

We have analyzed 3 cases, by the concept of physical integration, more than one FR in a single component is integrated together.

The original design was made of 8 parts and had 8 FRs, the proposed design has 5 parts- 5 FRs and 4 parts- 4FRs, where these 9 FRs are integrated together.

It should be noted that in this example, a square DSM is considered, however, the algorithm can be used for any design matrices. We have used square matrices since we wanted to consider the independence of FRs as suggested in the axiomatic design. The purpose of this example is to show how the proposed algorithm can help designers improve the degree of physical integration. This proposed designs can be obtained from simple design assumptions as well, but in this case, the proposed algorithm is tested with this example.

Tables 3 and 4 show the comparison of these three cases in terms of the number of FRs, DPs, and the number of parts.

Table 3: The comparison of the 3 resulting designs from graph partitioning method.

Case	**Design parameters**	**Number of parts**
Original design	9	8
Proposed design-1	5	5
Proposed design-2	4	4

Table 4: The table shows the integration of FRs and the number of parts in each proposed design.

Inputs	**Number of FRs**	**Number of parts**	**Parts showing the integration of FRs**
Original Design	8	8	All 8 FRs satisfied by different parts
Proposed design-1	8	5	Part 1- FR1 Part 2- FR2, FR3, FR4, FR7 Part 3- FR5, FR9 Part 4-FR6 Part 5-FR8
Proposed design -2	8	4	Part 1-FR1 Part 2-FR3, FR6, FR7, FR8 Part 3-FR2, FR5 Part 4- FR4,FR9

If we look at the two proposed design, the 4-part design seems to work better.

With the emerging 3D printing technology, parts can be integrated and printed from a CAD file. So the concept of physical integration or simple design that often results in lower cost and higher quality is feasible through AM. For example, the 4-part design has a different set of FRs integrated together but the pencil works the way it is intended to. The penalty for this design was also zero. So this gives an edge over the 5-part design.

We should note that in a particular system, not all parts are suited for additive manufacturing. In other words, they do not bear the same potential for improvement. Therefore, analyzing the previous models and designs help us at arriving at a conclusion of which parts can be integrated together.

4. CONCLUSION AND FUTURE WORK

This research deals with analyzing the concept of physical integration originated in axiomatic design field. A graph partitioning method is proposed for determining the best pairs of FRs that can be physically integrated into a single part. The proposed method is employed for an example of designing a pencil, which initially is made of 8 parts. It has been shown that the number of parts can be reduced to 4 and 5 parts where all the FRs are independent and serves its purpose. Depending on the manufacturing constraints, cost and other technical considerations, designers may select one design over the other.

This research can be extended in several ways. The algorithm can be extended to determine the optimum assignment of FRs while satisfying the independence of the FRs as one of the main principles of Axiomatic Design. In addition, the information content of each design alternative can be calculated as another factor to be added to the algorithm. Furthermore, the proposed method can be extended to determine the optimal number of parts needed to satisfy the pre-defined set of FRs. The proposed method is one step towards developing methods that can help designers define the optimal degree of physical integration for design alternatives.

This study can be extended to more complex designs with more number of parts, FRs, and DPs. The implementation of the outputs of the proposed algorithm is feasible through the recent advancement in additive manufacturing where parts with different geometries and shapes are manufactureable. Physically integrated parts can be manufactured using the capabilities of additive manufacturing technology. Another area for future research is to study the economic viability and efficiency of physically integrated parts considering the point that the number of parts, manufacturing processes, and assembly times are reduced. In addition, the proposed algorithm can be run for more complicated designs to better reveal the performance of the algorithm under different conditions. In the case of complex systems, not every part needs to be printed at the same time. Instead integrated parts can be printed separately and be assembled together to reduce the operation time and cost.

Acknowledgments

This material is based upon work supported by the National Science Foundation–USA under grants # CMMI-1727190 and CMMI-1727743. 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.

5. REFERENCES

[1] Shirwaiker, R. A., and Okudan, G. E., 2008, “Triz and Axiomatic Design: A Review of Case-Studies and a Proposed Synergistic Use,” Journal of Intelligent Manufacturing, **19**(1), pp. 33–47.

[2] Suh, N. P., 1998, “Axiomatic Design Theory for Systems,” Research in Engineering Design, **10**(4), pp. 189–209.

[3] Suh, N. P., 2016, “Challenges in Designing and Implementing Large Systems (Overcoming Cost Overruns and Missed Project Schedules),” *Axiomatic Design in Large Systems*, Springer International Publishing, Cham, pp. 273–309.

[4] Farid, A. M., and Suh, N. P., 2016, *Axiomatic Design in Large Systems: Complex Products, Buildings and Manufacturing Systems*, Springer.

[5] Park, G.-J., 2014, “Teaching Conceptual Design Using Axiomatic Design to Engineering Students and Practitioners,” Journal of Mechanical Science and Technology, **28**(3), pp. 989–998.

[6] Hirani, H., and Suh, N. P., 2005, “Journal Bearing Design Using Multiobjective Genetic Algorithm and Axiomatic Design Approaches,” Tribology International, **38**(5), pp. 481–491.

[7] Tate, D., 2015, “Axiomatic Design: Review, Impact, and Future Direction,” *Volume 15: Advances in Multidisciplinary Engineering*, ASME, p. V015T19A006.

[8] Park, G.-J., 2007, *Analytic Methods for Design Practice*, Springer Science & Business Media.

[9] “The 10th International Conference on Axiomatic Design | September 21-23, 2016, Xi’an China” [Online]. Available: http://icad2016.org/.

[10] Bosire, J., Wang, S., Khasawneh, M., Gandhi, T., and Srihari, K., 2016, “Designing an Integrated Surgical Care Delivery System Using Axiomatic Design and Petri Net Modeling,” Springer International Publishing, pp. 73–101.

[11] Girgenti, A., Giorgetti, A., Citti, P., and Romanelli, M., 2015, “Development of a Custom Software for Processing the Stress Corrosion Experimental Data through Axiomatic Design,” Procedia CIRP, **34**, pp.

250–255.

[12] Cochran, D. S., Eversheim, W., Kubin, G., and Sesterhenn, M. L., 2000, "The Application of Axiomatic Design and Lean Management Principles in the Scope of Production System Segmentation," International Journal of Production Research, **38**(6), pp. 1377–1396.

[13] Fan, S. H., Li, J. H., Jiang, Z. B., and Zhang, Z. G., 2014, "Axiomatic Design of Facility Layout for Reconfigurable Manufacturing System," Applied Mechanics and Materials, **703**, pp. 273–276.

[14] Zaralı, F., and Yazgan, H. R., 2016, "Solution of Logistics Center Selection Problem Using the Axiomatic Design Method," World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, **10**(3), pp. 489–495.

[15] Foley, J. T., and Harðardóttir, S., 2016, "Creative Axiomatic Design," Procedia CIRP, **50**, pp. 240–245.

[16] Fan, L. X., Cai, M. Y., Lin, Y., and Zhang, W. J., 2015, "Axiomatic Design Theory: Further Notes and Its Guideline to Applications," International Journal of Materials and Product Technology.

[17] Salonitis, K., 2016, "Design for Additive Manufacturing Based on the Axiomatic Design Method," The International Journal of Advanced Manufacturing Technology, pp. 1–8.

[18] Maier, J. R. A., and Fadel, G. M., 2009, "Affordance Based Design: A Relational Theory for Design," Research in Engineering Design, **20**(1), pp. 13–27.

[19] Hazelrigg, G. A., 2003, "Validation of Engineering Design Alternative Selection Methods," Engineering Optimization, **35**(2), pp. 103–120.

[20] Olewnik, A., and Lewis, K., 2005, "On Validating Engineering Design Decision Support Tools."

[21] Allen, K. R. and Carlson-Skalak, S., 1998, "No Title," *Defining Product Architecture during Conceptual Design*, Proceedings of the 1998 ASME Design Engineering Technical Conference, Atlanta, GA.

[22] Gershenson, J. K., Prasad, G. J., and Zhang, Y., 2003, "Product Modularity: Definitions and Benefits," Journal of Engineering design, **14**(3), pp. 295–313.

[23] Ishii, K., Juengel, C. and Eubanks, C. F., 1995, "Design for Product Variety: Key to Product Line Structuring," *Proceedings of the 1995 ASME Design Engineering Technical Conferences – 7 Th International Conference on Design Theory and Methodology. The American Society of Mechanical Engineers.*

[24] Ulrich, K. and Eppinger, S. D., 1995, *Product Design and Development*, McGraw-Hill.

[25] Guenov, M. D., and Barker, S. G., 2005, "Application of Axiomatic Design and Design Structure Matrix to the Decomposition of Engineering Systems," Systems engineering, **8**(1), pp. 29–40.

[26] Eppinger, S. D., and Browning, T. R., 2012, *Design Structure Matrix Methods and Applications*, MIT press.

[27] Pektaş, Ş. T., and Pultar, M., 2006, "Modelling Detailed Information Flows in Building Design with the Parameter-Based Design Structure Matrix," Design Studies, **27**(1), pp. 99–122.

[28] Tang, D., Zhang, G., and Dai, S., 2009, "Design as Integration of Axiomatic Design and Design Structure Matrix," Robotics and Computer-Integrated Manufacturing, **25**(3), pp. 610–619.

[29] Foith-Förster, P., Wiedenmann, M., Seichter, D., and Bauernhansl, T., 2016, "Axiomatic Approach to Flexible and Changeable Production System Design," Procedia CIRP, **53**, pp. 8–14.

[30] Suh, N. P., 1990, *The Principles of Design*, Oxford University Press New York.

[31] Zetterberg, A., Mörtberg, U. M., and Balfors, B., 2010, "Making Graph Theory Operational for Landscape Ecological Assessments, Planning, and Design," Landscape and urban planning, **95**(4), pp. 181–191.

[32] Bondy, J. A., and Murty, U. S. R., 1976, *Graph Theory with Applications*, Citeseer.

[33] Papalambros, P. Y., 1995, "Optimal Design of Mechanical Engineering Systems," Journal of Vibration and Acoustics, **117**(B), pp. 55–62.

[34] Buluç, A., Meyerhenke, H., Safro, I., Sanders, P., and Schulz, C., 2016, "Recent Advances in Graph Partitioning," *Algorithm Engineering*, Springer, pp. 117–158.

[35] Christian Schulz, *Graph Partitioning and Graph Clustering in Theory and Practice*.

[36] Fern, X. Z., and Brodley, C. E., 2004, "Solving Cluster Ensemble Problems by Bipartite Graph Partitioning," *Proceedings of the Twenty-First International Conference on Machine Learning*, ACM, p. 36.

[37] Hendrickson, B., and Kolda, T. G., 2000, "Graph Partitioning Models for Parallel Computing," Parallel computing, **26**(12), pp. 1519–1534.

[38] Li, M., Zhang, Y. F., and Fuh, J. Y. H., 2010, "Retrieving Reusable 3D CAD Models Using Knowledge-Driven Dependency Graph Partitioning," Computer-Aided Design and Applications, **7**(3), pp. 417–430.

[39] Borisovsky, P., Dolgui, A., and Kovalev, S., 2012, "Algorithms and Implementation of a Set Partitioning Approach for Modular Machining Line Design,"

Computers & Operations Research, **39**(12), pp. 3147–3155.

[40] Kim, N., Zheng, Z., Elmetwaly, S., and Schlick, T., 2014, “Rna Graph Partitioning for the Discovery of Rna Modularity: A Novel Application of Graph Partition Algorithm to Biology,” PloS one, **9**(9), p. e106074.

[41] EOS, “Functional Integration.”

[42] L. Lovasz, 1973, “Coverings and Coloring of Hypergraphs,” Proceedings of the 4th Southeastern Conference on Combinatorics,Graph Theory, and Computing, Utilitas Mathematica, pp. 9–12.

[43] Björklund, A., Husfeldt, T., and Koivisto, M., 2009, “Set Partitioning via Inclusion-Exclusion,” SIAM Journal on Computing, **39**(2), pp. 546–563.

[44] Quality Logo Products Inc., 2018, “The Anatomy of Pens and Pencils” [Online]. Available: https://www.qualitylogoproducts.com/promo-university/anatomy-of-pens-pencils.htm.

[45] N. P. Suh, 2001, *Axiomatic Design: Advances and Applications*, The Oxford Series on Advanced Manufacturing.

[46] Grunewald, S., 2014, “3D Printing Because You Can: Creating a 3D Printable Mechanical Pencil” [Online]. Available: https://3dprintingindustry.com/news/3d-printing-can-3d-printable-mechanical-pencil-29778/.