8-2013

Automated Complexity Based Assembly Time Estimation Method

Essam Namouz
Clemson University, enamouz@gmail.com

AUTOMATED COMPLEXITY BASED ASSEMBLY TIME ESTIMATION METHOD

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Industrial Engineering

---

by
Essam Zuhair Namouz
August 2013

Accepted by:
Dr. Joshua D. Summers, Committee Co-Chair
Dr. Mary Elizabeth Kurz, Committee Co-Chair
Dr. David Neyens
Dr. Anand K. Gramopadhye

ABSTRACT

The overall goal of this research is to create an automated assembly time estimation method that is accurate and repeatable in an effort to reduce the analysis time required in estimating assembly times. Often, design for assembly (DFA) approaches are not used in industry due to the amount of time required to train engineers in the use of DFA, the time required to conduct the analysis, and the product level of detail needed. To decrease the analysis time and effort required in implementing the assembly time estimation portion of DFA, a tool is needed to estimate the assembly time of products while reducing the amount of information required to be manually input from the designer.

The Interference Detection Method (IDM) developed in this research retrieves part connectivity information from a computer-aided design (CAD) assembly model, based on a parts' relative location in the assembly space. The IDM is used to create the bi-partite graphs that are parsed into complexity vectors used with the artificial neural network complexity connectivity method to predict assembly times. The IDM is compared to the Assembly Mate Method which creates the connectivity graph based on the assembly mates used in creating the assembly model in CAD (SolidWorks). The results indicate that the IDM has a similar but larger percent error in estimating assembly time than the AMM. However, the variance of the AMM is larger than the variance observed with the IDM.

The AMM requires the assembly mates to create the connectivity graph, which may vary based on the designer creating the assembly model. The IDM, based on part location within the assembly model, is independent of any mates used to create the assembly. Finally, the assembly mate information is only stored in the SW assembly file, limiting the functionality of the AMM to SolidWorks assembly files. The IDM operates on the solid bodies in the assembly model, and therefore can be executed on an assembly after being imported by SW using common CAD exchange file types: assembly file (*.sldasm), IGES (*.iges), parasolid(*.x_t), and STEP (*.step;*.stp).

The IDM was also trained and tested as a tool for use during the conceptual phase of the design process. Assembly models were reduced in fidelity to represent a solid model created early in the design process when detailed information regarding the part geometry is not known. The complexity vectors of the reduced fidelity model are used as the input into a modified complexity connectivity method to estimate assembly time. The results indicate that the IDM can be used to predict the assembly time of products early in the design phase and performs best using a neural network trained using complexity vectors from high fidelity models.

To explore the potential for separating the objective handling times from the subjective insertion times, a Split Interference Detection Method is developed to use CAD part information to determine the handling time of the Boothroyd and Dewhurst assembly time estimation method and a modified complexity connectivity method approach is used to determine the insertion times. The handling and insertion times are separated because the handling times can be mostly determined using quantitative objective product information, while the insertion questions are subjective and cannot be quantitatively determined. The results suggest separation of the insertion and handling time does not reduce the percent error in estimating the assembly time of a product in comparison to the IDM. The handling portion of the SIDM can be used as a separate automated tool to determine the handling code and handling time of a product. The insertion portion of the Boothroyd and Dewhurst assembly time estimation method would still need to be calculated manually. The ultimate goal of this research is to develop and automated assembly time estimation method.

DEDICATION

This dissertation is dedicated to my parents, Zuhair and Basima Namouz, my siblings, Hani and Rana Namouz, and my soon to be wife Misty McDowell. Their support and love has provided me the motivation needed to finish this dissertation. Without my family, I would not have had this wonderful opportunity for higher education. Words cannot express my gratitude to all of my family.

الحاجة ام الاختراع

Necessity is the Mother of Invention

ACKNOWLEDGEMENTS

I would like to thank my entire advisory committee. Specifically, I would like to thank Dr. Joshua D. Summers. Dr. Summers is the main reason that I attended graduate school and he has challenged me since that day. Without Dr. Summers I would not be at this point. His advice and criticism have prepared for the next chapters of my life. I am happy to say that he is more than just an advisor, but a lifelong friend.

I would like to thank all members of the CEDAR lab. Discussions and collaboration with the member of CEDAR helped me further advance my education outside of the classes, and helped me form the basis for my research.

TABLE OF CONTENTS PAGE

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
ASSEMBLY TIME ESTIMATION METHODS: A REVIEW

Design for assembly (DFA) is a well-accepted technique that is based on empirical time studies and is used for analyzing products with the goal of reducing the assembly time [1–4]. One popular method within the larger set of DFA approaches is the assembly time estimation method developed by Boothroyd and Dewhurst [4]. This research explores opportunities in automating the design for assembly time estimation method.

Assembly time reduction has become a common focal point in an effort to reduce manufacturing costs [1–20]. Design for Assembly is an approach for reducing the manufacturing costs by improving the assemblability of a product [21]. Use of the design for manufacturing and design for assembly approaches can help reduce the cost of manufacturing, reduce component count, and increase quality, while increasing yield manufacturing output [4]. Implementation of various DFA methods has shown financial gain to industry based on assembly time reduction for a product between 50-75% [4]. A number of different methods including Methods Time Measurement (MTM), Lucas Method, Complexity Connectivity Method, Hitachi Method, and Boothroyd and Dewhurst DFA method have been developed to help aid designers in improving assembly [4,9,22,23]. Each of these DFA approaches contains a method to estimate assembly time.

The assembly time estimation methods of each approach can be further classified into two categories: process based or product based (see Table 1.1). A review of both process and product based approaches is included for completeness, but this research will focus on the product based approach.

Table 1.1: Summary of Design for Assembly Methods

Method	Citations	Stage of Design	Process/ Product Based	Information Required	Outcome/ Output
Methods Time Measurement	[9,24]	Redesign	Process	Assembly process and part geometry	Time or relative percentage
Lucas Method	[13,15,25,26]	Detail Design/ Redesign	Process	Part geometry, mass properties, part feeding	Manufacturing index (relative comparison)
Hitachi	[22]	Detail Design/ Redesign	Process	Product assembly steps	Assemblability score
Boothroyd and Dewhurst	[4]	Detail Design/ Redesign	Product	Part geometry and mass properties	Absolute Time or relative time
Complexity Connectivity	[23,27–31]	Detail Design/ Redesign	Product	Graphical representation of the product assembly	Absolute time or relative time

1.1 Process Based Assembly Time Estimation

The process based time estimates (Lucas, Hitachi, and MTM) are conducted by considering the operations or motions that are undertaken to assembly products [9,22,32,33]. These methods require minimal information about the parts themselves, but rather focus on the movements needed in the assembly process.

1.1.1 Hitachi Assemblability Method

The Hitachi Assemblability Evaluation Method (AEM) evaluates the ease of assembly of a product by using an assemblability evaluation score ratio (E) and assembly cost ratio (K) [33]. The assembly evaluation score ratio is determined based on the difficulty of each of the operations needed to assemble the product. The assembly cost ratio is used to project elements of the assembly cost. The Hitachi AEM is unique from the other DFA methods as it takes quality into account as well as reducing assembly cost. The Hitachi AEM categorizes most assembly into twenty elementary, but non-exclusive, assembly tasks [33]. The Hitachi AEM focuses on the insertion and fastening of components, while other methods such as Boothroyd and Dewhurst assembly method is focused on the handling of the parts as well as the insertion. Each part of an assembly is assigned a score indicating the difficulty of assembly for the part. All the parts of the assembly are then summed to give the assembly an overall assemblability score.

The Hitachi AEM, similar to the other methods, is implemented after a design has been created and then iterated on to improve assemblability. A flowchart showing the general sequence of analyzing an assembly using the Hitachi AEM is shown in Figure 1.1.

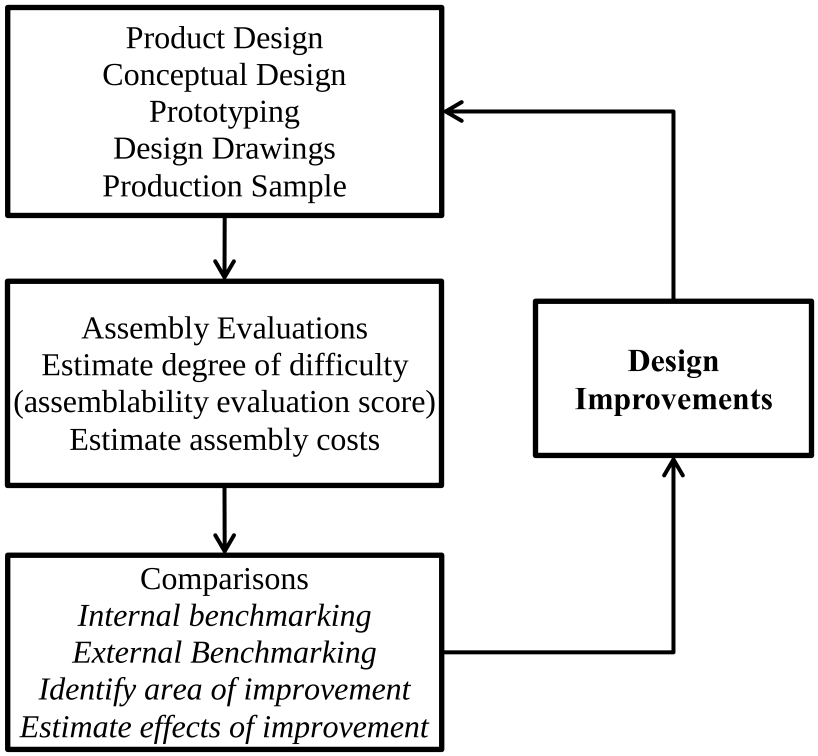


Figure 1.1: Hitachi Assemblability Method Flowchart (Adapted from [33]**)**

Once the initial product design has been created including the conceptual design, prototyping, and engineering drawings, a sample product can be created. The sample is then used to determine the assembly scores for each part which is used to estimate the assembly cost. The assembly score is then used to compare the new design to current designs within the company, as well as benchmark against products developed by other companies in terms of assemblability. Areas of potential improvement are identified and the ideas that show potential in improving assemblability are identified and improved upon. This process is an iterative process, so once design improvements are implemented, new engineering drawings and samples/prototypes can be created for re-evaluation.

1.1.2 Lucas Method

The Lucas Method, or more formally known as the Lucas Design for Assembly Method, is based on three separate sequentially conducted analyses: functional analysis, feeding analysis, and fitting analysis [33]. The first step, the functional analysis, requires that the parts be split into one of two groups. The “A” group is reserved for parts that perform a fundamental function. The “B” group is reserved for parts that are not essential to the assembly, such as fasteners [33]. A design efficiency factor (DE), can then be calculated using equation(1). The target efficiency for a product based on the design efficiency equations above is approximately 60% [33].

$$DE = A/(A+B)*100 \tag{1}$$

Where:

A: Number of parts that perform a fundamental function

B: Number of parts that are not essential to the assembly

The next part of the analysis is the feeding analysis. The feeding analysis portion is focused on the difficulty of handling parts before they are added to the system [33]. The feeding portion of the analysis is completed by answering a set of questions concerning the size, weight, handling difficulty, and orientation. The answers to each of these questions results in a handling index and a fitting index which can be found in the handling analysis table (Table 1.2) and the fitting analysis table (Table 1.3).

Table 1.2: Lucas Method Handling Analysis [33]

		Score
A	**Size and weight of part**	
	Very small, requires tools	1.5
	Convenient, hands only	1
	Large and/or heavy, requires more than one hand	1.5
	Large and/or heavy, requires hoist or two people	3
B	**Handling Difficulties**	
	Delicate	0.4
	Flexible	0.6
	Sticky	0.5
	Tangible	0.8
	Severely Nesting	0.7
	Sharp or abrasive	0.3
	Untouchable	0.5
	Gripping problem, slippery	0.2
	No handling difficulties	0
C	**Orientation of Part**	
	Symmetrical, no orientation required	0
	End to end, easy to see	0.1
	End to end, not visible	0.5
D	**Rotational Orientation of Part**	
	Rotational symmetry	0
	Rotational orientation, easy to see	0.2
	Rotational orientation, hard to see	0.4

The handling index is calculated by adding the score from each of the sections, A-D, of the handling table. The handling ratio can then be calculated from equation(2). The target value for the handling ratio is 2.5 [33].

$$\text{Handling Ratio} = \left(\text{Handling Index}\right) / \left(\text{Number of Essential Components } \left(\text{A}\right)\right) \quad (2)$$

The fitting index is determined from the fitting table (Table 1.3).

Table 1.3: Lucas Method Fitting Analysis[33]

		Score
A	**Part Placing and Fastening**	
	Self-holding orientation	1.0
	Requires Holding	2.0
	Plus one of the following:	
	Self-securing (snaps)	1.3
	Screwing	4.0
	Riveting	4.0
B	**Process Direction**	
	Straight line from above	0
	Straight line not from above	0.1
	Not a straight line	1.6
	Bending	4.0
C	**Insertion**	
	Single insertion	0
	Multiple insertions	0.7
	Simultaneous multiple insertions	1.2
D	**Access and/or vision**	
	Direct	0
E	**Alignment**	
	Easy to align	0
	Difficult to align	0.7
F	**Insertion Force**	
	No resistance to insertion	0
	Resistance to insertion	0.6
	Restricted	1.5

To determine the overall fitting index, each of the fitting scores for parts A-F are summed for each part. The fitting ratio can then be calculated by equation(3). The target value for the fitting ratio is 2.5 [33] .

$$\text{Fitting Ratio} = \left(\text{Fitting Index}\right)/\left(\text{Number of Essential Components }\left(\text{A}\right)\right) \qquad (3)$$

The third and final part of the analysis is the cost of manufacturing. This analysis does not return an absolute cost, but a relative cost that can be used to compare parts and manufacturing processes [33]. The following part manufacturing cost index can be calculated from:

$$M_i = R_c P_c + M_c \tag{4}$$

Where:

$R_c = C_c C_{mp} C_s * (C_t or C_f)$: Relative Cost
R_c = Complexity Factor
C_{mp} = Material Factor
C_s = Minimum Section
C_t = Tolerance factor
C_f =Finish Factor
P_c =Processing cost
$M_c = VC_{mt}$
V = Volume (mm3)
C_{mt} = Material Cost
W_c = Waste coefficient

While the Lucas method can be used as a relative tool to compare multiple design ideas, it does not provide an absolute assembly time estimate. It does however provide a manufacturing index, which many of the other DFA methods do not provide.

1.1.3 Methods-Time Measurement

The Methods-Time Measurement (MTM) method assembly time estimation method is based on the movements that an operator makes when assembling a product [9]. The MTM methods (developed by HB Maynard) is just a portion of a larger set of Methods Engineering developed by Frederick Taylor and Frank Gilbreth in the early 20th century [9]. Methods Engineering involves investigating every operation on a product to eliminate any unnecessary actions and optimize the work process [9,24]. Based on the investigation of all the necessary operations needed to complete work on a product, the time required for a standard worker to complete the job can be estimated. Specifically, Methods-Time Measurement is defined as:

> *"procedure which analyzes any manual operation or method into the basic motions required to perform it and assigns to each motion a predetermined time standard which is determined by the nature of the motion and the conditions under which it is made"* [9]

MTM is one of the first attempts at creating a tool to enable engineers to estimate assembly times without the need for stop-watch time studies, specifically to support product analysis before production [9]. The motion data originally started from analysis of shop workers using a drill press or fixture loading and positioning jig under spindle [9]. As the worker was conducting the work, they were being filmed the entire time and investigator was asked to fill out a methods analysis sheet which required information such as but not limited to:

- Date
- Part
- Material
- Description of Operation
- Machine Description
- Description of Method
- Diameter of tool
- Depth
- Speed
- Feed

From the collected work data (analysis sheet), the most observed motions or operations were noted for a total of 60 operations [9]. Using empirical data collected, the estimated time to complete the routines were measured. The operations mentioned were combined into tabular form, and were broken down into the most basic forms of motion and are incorporated into the seven main tables developed:

1.Reach
2.Move
3.Turn (Including apply pressure)
4.Grasp
5.Position
6.Disengage
7.Release

Seven tables were created to classify each of the above motions with additional detail. For reference, the table for grasp has been recreated (see Table 1.4). The time for each operation is measure in a time measurement unit (T.M.U.). One TMU unit is equivalent to 36 milliseconds.

Table 1.4: MTM Grasp Table (Adapted from [9])

Grasp		
Case	**Description**	**Time T.M.U**
1a	Pick up grasp- Small, medium, or large object by itself – easily grasped	1.7
1b	Very small object or tool handle lying close against flat surface	3.5
1c	Interference with grasp on bottom and one side of object	8.7
2	Regrasp	5.6
3	Transfer Grasp	5.6
4	Object jumbled with other objects so that search and select occur	8.7
5	Contact, sliding or hook grasp	0

The MTM method has served as a basis for supporting automated assembly time estimation for an automotive OEM [34,35]. This method has been augmented with a controlled vocabulary of assembly verbs and activities, building on previous work that seeks to demonstrate that the free text description of assembly activities can predict assembly times [29,36].

1.2 Product Based Assembly Time Estimation

The product based approaches (Connectivity Complexity and Boothroyd and Dewhurst) are based on the products themselves and do not require extensive knowledge of the assembly process [4,23,29]. To clarify, Boothroyd and Dewhurst does require knowledge of the assembly of one product to the other to determine insertion times, but does not require knowledge of the process to accomplish it such as where the parts are located on the assembly line and if the worker must walk to retrieve the parts before assembly.

1.2.1 Boothroyd and Dewhurst DFA

One method developed by Boothroyd and Dewhurst estimates the assembly time of a product by focusing on estimating a handling time and an insertion time. A user implements the assembly time estimation method by navigating a set of hierarchical charts in which each level requires additional information about the part to be input by the user [37]. The information provided by the user about the part determines the route that will be travelled down the chart, resulting in a handling code and insertion code, from which the user can directly retrieve the associated assembly times. The handling time and insertion time are then summed to determine the overall assembly time of a part.

Boothroyd and Dewhurst empirically developed a set of charts that are used to estimate the assembly time of different products [4]. The charts are used to estimate the assembly time of a product based on two categories: handling and insertion. The user determines a two-digit handling code based on part information such as number of hands needed to handle, the size of the part, and whether the parts nested or tangled together. The two-digit code can then be used to determine the estimated handing time of the part. The same procedure would be followed to determine the insertion time of the part. The two times would then be summed to determine the total assembly time for that part. This is repeated for all the parts of a system to determine the assembly time of the complete system. Typically the best values of the charts, such as the lowest assembly times, are found in the upper left corner while the assembly time generally increases towards the lower right corner [38] (see Table 1.5).

Table 1.5: One Hand Handling Chart [4]

		Parts easy to grasp and manipulate					Parts present handling difficulties				
		T > 2mm			T ≤ 2 mm		T > 2 mm			T ≤ 2 mm	
		S > 15 mm	6 mm ≤ S ≤ 15mm	S < 6 mm	S > 6 mm	S ≤ 6 mm	S > 15 mm	6 mm ≤ S ≤ 15mm	S < 6 mm	S > 6 mm	S ≤ 6 mm
		0	1	2	3	4	5	6	7	8	9
(α+β) < 360	0	1.13	1.43	1.88	1.69	2.18	1.84	2.17	2.65	2.45	2.98
360 ≤ (α+β) < 540	1	1.5	1.8	2.25	2.06	2.55	2.25	2.57	3.06	3	3.38
540 ≤ (α+β) < 720	2	1.8	2.1	2.55	2.36	2.85	2.57	2.9	3.38	3.18	3.7
(α+β) = 720	3	1.95	2.25	2.7	2.51	3	2.73	3.06	3.55	3.34	4

The tables are a collection of historical time data for assembly of different components. A portion of the handling table is shown below in a decision tree type of representation (Figure 1.2) and based on a choice the user makes reveals more possible decisions until the user arrives at the associated handling or insertion code.

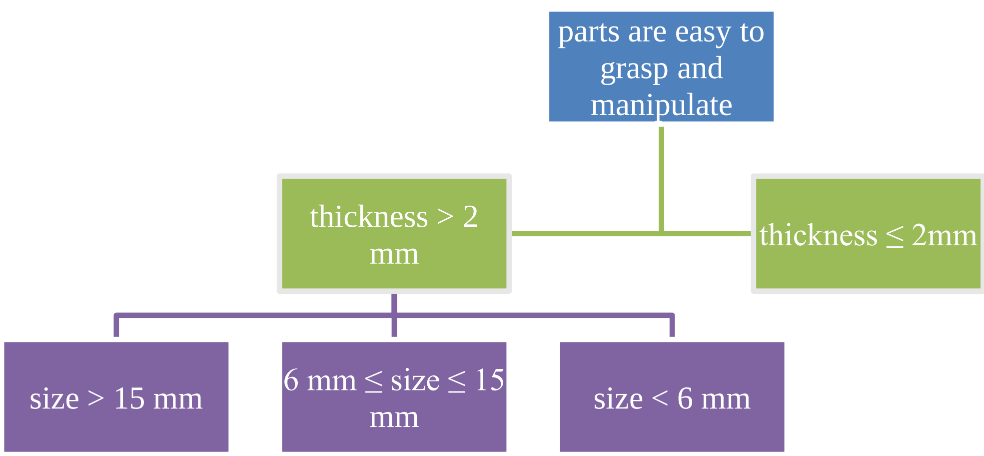


Figure 1.2: Partial Handling Code Decision Tree (Adapted from [4]**)**

The time estimate charts are a manual method to estimate the assembly time of different parts. Boothroyd and Dewhurst Inc. have implemented the time estimate method into a computer tool that can assist designers in estimating assembly time[1].

1.2.2 Complexity Surrogate Modeling

The term complexity is used in various fields including engineering design, supply chain management, manufacturing, operations management, and assembly [39–48]. Furthermore, these areas of complexity can be further generalized into market complexity, product complexity, process complexity, and organizational complexity [39] (see Figure 1.3).

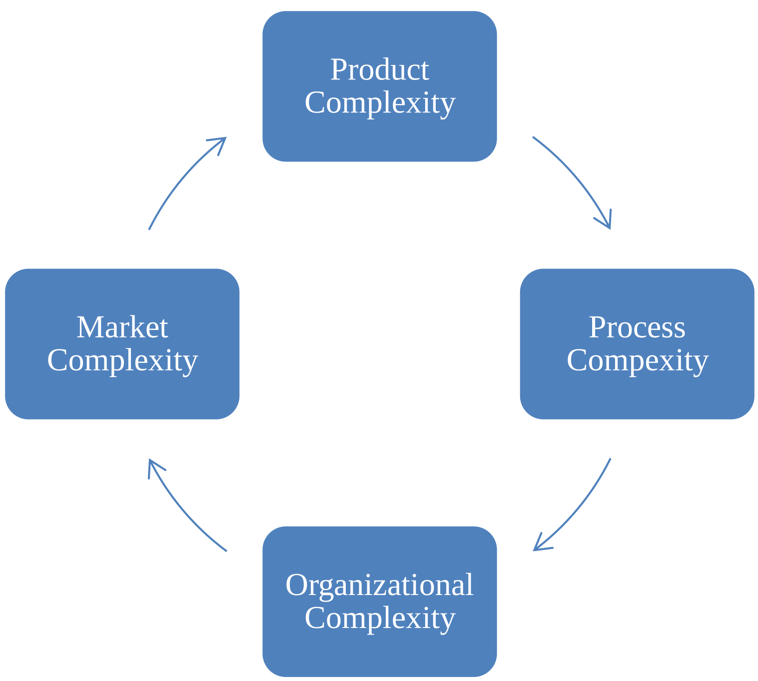


Figure 1.3: Aspects of Complexity (Adapted from [39]**)**

[1] http://dfma.com/ , accessed on 2/19/2012

All of these aspects of complexity are interrelated, while the definition of complexity varies between the different organizations. Product complexity is generally used to represent the interrelatedness of an assembly, or the geometry that composes a part [49–52]. Recent research has used complexity representations to model and operate on difference phases of engineering design. For example, product complexity has been used as a surrogate for a number of computer aided design tools including design for manufacturing and design for assembly [41,51,53–56]. Specifically the focus of this research is on the use of complexity as a surrogate model for assembly time estimation.

1.2.2.1 Complexity Connectivity Method

The complexity connectivity method uses a complexity vector composed of twenty-nine complexity metrics to estimate the assembly time of a product [23,27,28,30,31] (see Table 1.6).

Table 1.6: Complexity Metrics

Complexity Metrics	Size	Dim		Elements
				Relations
		Conn		DOF
				Connections
	Interconnection	Shortest Path		Sum
				Max
				Mean
				Density
		Flow Rate		Sum
				Max
				Mean
				Density
	Centrality	Betweenness		Sum
				Max
				Mean
				Density
		Clustering Coefficient		Sum
				Max
				Mean
				Density
	Decomposition	Ameri Summers		
		Core Numbers	In	Sum
				Max
				Mean
				Density
			Out	Sum
				Max
				Mean
				Density

The complexity metrics are calculated based on the bi-partite representation of a product (See Figure 1.4). For brevity, the discussion, details, and calculations of the complexity metrics are not included here but can be found in previous literature [23,28,30].

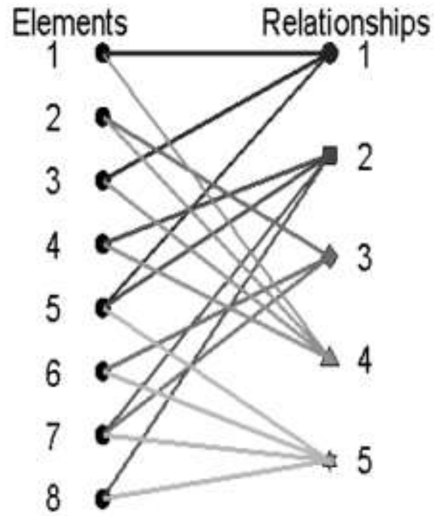


Figure 1.4: Bi-partite Graph [28]

Initially the complexity connectivity method used a linear regression to model the relationship between the complexity metrics and the assembly time of a product [23]. To improve the predictive ability of the connectivity complexity method, the relationship model evolved from a linear regression to an artificial neural network [31]. The ANN complexity connectivity method (ANN-CCM) is trained using the complexity vector of a product as the input into the neural network and the known assembly time is the training target. The neural network is used as a data mining tool to find the relationships between the complexity vector and the known assembly time to create predictive models. The use of the artificial neural network was shown to improve the predictive ability of the method over initial regression fitting attempts [57], however the manual bi-partite graph generation was still time consuming and inherently subjective due to manual creation [27,29,31].

1.2.2.2 Complexity Graph Generation- Assembly Mate Method

The original complexity connectivity method (CCM) manually created the bi-partite graph, but due to the extensive effort required to create the bi-partite graphs, an automated graph generation method is desired. The next improvement to the complexity connectivity method was an automated graph generation tool based on the mates used to create the assembly model [30].

The Assembly Mate Method (AMM) uses SolidWorks (SW) assembly mate information to create the connectivity graphs needed for the complexity connectivity method [30]. The mates in SW are the relationship that a user specifies to locate a part in the model relative to another part, assembly, or model feature such as a coincident mate or concentric mate (see Figure 1.5 for additional standard SolidWorks mate types).

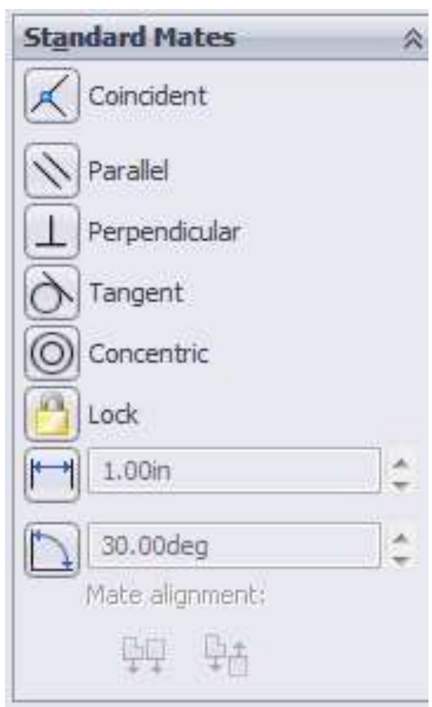


Figure 1.5: Standard SolidWorks Mates

The mate features create a relationship between two components and SolidWorks retains this relationship information as a parent/child relationship. For example, consider a block with a circular hole and a pin (see Figure 1.6).

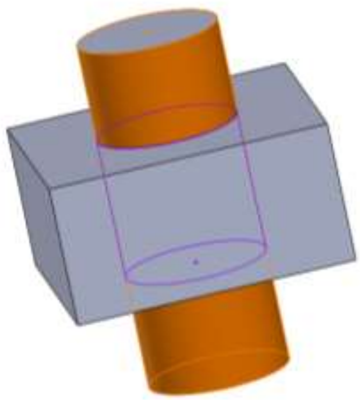


Figure 1.6: Block and Pin Assembly

The automated graph generation tool uses the "Parent/Child Relationship" information to find the connections between parts in the assembly (see Figure 1.7) [30]. The concentric relationship exists between the "Block-1" and the "Pin-1" (see Figure 1.7).

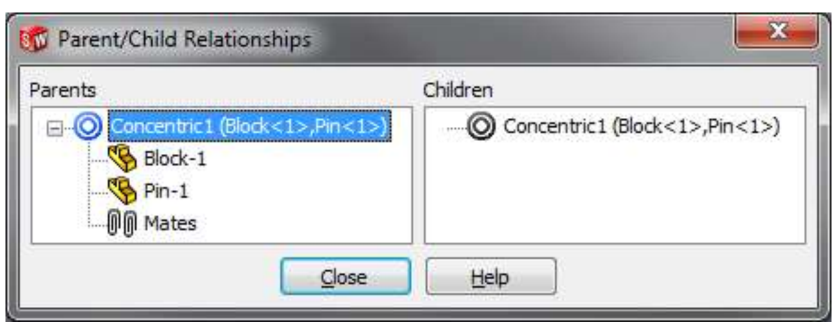


Figure 1.7: Parent-Child Relationship

The assembly mate method iterates through every mate in the assembly to create a list of parent/child relationships. The list of parent/child relationships is output as a text file which is parsed to create the bi-partite graph to find the values of the complexity vector. The AMM is able to quickly create the relationship between parts in a product based on the assembly mates; however the method still has a few limitations [58]. One limitation of the AMM is its inherent variability based on the designer that created the assembly model. An assembly model can be mated together in numerous ways based on the designer. One designer may use different assembly mates when creating the model compared to another designer, and this would result in different parent child relationship lists. Another current limitation of the AMM is that it requires an assembly model created in SW with all of the assembly mates included. Ideally the system would be able to supports multiple CAD platforms and file types, including standard CAD exchange file types for collaboration between companies.

The time and information input needed to conduct the aforementioned DFA assembly time estimation methods provide motivation for an automated assembly time estimation method. This thesis will focus on the development and testing of an automated assembly time estimation method.

CHAPTER TWO
DEFINING THE RESEARCH MOTIVATION

To increase profit margins, companies are continually looking for ways to decrease the cost of products. One major area of focus in reducing the cost of the product is by reducing assembly costs. This motivated the development and application of design for assembly approaches and guidelines [4,9,10,49,50]. These guidelines are used as the basis for improving product design with a specific focus of decreasing the required assembly time. To assess and measure the gained benefits of applying these DFA guidelines, a way to measure the expected assembly time savings is desired. Multiple methods have been developed that can estimate the assembly time of a product including the ones discussed in Chapter One (Hitachi Method, Lucas Method, Methods Time Measurement Method, Boothroyd and Dewhurst Method, and the Complexity Connectivity Method).

While previous research has shown the large potential benefits of applying DFA methods, the analysis time required in analyzing products has discouraged application of the methods [4,49]. Specifically, estimating the assembly time of a product before and after application of DFA methods is very tedious and time consuming [4,27,49]. Another limitation of the identified assembly time estimation methods is the amount of detail required. The identified methods are generally applied as a redesign approach or during detailed design when market ready prototypes have been prepared. The assembly time estimation methods required detailed information about either the process with specific body movements required for the product assembly or the product based on geometry, size, and symmetry to estimate assembly time. This dissertation is focused on designing a tool which automates the assembly time estimation of products. The goals of which is to increase the accuracy and repeatability of the assembly time estimation, decreasing the analysis time, and reducing the amount of information required by the designer to perform the analysis.

The Complexity Connectivity Method and the Boothroyd and Dewhurst assembly time estimation method will be used as the backbone of this research. A visual representation of the Complexity Connectivity Method process will help to illustrate the research that was conducted and how it impacts the overall process (see Figure 2.1).

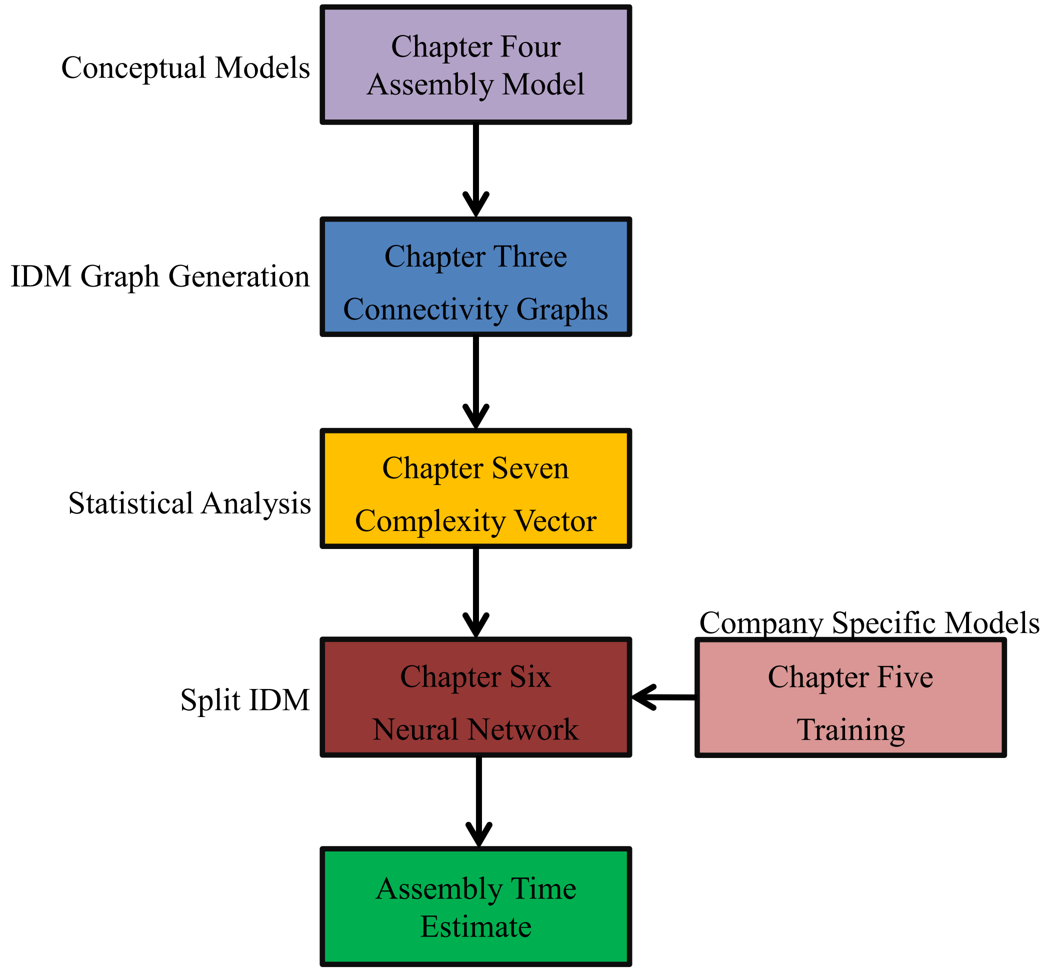


Figure 2.1: Complexity Connectivity Process Flowchart with Research Contributions

The summarized complexity method (illustrated in Figure 2.1) starts with an assembly model either represented in CAD or a physical model. The assembly model is used to form the connectivity graph based on connections between parts in the assembly. The connectivity graphs are then operated on to calculate a complexity vector consisting of 29 metrics that represent the assembly. The complexity vector is then used as the input into a neural network to estimate assembly time. The neural network is trained using the complexity vectors of products with known assembly times, or assembly times estimated by the manual Boothroyd and Dewhurst assembly time estimation method. This research will study different aspects pertaining to each step of the complexity connectivity method.

The experiments described in this research are not conducted in the same chronological order as the process of the complexity connectivity method. First, the Interference Detection graph generation method (IDM) is created and tested (see Chapter Three and Chapter 0). The IDM is a new method to create the connectivity graphs needed to calculate the complexity vector of an assembly. The IDM will use part connections to create the connectivity graph required as the input into the neural network to estimate assembly time (Figure 2.2).

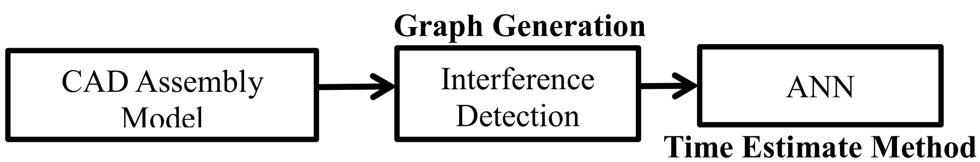


Figure 2.2: Interference Detection Method (IDM)

With increasing interest in developing design tools for use early in the design phase, the ability of the IDM to estimate the assembly time of products during the conceptual design phase will be analyzed (see Chapter Four) [59]. Part and assembly models are altered to represent low fidelity models which can be expected when little detail is known about the product. The general process used for the IDM will be used in this portion of the research, but a low fidelity model will be used as the input to the ANN (see Figure 2.3).

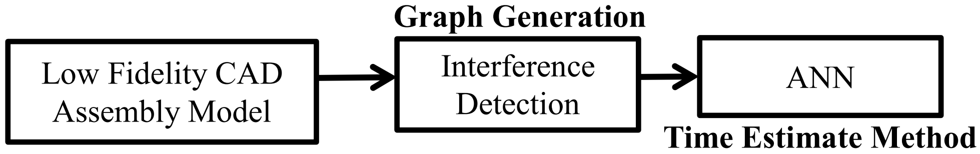


Figure 2.3: Interference Detection Method (IDM) for Conceptual Design Stage

The next portion of research is focused on the ANN training. A set of products and actual assembly times were supplied by a local power tools manufacturer. An ANN is trained using only products supplied by the power tools company and is compared to an ANN that was trained on a variety of consumer electromechanical products. The ANNs are compared to determine if an ANN trained with company specific products can better estimate the assembly time of products from within that company, rather than an ANN trained on a wide variety of general products (see Chapter Five).

The Complexity Connectivity Method is an alternate means to the Boothroyd and Dewhurst assembly time estimation method to calculate assembly times of a product. While the Boothroyd and Dewhurst assembly time estimation method is widely accepted in academia and industry, the information needed to conduct the analysis hindered the automation of the method. The Boothroyd and Dewhurst assembly time estimation method is composed of a handling time and an insertion time. The handling time is mostly quantitative while the insertion time is mostly qualitative (see Chapter 1.2.1). The qualitative nature of information needed to determine the insertion portion prevented automation of the method, and motivated the development of the complexity connectivity method [27,57]. This portion of the research aims to determine if the complexity connectivity method and the Boothroyd and Dewhurst assembly time estimation method can be combined to create a single tool that outperforms the complexity connectivity method alone (see Chapter Six). The quantitative handling time will be calculated by retrieving part information from the CAD model, and a modified complexity connectivity method will be used to determine the insertion time (see Figure 2.4).

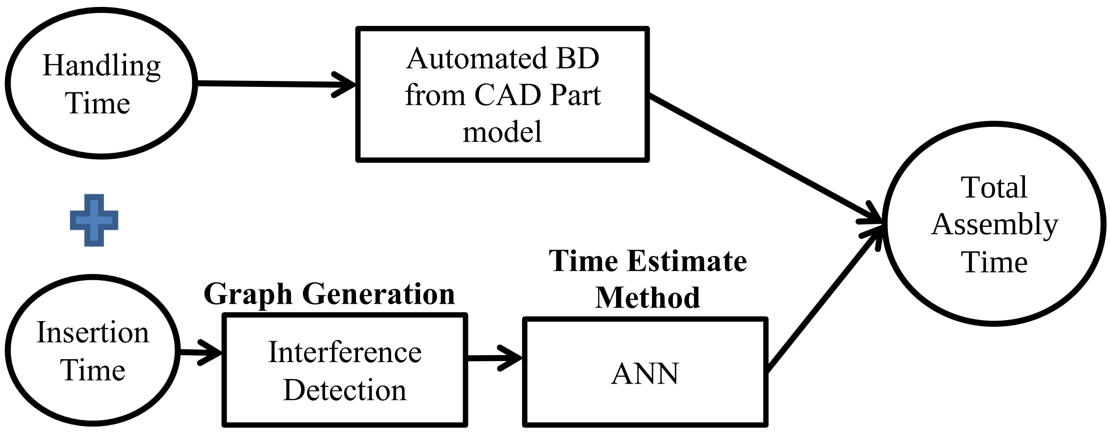


Figure 2.4: Split Interference Detection Method

The final portion of this research will explore the twenty nine complexity metrics which compose the complexity vector. The complexity vector was introduced with the complexity connectivity method; however no work has been conducted to determine if all of twenty nine complexity metrics are necessary to represent a product beyond the initial subjective reduction to three metrics [23]. A statistical analysis is used to try to reduce the number of complexity metrics needed in the complexity vector, to ultimately reduce the computational effort required by the use of the complexity vector as a surrogate for assembly time estimation (see Chapter Seven).

2.1 Research Questions

The focus of this research is designing and implementing a method to automate product assembly time estimation. The Boothroyd and Dewhurst [4] assembly time estimation method and the complexity connectivity method [23,29–31] will be used as the backbone of this research. Previous research has indicated that the use of CAD platforms is replacing sketching early in the design process [59]. In an effort to design this method for application throughout the design process, including early conceptual design, the focus will be on product based approaches. The product design is captured using CAD and can be used as a source for analysis. A commercial CAD package (SolidWorks) will be used to retrieve information about parts and assemblies to determine handling and insertion codes for the Boothroyd and Dewhurst assembly time estimation method. In developing and implementing the tool, a number of research questions will be answered regarding the assembly time estimation method and the capabilities of implementing assembly time estimation with support from a CAD system:

Can an assembly time estimation method be automated to estimate product assembly time based on the CAD models (part and assembly files)? If so, what information is needed and where will this information be retrieved from?

RQ1: How much variability can be expected in the current Boothroyd and Dewhurst assembly time estimation method? Answering this research question motivates the need for an assembly time estimate that is both accurate and repeatable. The automated assembly time should be able to accurately estimate the assembly time of a product without variation caused by detailed subjective user inputs.

RQ1.1. Is the predicted 50% variability indicated by Boothroyd and Dewhurst an accurate variability estimate when the method is applied by students to an existing product?

RH1.1 The variability of the Boothroyd and Dewhurst Manual Assembly Time Estimation method is less than or equal to 50% [4].

RQ2: How can the connectivity complexity method be improved to provide more accurate and repeatable assembly time estimates? The current complexity method which utilizes the Assembly Mate Graph Generation method [30] is dependent on the designer that has created the model. Answering this question will provide an automated method that is not dependent on the designer that creates the model, while maintaining or improving the accuracy of the time estimate.

RQ2.1. Can the accuracy and repeatability of the complexity connectivity method be improved by providing a method to objectively create the part connections graph independent of designer definition of assembly mates [30]?

RH2.1 The accuracy and repeatability of the complexity connectivity method can be improved by creating assembly connectivity graphs based on physical locations and part interference in the assembly model space instead of depending on a designers definition of assembly mates [30].

RQ2.2. Which complexity metrics have the largest influence on the assembly time estimation?

RQ2.2.1. Are all of the current complexity metrics required to achieve an acceptable (within 50%) time estimate?

RQ3: Can the Boothroyd and Dewhurst assembly time estimate method be automated by retrieving part and assembly information from a CAD model? Answering this question provides a tool to help designers analyze products and estimate the expected benefits of the proposed design for assembly efforts. Currently the assembly time estimation method is tedious and time consuming resulting in resistance to application of design for assembly.

RQ3.1. Does this automated method provide an improvement in assembly time estimate over the current complexity method and Boothroyd and Dewhurst time estimate method in terms of: accuracy, repeatability, and computation time, and level of detail of information input?

RH3.1 The Boothroyd and Dewhurst assembly time estimation method can be automated by separating the handling and insertion time estimates. The objective information that is required to determine a handling code can be directly retrieved from the part models. The subjective insertion information will be determined by using the assembly model to create part connectivity graphs and using a modified complexity connectivity method to estimate the insertion time. With the combination of the two methods, an improved assembly time estimate method can be automated that is more accurate, repeatable, requires less analysis time, and less detailed input information than the other estimate methods.

RQ4: Can a modified complexity connectivity method, utilizing the interference detection method to create connection graphs be used to estimate assembly times of products in the conceptual design phase, based on low fidelity CAD models? Answering this question will provide designers a tool that can used early in the design process when detailed part information is not known. A tool that can be used early in the design process or in the conceptual phase of design can support design for assembly through the design process as opposed to only in the detailed design phase or as a redesign tool.

2.2 Research Roadmap

The table below summarizes the research questions that are answered, the topic of the research question, the expected research method to be used to answer the research question, and the deliverable (see Table 2.1).

Table 2.1: Research Questions

Research Question	Topic	Research Method	Deliverable	Included in:
RQ1				
RQ1.1	Variability of the Boothroyd and Dewhurst Assembly Time Estimation Method	Survey and User Study (ME455 Pen Study)	[27,60]	Chapter Two
RQ2				
RQ2.1	An Objective Connectivity Graph Creation Method	Statistical Test of assembly time estimates	[61,62]	Chapter Three
RQ2.2	Main Complexity Factors for Estimating Assembly Time	Statistical Analysis (Factor Analysis)	Dissertation	Chapter Seven
RQ3				
RQ3.1	Automated Assembly Time Method - Algorithm and Demonstration	Separation of Handling and Insertion Times	Chapter Six	Chapter Six
RQ3.2	Automated Assembly Time Estimation Method: A Validation Study	Test Cases (Industry models and actual assembly times)	Dissertation	Chapter Five
RQ4	Conceptual Models	Test Cases	[63]	Chapter Four

To explain the formation of this research and the research questions associated with it, a brief review of the previous work completed in the CEDAR (Clemson Engineering Design Applications and Research) Group is provided (Figure 2.5).

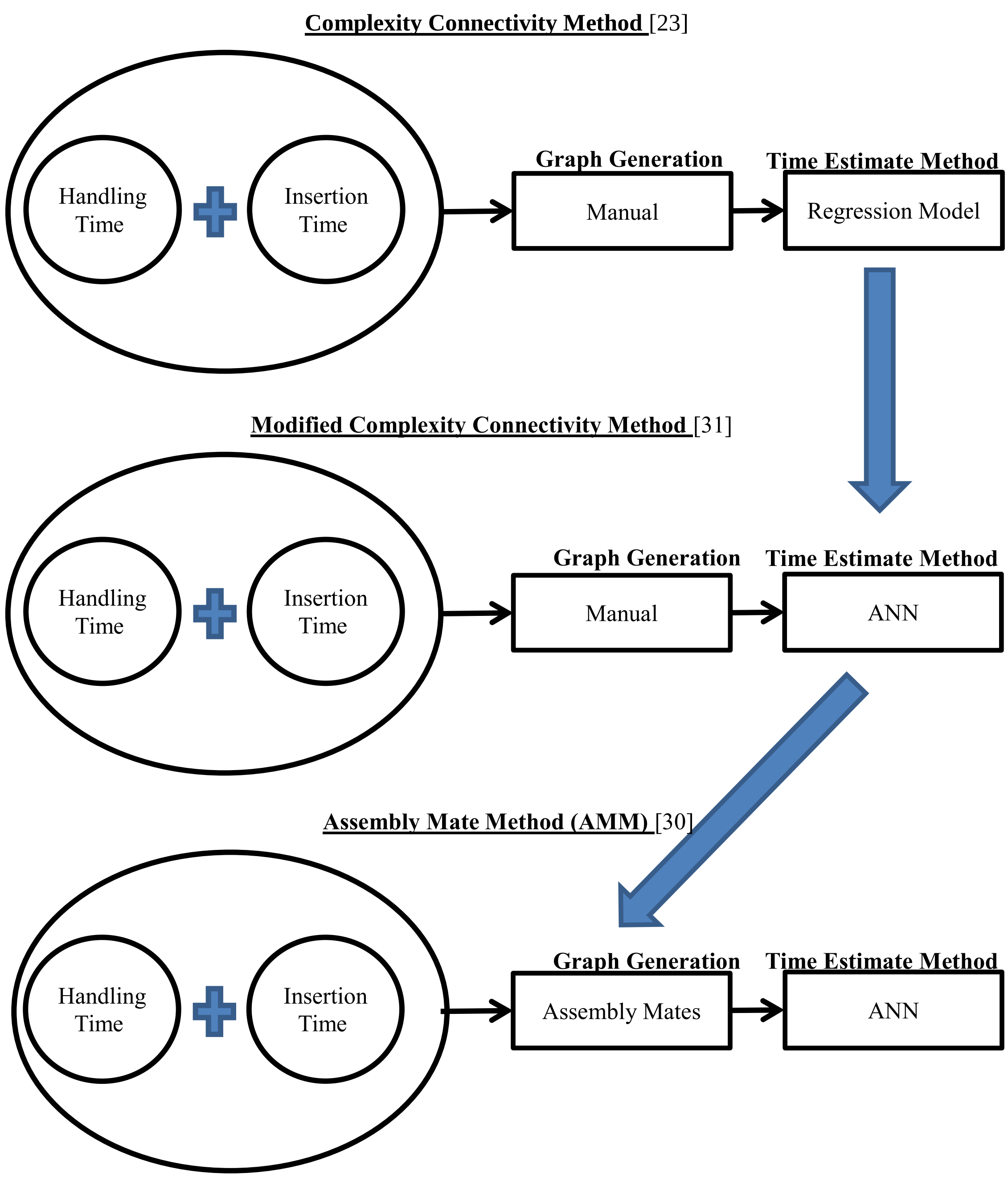


Figure 2.5: Progression of Connectivity Complexity Method (previous work)

The AMM uses part mating information from CAD assembly models to create the part connectivity graphs. While this is an improvement in the complexity method, the part connection graphs are still dependent on the designer that created the model and the types of mates they chose. The next transformation, Interference Detection Method, uses part interference to create the part connectivity graphs. The next step of this research Split Interference Detection Method (SIDM) will separate the insertion and handling times, which together form the total assembly time (see Figure 2.6). Information from the part CAD model will be used to determine the handling time, and a new ANN will be trained to estimate only the insertion time based on the part connectivity graphs.

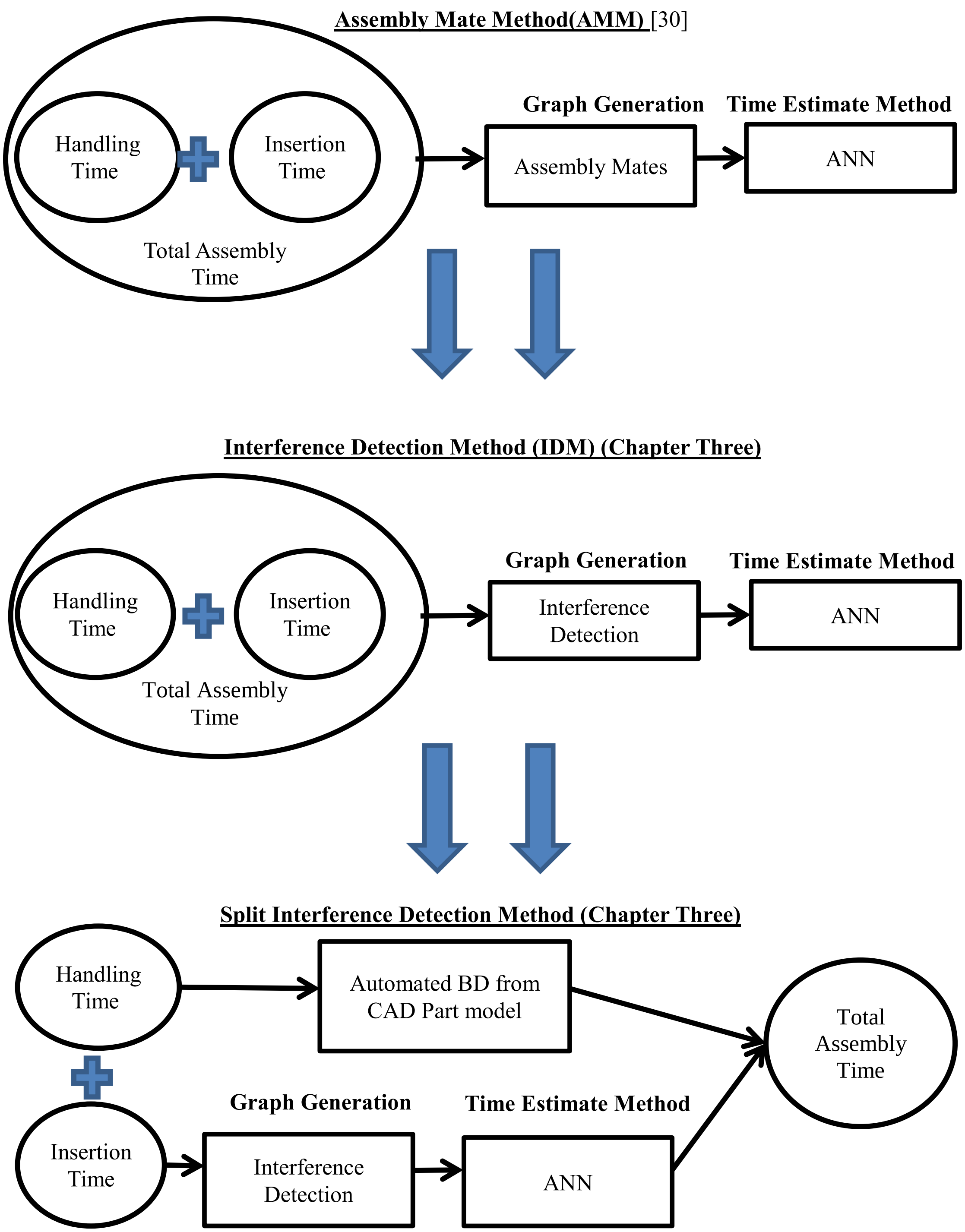


Figure 2.6: Progression of Split Interference Detection Method

To explore the variability inherent in the Boothroyd and Dewhurst manual assembly time estimation method and provide motivation for an automated assembly time estimation method, a pilot study was conducted to estimate the assembly time of a clicker pen using the manual charts.

2.3 Exploratory Study

An integrated senior and graduate level mechanical engineering class was trained on the Boothroyd and Dewhurst method and assembly time estimate charts as part of a design for manufacturing course (ME 455/655). The students in the course were asked to complete an assembly analysis and estimate assembly time of a Pilot G-2 clicker pen (Figure 2.7) using the manual assembly time estimation charts. This study was approved under IRB2012-250.

Figure 2.7 Fully Assembled Clicker Pen[2]

2.3.1 Participants

The participants for the pilot study consisted of students from a senior and graduate level mechanical engineering manufacturing course. The students were allowed to divide amongst themselves into groups of two. The students were trained in the two previous lectures, each lasting one hour and fifteen minutes, on the use and application of the assembly time estimate method. The students were all similarly trained with the method, and considered to be comparable in experience to an entry level manufacturing

[2] http://www.officespecialties.com/pilot_31277_g2_ultra_fine_retractable_pen_42038_prd1.htm, accessed on 2/19/2012

engineer. Training for application of the method for an engineer may be conducted in a similar fashion, based on books or passed on from another engineer. One option that Boothroyd and Dewhurst offer is a special course in assembly time estimation. The course should improve the repeatability and use of the method by the engineer, but also has a number of drawbacks including cost and time required for training[3]. The instructor applied the method during a lecture to a pneumatic piston for demonstration purposes. The pen is the first assembly that the students analyzed independently, although the instructor was available to answer general questions on application of the method, but not any specifics on how to analyze the assembly or on the handling or insertion codes to choose for the different parts of the pen. The students conducted the time estimate in-class, and the assignment would count as an "In-class Activity", which as a category is worth 20% of the students' overall grade. This was not the first or last in-class activity that the students were given, so this particular assignment was typical and stylistically familiar to the students. A total of twenty groups were formed for the in-class assignment.

2.3.2 Process

In a Design for Manufacturing course (ME455) at Clemson University, students were asked to apply the Boothroyd and Dewhurst manual assembly estimation method to a Pilot G-2 Clicker Pen (Figure 2.7). The students were allowed a time limit of one class period (60 minutes) to complete the analysis with 15 minutes reserved for class discussion on the results. Each student group had a pen that they were allowed to

---

[3] http://www.dfma.com/services/dfmacore.htm

disassemble and reassemble to complete the assembly time estimate. Each individual group discussed the assembly time estimate and, consensus was reached, the group completed the worksheet. The students were provided a basic template to record the handling and insertion codes, as well as the handling and insertion times for each part, and additional cells to show the sum of the handling and insertion times for each of the individual parts resulting in a total assembly time. An example of a completed results table is shown in Table 2.2.

Table 2.2: Example Student Clicker Pen Time Estimate

Task	Description	Handling Code	Handling Time (s)	Insertion Code	Insertion Time (s)	Total Time (s)
1.1	Top	30	1.95	00	1.5	3.45
1.2	Bottom	10	1.5	00	1.5	3
1.3	Button	11	1.8	00	1.5	3.3
1.4	Cartridge	10	1.5	00	1.5	3
1.5	Spring	83	5.6	00	1.5	7.1
1.6	Base	10	1.5	38	6	7.5
1.7	Grip	10	1.5	31	5	6.5
					Total Assembly Time	33.85

2.3.3 Results

A summary of the results of the pilot study, including the handling time, insertion time, and total assembly time of the pen from the different groups is summarized in Table 2.3.

Table 2.3 Pen Data from In-Class Activity

Group	Handling Time (s)	Insertion Time (s)	Total Assembly Time (s)
1	11.77	25.50	37.27
2	15.69	16.00	31.69
3	8.58	25.35	33.93
4	14.03	16.50	30.53
5	15.83	18.00	33.83
6	17.10	24.50	41.60
7	17.10	24.50	41.60
8	13.03	24.00	37.03
9	11.77	25.50	37.27
10	11.92	29.10	41.02
11	12.60	26.00	38.60
12	12.51	19.50	32.01
13	14.14	23.50	37.64
14	7.45	16.50	23.95
15	11.14	12.50	23.64
16	13.40	18.00	31.40
17	13.70	26.50	40.20
18	10.05	17.00	27.05
19	13.39	31.50	44.89
20	15.35	18.50	33.85

The results of three of the groups (groups 3, 10, 18), shaded in Table 2.3 were eliminated due to incorrectly identifying a handling code for an insertion code or vice versa leaving a total of seventeen groups. For example, group 3 provided an insertion code of "87" with an associated insertion time of 5.85 s. The insertion charts do not include a value for an insertion code of "87", and to ensure the students did not flip the designation of "row * column", the value of insertion code "78" was also examined, recognizing that it also does not correspond to a value included in the insertion charts. However, a handling code of "87" does exist, and is associated with a time of 5.85 s. Each part requires a separate handling code and insertion code, and the two cannot be interchanged. While this is an error in the application of the method, this is not specifically the focus of this research and those values would influence the results. Therefore this, and similar results, were eliminated from the analysis.

A statistical analysis of the results of the data shown above, excluding the three cases which were eliminated due to circumstances discussed earlier is summarized in (Table 2.4).

Table 2.4 Clicker Pen Assembly Statistics

	Handling Time	Insertion Time	Total Time
Average	13.53	21.59	35.12
Standard Deviation	2.38	5.03	5.88
Max	17.10	31.50	44.89
Min	7.45	12.50	23.64
Range	9.65	19.00	21.25

The assembly time estimation for the clicker pen resulted in an average of 35.12 seconds and a range of 21.25 seconds. This suggests that multiple users that are equivalently trained and provided with the same product did not arrive at the same estimated assembly time. Observations of the data suggest that the decisions that the user makes to the Level 1 subjective questions for handling and insertion, contributes to the variation in assembly time estimates.

To determine the influence of answering the subjective question on the assembly time estimate, the alternate possible handling and insertion times assuming that Level 1 subjective question was answered alternatively was retrieved. The average of the two values was then used as the time estimate. This serves to simulate the user not having to answer the subjective question, but instead using the average value that could result. The maximum and minimum values of the alternate decision were also investigated, but resulted in values that exaggerated the variability of the method. The average value is used as a middle value to represent the user not making the decision and as a baseline time for this subjective question to add into the time analysis.

This process is repeated for each handling time and insertion time for each group to determine the effect of estimating the assembly time of the pen, while replacing the Level 1 subjective values with the average of the two values. The results of each group's initial assembly time estimate, and the derived estimate using the average of the two subjective values is shown in Table 2.5.

Table 2.5: Total Assembly Time Comparisons

Group	Total Assembly Time (s)	Total Assembly Time using average of Level 1 Subjective Question	Percent Difference
1	37.27	38.67	3.8
2	31.69	35.95	13.4
4	30.53	43.60	42.8
5	33.83	50.78	50.1
6	41.60	45.00	8.2
7	41.60	45.00	8.2
8	37.03	41.72	12.7
9	37.27	38.67	3.8
11	38.60	46.62	20.8
12	32.01	35.40	10.6
13	37.64	45.30	20.3
14	23.95	28.62	19.5
15	23.64	27.05	14.4
16	31.40	35.11	11.8
17	40.20	41.42	3.0
19	44.89	49.43	10.1
20	33.85	38.50	13.7

The basic statistics of the total assembly time using the average of Level 1 subjective questions indicates a mean of 40.4 seconds, with a standard deviation of 6.65 seconds which is larger than the student assembly time standard deviation (see Table 2.6).

Table 2.6: Statistical Comparison of Data Sets

	Student Assembly Time	**Assembly Time using Average of Level 1 Subjectivity**
Average	35.12	40.40
St. Deviation	5.88	6.65
Max	44.89	50.78
Min	23.64	27.05
Range	21.25	23.74

A statistical normality test (Anderson-Darling) was conducted on each set of data to ensure that each data set was normally distributed. The resulting p-values of the student estimates and the average of Level 1 subjectivity estimates are $p = 0.49$ and $p = 0.67$ respectively. This is required to justify the use a probability distribution plot to represent the data. A curve is fit to both sets of data and the resulting density plot is shown in Figure 2.8.

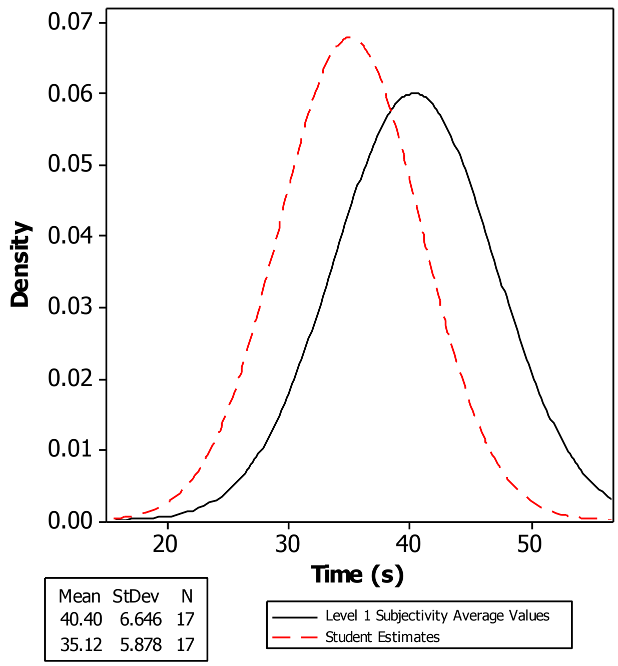


Figure 2.8: Plot of Student Time Estimates and Level 1 Subjective Questions Average

The mean of the estimates derived without the Level 1 subjective questions results in a conservative time estimate that is 5 seconds or 15% greater than the mean of time estimates from the in-class activity. This indicates that had the students not made a subjective decision on Level 1, the difference in means of the results would still be within 15%. A variation of 15% is a reasonable range considering Boothroyd and Dewhurst state that a variation of up to 50% can be seen when conducting the assembly time estimate [4]. In this specific case the time estimates without Level 1 subjectivity resulted in a value that was greater than the student estimate. If the students had selected a handling or insertion code with a higher time estimate, then the average may have resulted in a time that was less than the student estimated time. The range of values should also be considered to ensure that a lower estimate does not influence the designer to overlook a part with assembly difficulties.

Furthermore the area underneath the average subjectivity curve (Figure 2.9), which is shared by the student estimate curve is approximately 63%. The range of times that were considered is from the student minimum time estimate of 23.64 s to the student maximum estimate of 44.89 s. This indicates that using the average value of the Level 1 subjective questions would result in an estimated assembly time estimate which falls within the normal distribution of the student estimates 63% of the time.

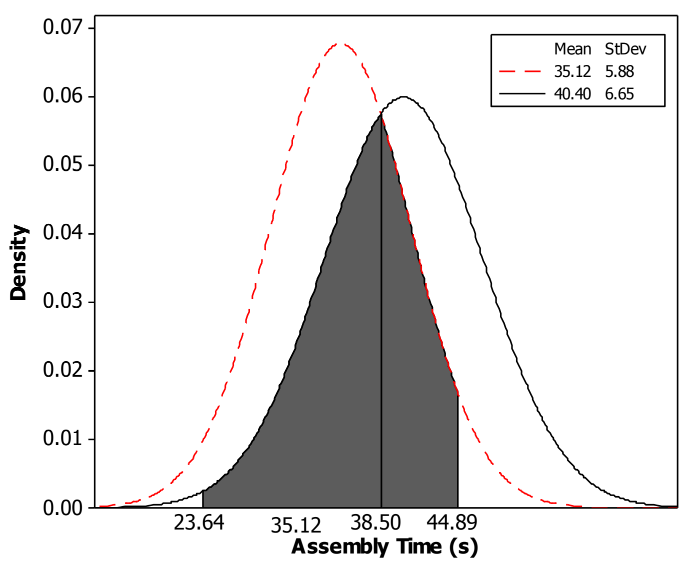


Figure 2.9: Area Overlap Under Data Curves

2.3.4 Conclusions and Future Work

The current assembly time estimation method requires subjective input from the individual conducting the analysis such as “is the part easy to grasp and manipulate”, “is the part easy to align and position during assembly”, “does the part present handling difficulties”, and “will the part nest or tangle”. Initial results from the in-class activity suggest that the subjective questions in the Boothroyd and Dewhurst manual assembly time estimate charts has an effect on the estimated assembly time of part. However, the results from the pilot study indicate that even if the user does not make the Level 1 subjective decision, an assembly time estimate within approximately 15% can be predicted relative to if the subjective decision had been made.

While the sample size used in the current pilot study is not large enough to generalize the conclusions, it does provide anecdotal evidence that there is an opportunity to reduce or eliminate the subjective questions in the Boothroyd and Dewhurst manual assembly time method. Reducing or eliminating can allow the user to estimate the assembly time with a certain confidence, such as providing a range of estimated assembly time as opposed to a single assembly time with a false sense of confidence. The assembly time estimate charts may be re-organized such that if the user is not confident in the answer of any of the questions, they may choose to not answer it. This lack of additional information will then result in a larger range of estimated assembly time with a certain confidence that the actual assembly time falls within this range. In order to accomplish this, further research is required to determine the specific effect of each subjective question on the overall assembly time estimate. This is out of scope for this dissertation, but addressing this subjectivity issue is addressed.

If an assembly time interval can be derived based on the questions that a user has answered (as discussed above), an opportunity exists to support assembly time estimation throughout the design process. For example, if a part is being studied during the conceptual phase for feasibility, an assembly time estimate within 50% may be sufficient, and if that is the case then less information may be needed about the part to provide the designer a rough estimate of the assembly time. The user may be able to estimate an assembly time of a product by providing the answer to only one question of the assembly chart, but this will decrease the confidence in the assembly time estimate. This will reduce the amount of time and information needed to implement the assembly time estimation method. Early product design stages dictate between 70-80% of the cost of product development and manufacturing, therefore an opportunity to estimate the assembly cost of a product at the conceptual stage, even with a large confidence interval may be beneficial in reducing manufacturing costs [4,49,50,64,65]. This is addressed in Chapter Four.

The results of the pilot study serve as the motivation for the overall objective of this research to automate the Boothroyd and Dewhurst assembly time estimate method as a tool that would interface with CAD software to retrieve required information. The tool should retrieve information from CAD such as dimensions, weight, material, and symmetry to provide an assembly time estimate (Chapter Three). This study demonstrates that the variation seen in the assembly time estimation of a simple product such as a pen may reach ranges of 30%, which conforms to the predicted 50% variation that Boothroyd and Dewhurst suggest. *Thus, the acceptable range of accuracy predicted assembly times for any developed tool is set to 50% of the actual assembly time.*

CHAPTER THREE
INTERFERENCE DETECTION METHOD – GRAPH GENERATION

The assembly mate method provided an automated method for creating the complexity graphs based on the mates used to create an assembly. The Interference Detection method is a tool for generating the complexity graphs using part interferences to create the complexity graphs.

The Interference Detection Method (IDM) utilizes the interference detection tool in SolidWorks to determine the connectivity between parts (see Figure 3.1). The interference detection tool detects overlapping part geometry between any two parts in the assembly. Furthermore, the interference detection tool has additional options to "treat coincidence as interference" and to "treat subassemblies as components". The "treat coincidence as interference" allows for situations when an interfering part has the same nominal size of a piece it is fitting into or when a face of a part is coincident with another. For example, in block and pin assembly the nominal size of the pin is the same as the size of the hole in the block. The interference detection tool detects this as interference when the option is enabled (see Figure 3.1).

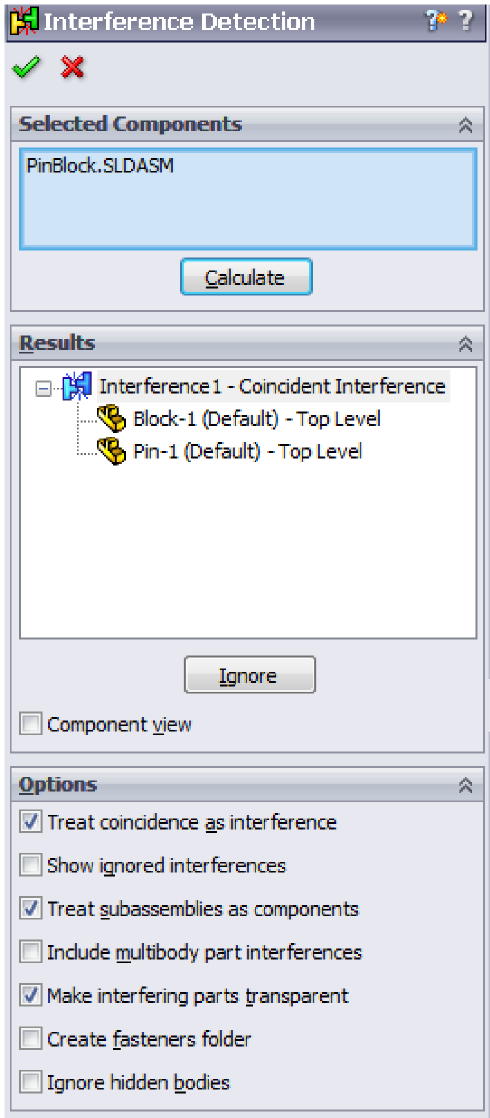


Figure 3.1: Interference Detection Tool

When a sub-assembly is placed into an assembly in SW, the entire subassembly is treated as one body or part. The "Treat subassemblies as components" option in the interference detection tool allows the tool to look at each part in the subassembly separately. The interference detection tool was run on the same block and pin assembly from earlier (see Figure 1.6). The results indicate that a connection was detected between the block and the pin (see Figure 3.1). Each portion of the part that is found to interfere is highlighted in red in (see Figure 3.2).

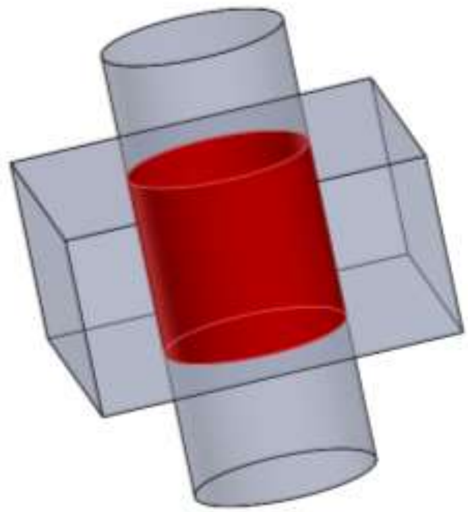


Figure 3.2: Block and Pin Interference Detection Tool Result

The process of finding interference is programmed in C++ using the SW API to find all interfering parts of the assembly and export a text file containing the part connection information. The interference detection tool may be run directly from the SW menu, by accessing the evaluate tab in an assembly file. The manual use of the interference detection tool results in a list of interferences in the SW GUI (see Figure 3.1).

3.1 Demonstration of IDM

To compare the two methods, a demonstration of the analysis on an ink pen is provided (see Figure 3.3).

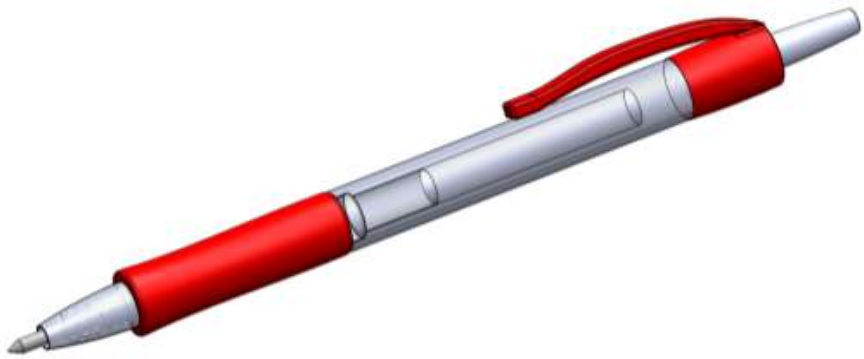


Figure 3.3: Ink Pen

The pen was chosen for demonstration due to a limited complexity and number of parts. This example does not demonstrate the full ability of the methods to create graphs for more complex products. The assembly model of the pen was opened in SW and the IDM method was executed. The output from the IDM is a text file indicating the connectivity between parts (see Table 3.1). Each row of the text file indicates a connection between the part located in the first column and the part located in the second column.

Table 3.1: Part Connections for IDM

Grip Body-1	Rubber Grip-1
Grip Body-1	Ink Body-1
Grip Body-1	Spring-1
Rubber Grip-1	Body-1
Press Button-1	Indexer-1
Press Button-1	Indexer-1
Press Button-1	Indexer-1
Press Button-1	Indexer-1
Press Button-1	Body-1
Spring-1	Ink Body-1

The bi-partite graph for the pen was also created for the IDM (see Figure 3.4). The connectivity between parts does not need to be represented in a graphical format;

however the complexity of a product is more apparent when compared in this format. The input into the algorithm to determine the complexity vector requires a table with a part in the first column and the part that is it connected to in the second column (see Table 3.1).

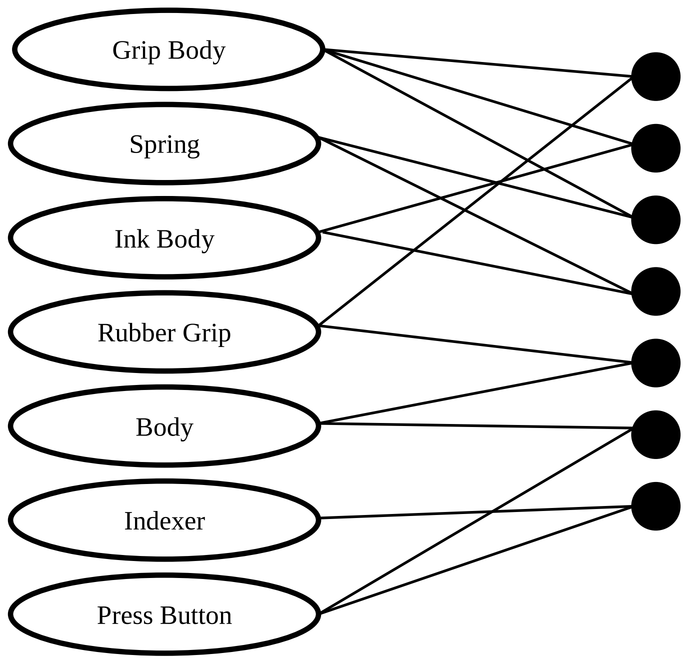
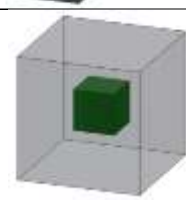
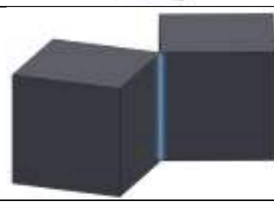


Figure 3.4: IDM Bi-Partite Graph of the Ink Pen

3.2 <u>Interference Detection Method Graph Generation - Test Cases</u>

To test the ability and limitations of the graph generation portion of the IDM, a number of test cases were developed. The test cases (see Table 3.2) are used to determine the topological limitations of the IDM in identifying two parts as being connected. The IDM detects overlapping or coincident interference between parts.

Table 3.2: IDM Graph Generation Test Cases

Assembly Description	Image	Interference Detected
Face to Face		✓
Partial Overlap		✓
Vertex Only		✗
One part completely within the other		✓
Edge Only		✓
Vertex on Edge		✗
Vertex on Face		✗

The IDM graph generation found a connection between parts for all of the test cases except for the cases with only a vertex connecting the two parts: Vertex only, Vertex on Edge, and Vertex on Face. While this is a limitation to the graph generation method, parts are generally not connected to another part by only a vertex. Ideally the graph generation method would still capture this relationship. Generally speaking, models are not assembled based on the vertex of a part. Connecting the parts based on the vertex does not completely restrict the movement of the part relative to another. Face to Face assembly was created by using a coincident mate between the two faces of the cube. For clarification, the IDM does not search the mate tree and does not require a list of SolidWorks mates to find the connectivity. For example, the Partial Overlap model was created by dragging the second cube in the assembly model space so that it overlapped with the first cube. There were no mates used to create the assembly model for the overlap, yet the IDM graph generation captures the connection between the parts.

To detect connectivity the parts are forced to be either interfering (overlapping) or share a coincident edge or face. One additional limitation that arises from this approach is often parts are designed with a tolerance in mind. For instance, a two inch diameter shaft being inserted into a two inch hole may have a tolerance modeled to allow the pin to slide in the hole without interference. If this tolerance is modeled in the solid model (pin nominal diameter is 2.000 inches, and the hole diameter is 2.002 inches) as opposed to only annotated on the engineering drawings, the IDM graph generation method will not identify the connection. This limitation will be reserved for future work, and possible approach updates and improvements will be suggested in (Chapter Eight).

3.3 Comparison of Graph Generation Methods

To compare the IDM and the AMM graph generation methods, a total of fourteen household products for which CAD models could be obtained or created were chosen for analysis. From the fourteen products used in the analysis, three products are withheld for testing. A summary of the products used for testing and training along with an image of each is presented in Table 3.3.

Table 3.3: CAD Models Used for Training and Testing

Product Name	**Training / Testing**	**CAD Model Image**	**Source**
Stapler	Testing		GICL Website [30]
Flashlight	Testing		SW 3D Content [30]
Blender	Testing		Reverse Engineered [30]
Ink Pen	Training		Reverse Engineered [30]

Table 3.3: CAD Models Used for Training and Testing

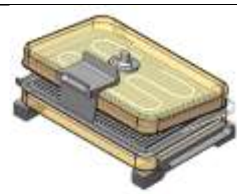
Product Name	Training / Testing	CAD Model Image	Source
Pencil Compass	Training		Reverse Engineered [30]
Electric Grill	Training		SW 3D Content [30]
Solar Yard Light	Training		Reverse Engineered [30]
Bench Vise	Training		Reverse Engineered [30]
Electric Drill	Training		Reverse Engineered [30]
Shift Frame	Training		OEM [30]

Table 3.3: CAD Models Used for Training and Testing

Product Name	**Training / Testing**	**CAD Model Image**	**Source**
Food Chopper	Training		Reverse Engineered [30]
Computer Mouse	Training		Reverse Engineered [30]
Piston	Training		Reverse Engineered [30]
3- Hole Punch	Training		Reverse Engineered [30]

3.3.1 Analysis Time

The time required to train, load, and run an ANN for the assembly time estimation using both methods is equal since both methods input the same amount and type of information. The required input for the ANN is simply the complexity vector. However, the time required to generate the connectivity graph based on a CAD model is less for the AMM compared to the IDM (see Table 3.4). The increase in analysis time for the IDM can be attributed to the algorithm complexity. The IDM must compare each part in the assembly to every other part to find interference, resulting in a computational complexity of $O(N^2)$. The AMM simply retrieves the created mates list to generate the part connectivity graph, resulting in a computational complexity of O(N).

Table 3.4: Graph Generation Time Comparison

	AMM			IDM		
	Graph Generation Time [s]	**# of Elements**	**# of Relations**	**Graph Generation Time [s]**	**# of Elements**	**# of Relations**
Flashlight	5	18	36	30	16	55
Stapler	1	14	27	43	14	20
Blender	1	48	105	97	43	129

The time to generate the graph for each of thirteen consumer products (see Table 3.3) was recorded to compare the theoretical complexities of the algorithms to the actual implementation. The graph generation time for the AMM and the IDM are plotted with respect to the number of elements and the number of relations (see Figure 3.5 and Figure 3.6). One may note that the number of elements and relations identified by each method are not identical and is not equal to the number of parts, therefore each graph generation time is plotted with respect to the number of elements and relations identified by the respective method.

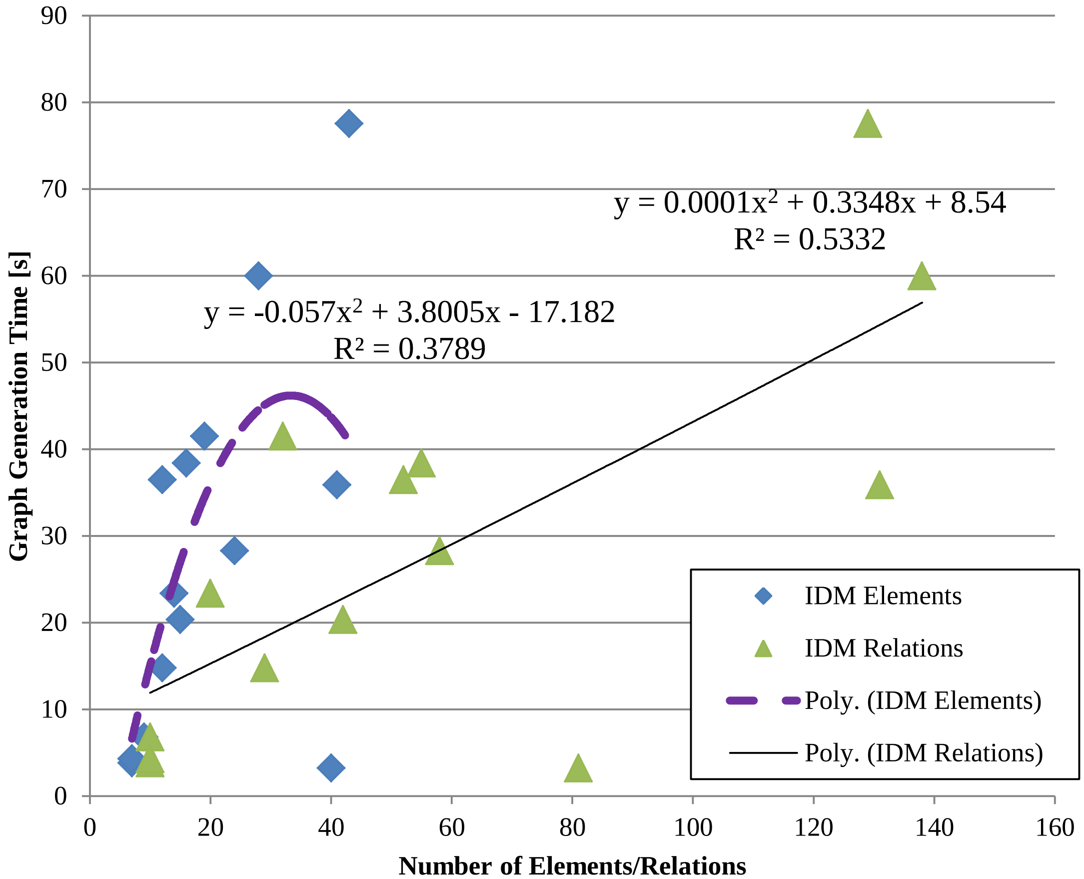


Figure 3.5: Graph Generation Times for IDM

Theoretically the IDM algorithm is polynomial, however the applied results of the graph generation times initially indicate that the polynomial fit based on number of elements or relations alone is not sufficient. A number of factors could be considered to be the cause of the discrepancy between the theoretical and applied graph generation times. First of all, the sample size is not sufficiently large to draw complete conclusions. A set of products with a larger range in number of parts and relations would need to be tested to further support the actual relationship between graph generation time and number of elements or relations. Another possible contribution to the discrepancy is the complexity of the part topology. To find the interference of a part with multiple edges and faces requires greater computation than a part with a simple geometry. This however will also need to be further tested. To do this, a study can be conducted in which an assembly composed of parts with simple geometries is compared to a similar assembly in which the geometry of the parts is changed, but the interfering components should remain the same. This is not the focus of this research and is reserved for future work.

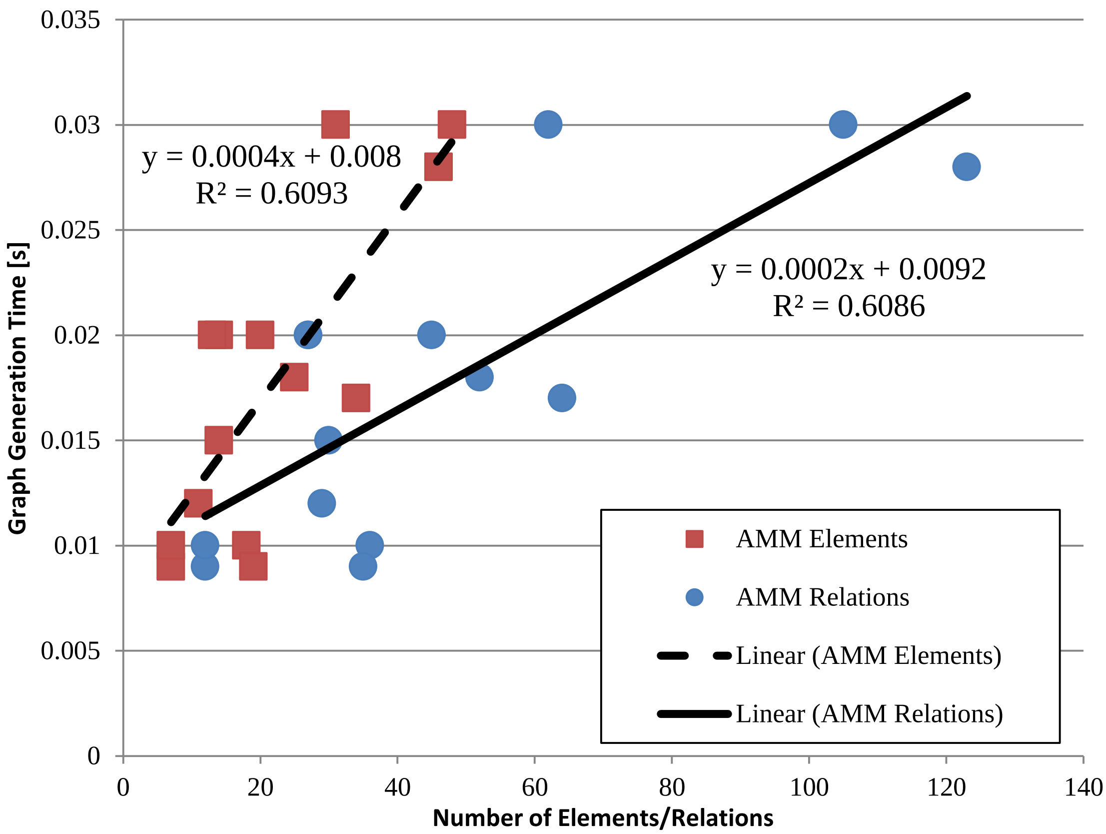


Figure 3.6: Graph Generation Times for AMM

The AMM reveals a relatively linear trend with the increase in elements or relations having a minimal effect on the graph generation time (see Figure 3.6). The AMM is traversing a list that has been created by the SW program during the assembly modeling, and then writing this information to a text file. For this reason the applied results generally follow the trend expected from the theoretical evaluation and are independent of factors such as part geometry and topology complexity.

Cooperation with a local original equipment manufacturer (OEM) provided two additional models with a higher element and relation count. While the sample size is not sufficient to make any general claims, the data still provides insight and demonstration that the IDM is capable of handling larger assemblies. The names of the products and the name of the local OEM are not disclosed due to proprietary reasons.

Table 3.5: Graph Generation Time for Large Assemblies

	IDM		
	Graph Generation Time [s]	**# of Elements**	**# of Relations**
Assembly 1	6557	159	872
Assembly 2	5012	75	367

When the results of the graph generation time for the IDM are added to the chart along with the previous products, the general polynomial trend is still evident for the number of elements; however the number of relations is better fit by an exponential model (see Figure 3.7). This case demonstrates that the IDM is able to handle larger scale assembly models, although the graph generation time appears to increase exponentially for the number of relations in the assembly.

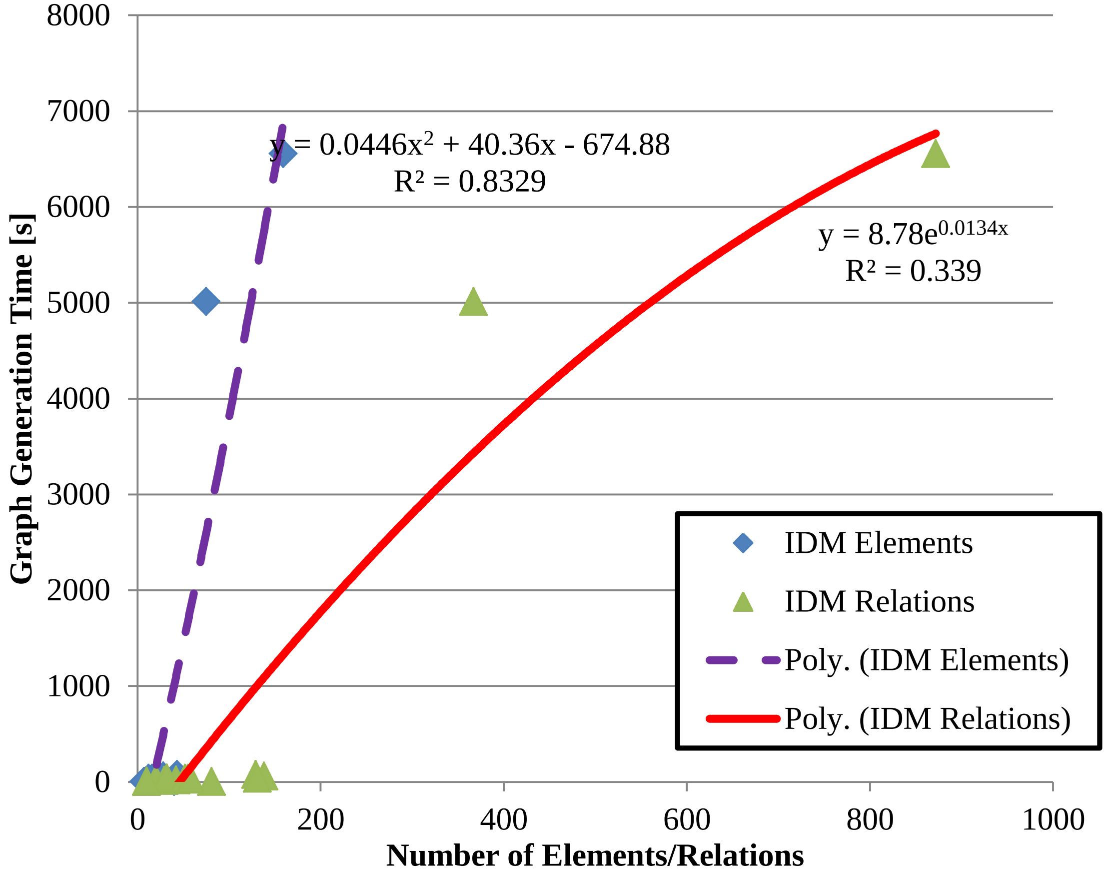


Figure 3.7: IDM Graph Generation Time for Large Assemblies

While the results generally follow the expected trends, the sample size and variation in number of elements and relations is still limited and requires additional testing to support these claims.

3.3.2 Supported CAD File Types

One advantage of the IDM over the AMM is the ability to handle additional file types other than SW Assembly Files. The AMM is dependent on having a SW assembly file to retrieve assembly mates from. Using the import features built in and provided by SW, multiple file types may be imported and converted to SW assembly files. However, when these files (see Table 3.6) are saved into a generic CAD format for exchange between systems, the assembly mates are not preserved. The IDM is able to create the connectivity graph of many different native file formats once imported using SW and has been tested on the following: SW assembly file (*.sldasm), IGES (*.iges), parasolid(*.x_t), and STEP (*.step;*.stp) (summarized in Table 3.6). The STL file type is not currently supported by the IDM.

Table 3.6: IDM Supported File Types

File Type	File Type Extension	Supported
SolidWorks Assembly	*.asm;*.sldasm	✓
IGES	*.iges; *.igs	✓
Parasolid	*.x_t;*.x_b;*.xmt_txt;*.xmt_bin	✓
STEP	*.step;*.stp	✓
STL	*.stl,	✗

While the IDM can support multiple file types, SW is still required as the add-in to run the interference detection tool is built using the SW API and as the base software for importing the various CAD transfer formats. However, the benefit is that files can be saved into a standard CAD file format from other CAD systems and imported into SW to run the IDM. Moreover, the algorithm for interference detection is straightforward and can be implemented into similar commercial systems. In fact, a major automotive OEM currently uses a similar algorithm, developed independently, to run clash and interference detection on vehicle assembly models. The use of this algorithm to generate connectivity graphs is currently the focus of on-going work at the OEM in cooperation with researchers at the CEDAR Group.

3.3.3 Designer Dependency

When creating a solid model, there are numerous ways a designer could model the product. The actual geometry and technique used to create a part may slightly vary by designer, but this is out of scope of this research. On the other hand, given a set of parts, different designers will mate them in different way to form the assembly. For instance, based on the ink pen example from earlier, an alternate designer may mate multiple parts to a reference plane. Furthermore, a designer may choose to limit the motion of all of the parts in the assembly to create a fully defined assembly in which all parts have zero degrees of freedom. [58] This situation would result in an entirely different connectivity graph based on the AMM. Since the AMM uses the mates from the assembly model to create the connection graph, all reference items which are used to mate the assembly are also included as entities (see Table 3.7).

Table 3.7: Part Connections for AMM

Grip Body-1	Rubber Grip-1
Grip Body-1	Ink Body-1
spring-1	Rubber Grip-1
Ink Body-1	Indexer-1
Press Button-1	Indexer-1
Grip Body-1	Body-1
Grip Body-1	Rubber Grip-1
spring-1	Grip Body-1
Ink Body-1	Grip Body-1
Press Button-1	Body-1
Press Button-1	Indexer-1
Rubber Grip-1	Body-1
Rubber Grip-1	Front Plane
spring-1	Front Plane
Ink Body-1	Front Plane
Press Button-1	Front Plane
Indexer-1	Front Plane
Body-1	Front Plane

These added relations increase the size of the connectivity graph and therefore also generate a different bi-partite graph and calculated complexity vector (see Figure 3.8)

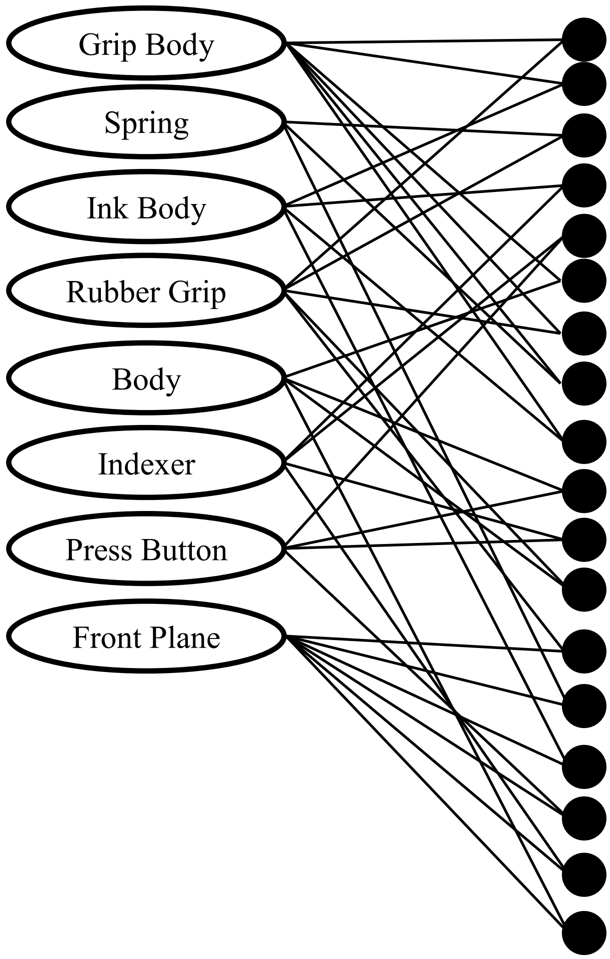


Figure 3.8: AMM Bi-Partite Graph of Fully Defined Ink Pen

Since the IDM is based on location of the parts in the modeling space, the connectivity graph is not dependent on the modeling style of the designer, but strictly on the location of the parts in the assembly space.

CHAPTER FOUR
APPLICATION OF IDM DURING THE CONCEPTUAL DESIGN STAGE

One shortcoming identified in earlier portion of this research is the limitation of application of design for assembly time estimation methods to the detailed design stage or for use as a redesign tool. The majority of the cost of a product is determined during the conceptual design stage, and therefore a tool to support design for assembly during conceptual design is desired. The connectivity complexity method does not require detailed information regarding the part (such as geometry), but strictly on the physical connection between the parts in a product. The IDM can be used to generate connectivity graphs of low fidelity CAD models as inputs into the connectivity complexity method to predict assembly times of products early in the design stage.

Previous work has focused on estimating assembly times from detailed component and assembly models [23,30,66]. This work evaluates the potential of using components represented at lower levels of detail, such as conceptual models or low-fidelity models. While the exact dimensions and features of the components are not known, the general system architecture and layout is captured early in design [50]. The form of the individual components are developed throughout the design process to create a completed CAD model with working drawings in the detailed design stage [50]. For clarity, low-fidelity models are those that are found in conceptual design and high-fidelity models are found in detailed design phases.

This portion of the research explores the use a modified complexity connectivity method to estimate the assembly time of models in the conceptual design phase. The estimated assembly time of the conceptual models is compared to the estimated assembly time of the complete models using the same modified complexity connectivity method.

4.1 Set of Models

The experiment presented in this chapter uses a total of thirteen products (Table 4.1) to compare the estimated assembly time of high-fidelity models and low-fidelity models. The models were used in previous work and were created by multiple designers by physically reverse engineering existing products or downloading models from the public domain [30]. The first three models are withheld for testing purposes.

Table 4.1. Products Used in Training and Testing

Common Name	Training/Testing	CAD Model Image
Stapler	Testing	
Flash Light	Testing	
Ink Pen	Testing	
Pencil Compass	Training	
Indoor Electric Grill	Training	
Solar Yard Light	Training	See Table 3.3
Table Vise	Training	
Drill	Training	See Table 3.3
Shift Frame	Training	See Table 3.3
Vegetable Chopper	Training	See Table 3.3
Computer Mouse	Training	See Table 3.3
Piston Assembly	Training	See Table 3.3
3 Hole Punch	Training	See Table 3.3

4.2 Reducing Model Fidelity

Low-fidelity CAD models are difficult to define and are often not distinctly saved by the designer before they are evolved to more detailed higher fidelity models. For this work, the high-fidelity models were reduced in fidelity to represent low-fidelity models in the conceptual design phase.

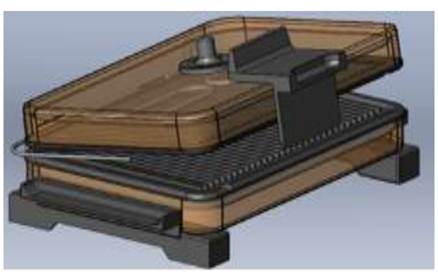
To do this, each part included in an assembly model was reduced to its lowest level feature. In SolidWorks the feature tree stores the features used to create a part and the order in which those features were created. To decrease bias in the reduction of fidelity of the parts, the feature tree was reduced to the top level feature for each part. It should be noted that if multiple designers create the same part, a different conceptual model may result. This uncertainty is not the focus of this research and is reserved for future work.

As an example, the first feature used to create a bolt may be an extruded shaft (Boss-Extrude1). Next, a swept extrusion (Sweep1) is used to create the threads around the shaft of the bolt. An additional extrude (Boss-Extrude2) is used to create the bolt head and then an extruded cut (Cut-Extrude1) is used to cut the hex in the top of the bolt head. Starting from the bottom of the feature design tree, the Cut-Extrude1 is deleted, followed by Boss-Extrude2 and Sweep1 leaving only the initial extrude as an example of a conceptual model for a bolt (see Table 4.2).

Table 4.2. Reduction of Fidelity of a Bolt Model to Create a Low Fidelity Model

High Fidelity (final part)	Intermediate 2	Intermediate 1	Low Fidelity (raw part)
			
Cut-Extrude1	Boss-Extrude2	Sweep1	Boss-Extrude1

This removes detail from the parts in the CAD model, leaving a low-fidelity model of the product simulating a model created in the conceptual phase of the design process. The indoor electric grill (Figure 4.1) is similarly reduced from a detailed model to an assembly of the low-fidelity part models. Mating relationships may be lost in this transformation, precluding the use of previous graph generation tools [30]. Therefore, a mate-independent method for generating the connectivity graphs is used based on interference checks.



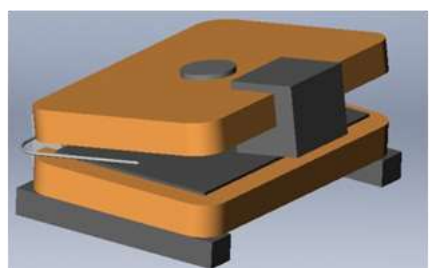


Figure 4.1. Electric Grill from High Fidelity Model to Low Fidelity Model

4.3 Artificial Neural Network Generation

The artificial neural network (ANN) used for this research is a supervised back propagation network [30,31,67,68]. The ANN is trained by providing a set of input vectors and a set of target values. The ANN then creates a relationship model between the input values and the target value. In this case, the complexity vector of 29 metrics is the input vector and the assembly time of the product will be used as the output. Once an ANN is trained, a new complexity metric is input and the ANN provides an assembly time.

4.4 Experimental Sets

Two separate neural networks are defined, trained, and compared. The first ANN uses the complexity vector of the high-fidelity models as input and assembly times as the targets. The second ANN uses the complexity vectors of the low-fidelity models as the training inputs and the same assembly times as target times. The low fidelity complexity vector and high fidelity complexity vector for each product differ, since the physical connection between elements is altered. The low fidelity and high fidelity complexity vectors of a pen are included for reference (see Table 4.3).

Table 4.3 High and Low Fidelity Complexity Vector for Pen

					High Fidelity	Low Fidelity
Complexity Metrics	Size	Dim		elements	7.00	7.00
				relations	10.00	12.00
		Conn.		DOF	10.00	12.00
				conn	20.00	24.00
	Interconnection	Shortest Path		sum	102.00	64.00
				max	5.00	3.00
				mean	2.43	1.52
				density	0.24	0.13
		Flow Rate		sum	54.00	110.00
				max	4.00	4.00
				mean	1.10	2.24
				density	0.11	0.19
	Centrality	Betweenness		sum	60.00	22.00
				max	18.00	16.00
				mean	8.57	3.14
				density	0.86	0.26
		Clustering Coefficient		sum	2.33	5.90
				max	1.00	1.00
				mean	0.33	0.84
				density	0.03	0.07
	Decomposition	Ameri Summers			20.00	35.00
		Core Numbers	In	sum	10.00	19.00
				max	2.00	3.00
				mean	1.43	2.71
				density	0.14	0.23
			Out	sum	10.00	19.00
				max	2.00	3.00
				mean	1.43	2.71
				density	0.14	0.23

This approach is used to test the ability to train a neural network to find a relationship between low-fidelity complexity vectors and product assembly times. Each ANN is used to predict the assembly time of a test data set (three products) using the high-fidelity and low-fidelity models. The experimental sets are summarized in Table 4.4.

Table 4.4. Experiment Design Sets

Set Number	ANN Trained on:	Input Vector Set Type:
1	High Fidelity Models Vectors	High Fidelity Model Test Vector
2	High Fidelity Models Vectors	Low Fidelity Model Test Vectors
3	Low Fidelity Model Vectors	High Fidelity Model Test Vector
4	Low Fidelity Model Vectors	Low Fidelity Model Test Vectors

4.5 Conceptual Model Time Estimate Results

After the two ANNs are trained, the input vectors are passed back in to the neural network to gain a qualitative assessment of ANN fit to the training set. The percent error is calculated as the normalized difference from the target time (see Eqn. (1)). A positive percent error indicates that the predicted time was greater than the target time, and a negative percent error indicates that the predicted time is less than the target time.

$$\% \text{ Error} = \frac{P - T}{T} \; x \; 100 \tag{1}$$

The ANNs are able to estimate the training set assembly times within 70% of the target time, and does not visually appear to be over fit to the training set data (see Figure 4.3). Overtraining results in a model that represents the current data set, but limits the ability of the ANN to extrapolate to new data sets [67,69,70]. An over fit training set would result in a predicted time that overlays the training data very closely. In an over fit case each point of the predicted time would fall on the training time.

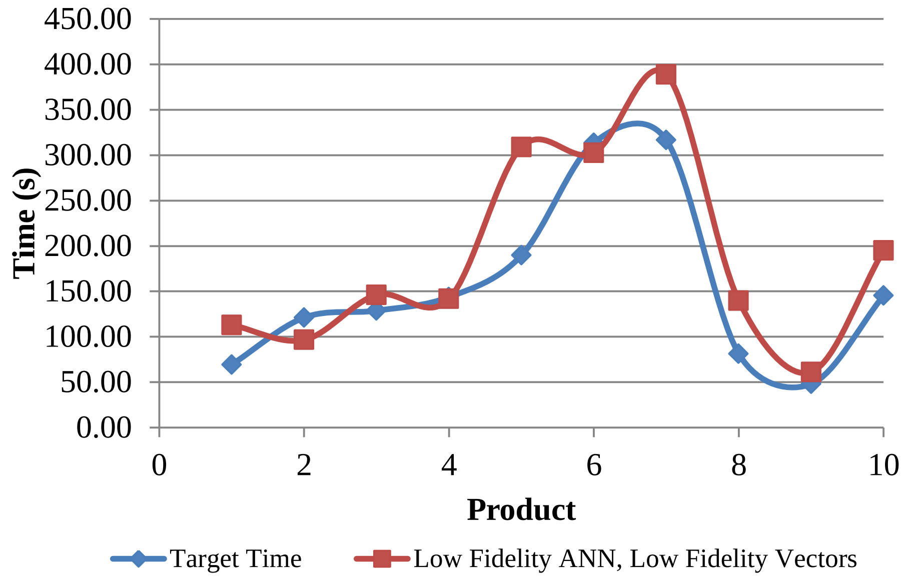


Figure 4.2: Training Times and Predicted Times

Previous research offers numerical techniques to detect and prevent ANN over fit and improve performance of ANN by varying ANN parameters [71]. As the focus of this research is to demonstrate the potential to use ANN to predict assembly times of low-fidelity models, the improvement in design of the ANN itself is reserved for future work.

To test the performance of the two ANNs in predicting the assembly times, complexity vectors of three products, the stapler, flash light, and ink pen, not used in the training are used for testing. For each of the test products the high fidelity and low fidelity graph complexity vectors were calculated and used as the input to both ANNs trained, both high fidelity and low fidelity.

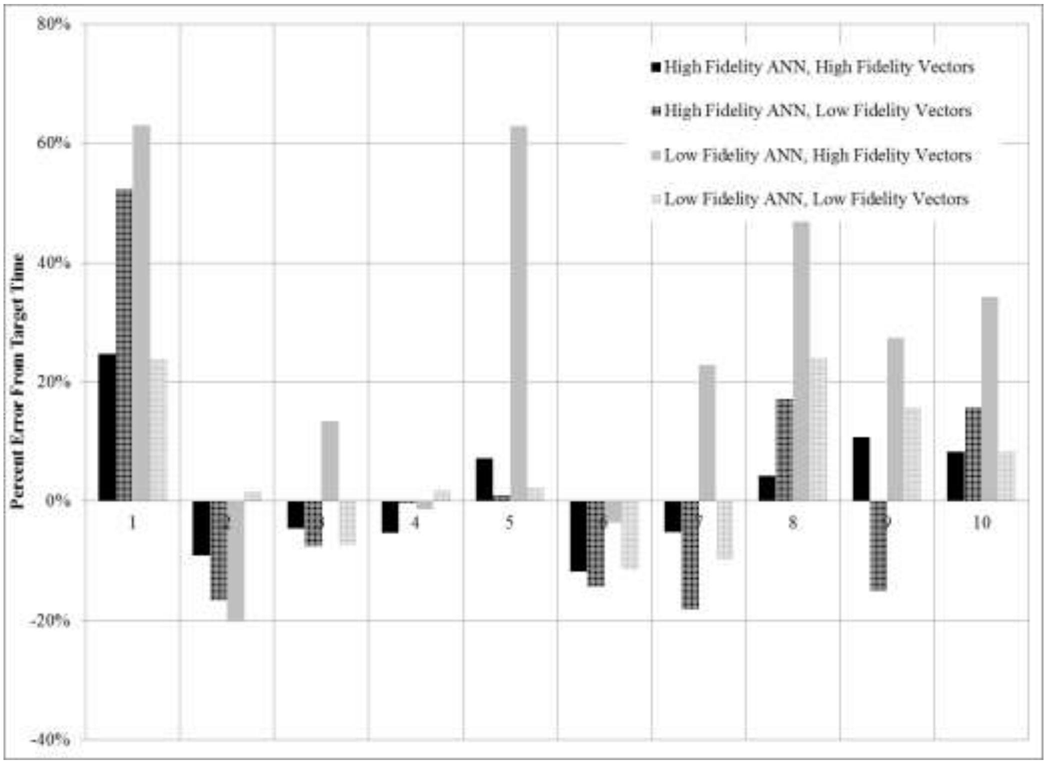


Figure 4.3. Training Set Percent Error from Target Time

The target time, the predicted time, and the percent error for each of the three test cases are presented in Table 4.5. Each ANN predicted an assembly time greater than the target time for the test cases except for the high-fidelity ANN for the stapler. The test products varied in target assembly times from 34 seconds to 123 seconds. Additional test cases with a larger range of assembly times are needed to determine if the ANN time estimate accuracy is dependent on the assembly time or the complexity of the product being studied.

Table 4.5. Test Products Results Summary

Fidelity Levels		Predicted Time [s] (Percent Error)		
ANN	**Test Assembly**	**Stapler**	**Flash Light**	**Ink Pen**
High	**High**	115.84 (-6%)	107.65 (43%)	54.78 (59%)
High	**Low**	119.43 (-3%)	91.79 (22%)	46.41 (35%)
Low	**High**	157.19 (27%)	109.89 (46%)	72.36 (110%)
Low	**Low**	198.30 (61%)	95.19 (26%)	51.65 (50%)
	Target Time [s]	123.51	75.40	34.40

The percent error from the target time was calculated for each of the outcomes (see Figure 4.4, Figure 4.5, and Figure 4.6).

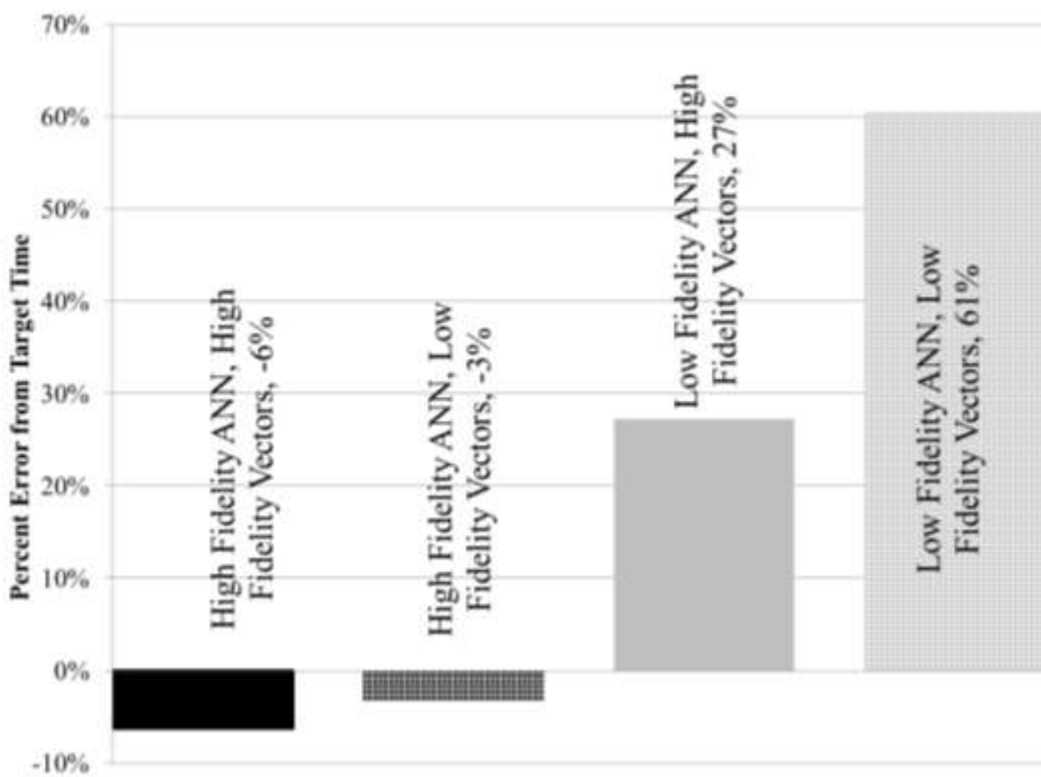


Figure 4.4. Test Case Results for Stapler

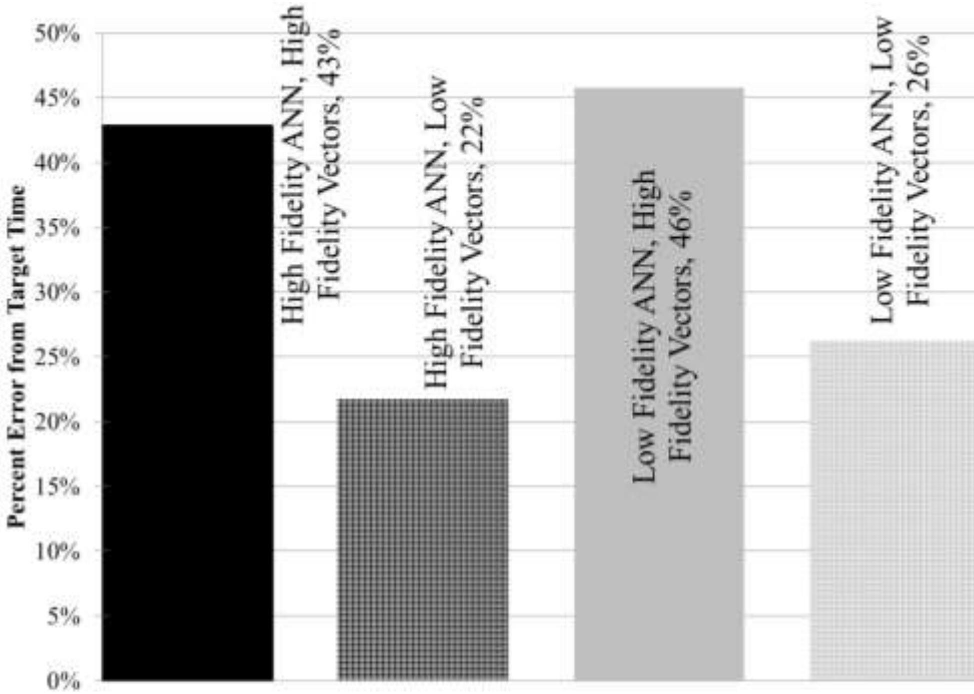


Figure 4.5. Test Case Results for Flash Light

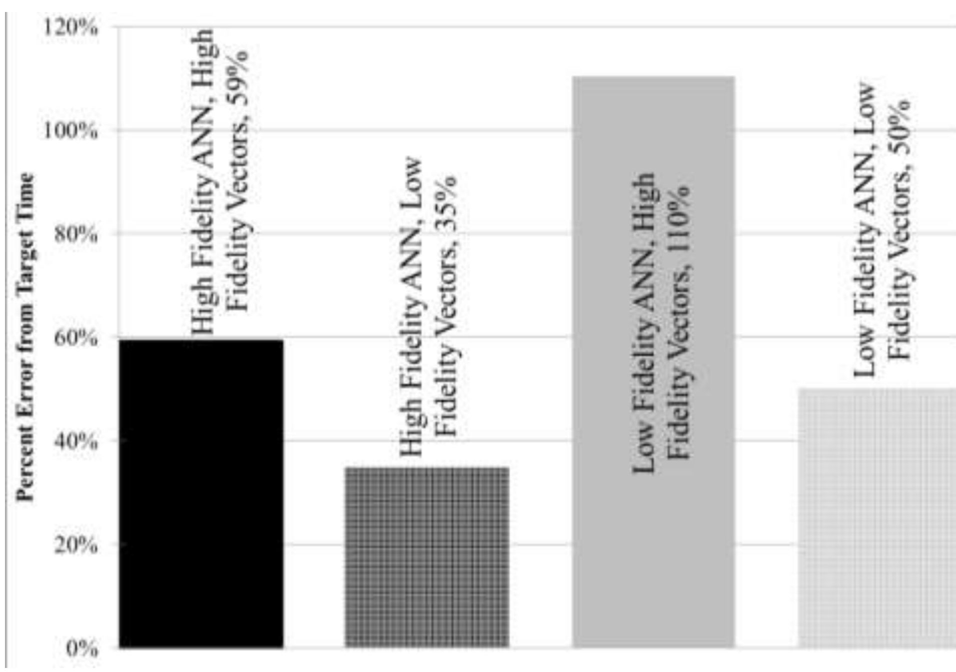


Figure 4.6. Test Case Results for Ink Pen

Visual inspection of the results suggests that both of the ANNs (high fidelity and low fidelity trained) were able to predict the assembly time of the test cases to within 120% independent of the type of input vector used. However, the low fidelity ANN was the generally the worst at predicting assembly time when presented with a high fidelity input vector. The best combination of ANN and input vectors, based on the lowest percent error for all three test cases is the high fidelity ANN being provide low fidelity input vectors. The focus of this research is if an ANN can predict the assembly time of a low fidelity model. Both the high fidelity ANN and the low fidelity ANN were able to predict the assembly time of the conceptual test models to within 120% of the target time.

To statistically investigate the results of the ANNs and the input vector fidelity, an analysis of variance (ANOVA) is used. The fidelity level of the ANN (factor 1) and the input vector (factor 2) has either a low fidelity or a high fidelity. Each experiment had three replications since it was tested using three products, stapler, flash light, and pen (see Table 4.6).

Table 4.6: Experiment Design for Test Cases

Experiment #	ANN Fidelity (Factor 1)	Input Vector Fidelity (Factor 2)	Replications
1	Low	Low	3
2	Low	High	3
3	High	Low	3
4	High	High	3

The effect of ANN fidelity was not significant ($p = 0.147$) and the effect of the model fidelity was also not significant ($p = 0.4297$) at an alpha value of 0.05. Previous research has suggested using a more lenient alpha value when studying human subjects or experiments with low sample sizes [72,73]. At an alpha value of 0.15 (85% confidence), there is some evidence to suggest that the fidelity of the ANN does reduce the % error in predicting the assembly time of a product.

To further explore the effect of the fidelity of the ANN and the fidelity of the input vectors on the assembly time estimation, the entire set of thirteen products (training and testing) are considered (see Table 4.7). The focus of this portion of the research is to determine if the assembly time of products early in the design stage (low model fidelity) can be estimated using the IDM, and the type of ANN training (low fidelity or high fidelity) should be used. Theoretically, the training products should return an assembly time estimate with lower percent error since these products were used to train the neural network. Therefore, the values for the percent error in this portion should not be used to generalize expected error for applying the method to future products, but only for comparison purposes.

Table 4.7: Experiment Design for Entire Sample

Experiment #	ANN Fidelity (Factor 1)	Input Vector Fidelity (Factor 2)	Replications
1	Low	Low	13
2	Low	High	13
3	High	Low	13
4	High	High	13

The results indicate that at an alpha value of 0.05, the fidelity of the ANN is a significant factor (p = 0.018). The fidelity of the input vector is not significant (p = 0.103). The mean percent error of the low fidelity ANN and the high fidelity ANN are 7.115 and 24.692 respectively. The mean percent error seen by using the ANN trained with the high fidelity vectors had less mean percent error than the ANN trained with the low fidelity vectors. The numerical value or difference between the means is not meaningful for generalization because the training products are used in the analysis to increase the replication size. The training sets and the test cases were limited in number and could potentially influence the results. The results of this study serve as demonstration that there is potential to use an ANN to estimate the assembly time of models early in the design process.

4.6 Conclusions and Future Work

The ability of a neural network to create a relationship between input vectors and output vectors depends on the training set provided. The larger the training set, to a degree to avoid over fitting, the better the neural network is at predicting the output. While the input vectors used to train the neural network in this research are limited to ten training products, future work includes increasing the training set to determine if the assembly time estimation can be further improved. The number of test products will also be increased to ensure the trends in this limited population are valid.

The findings of this study suggest that the high fidelity assembly model based neural networks provide good prediction tools for estimating assembly time for both high fidelity and low fidelity conceptual models. This tool shows promise for providing engineers in conceptual stages of product development with useful information about production costs via assembly time estimation early in the design process. The accuracy of these predicted times are sufficient to provide justification for alternative engineering selection decisions at early stages. While this research is not specifically being conducted for conceptual design, the application of a design for assembly method in early design stages is desirable. This study provides suggests that the application of the IDM method for early design stages is viable. More significantly, this approach is demonstrated to operate on assembly models in earlier stages of development than any other reviewed DFA methods and approaches.

CHAPTER FIVE
TESTING OF INTERFERENCE DETECTION METHOD

To test the performance of the IDM, the method is used for internal testing and external testing. For internal testing, a set of fourteen products are reverse engineered and the assembly time of each is calculated using the Boothroyd and Dewhurst assembly time estimation manual chart method. The assembly times are used to train an ANN to estimate the assembly time of the product in comparison to times found the AMM (see Chapter 1.2.2.2). For external testing, a local OEM has provided assembly models and actual in plant assembly times of fourteen products. These models and assembly times are used to train another ANN.

5.1 Internal Testing – CEDAR Products

The connectivity graph for the eleven training products was obtained using both the AMM and the IDM methods. The complexity metrics for each respective method was obtained and was used as the input for training of an artificial neural network. The target time for each of the products was calculated using the manual Boothroyd and Dewhurst assembly time estimation charts [4].

The connectivity graphs and complexity vectors for the test products were then generated using each of the graph generation methods. The previously trained neural networks were then used as a prediction tool to estimate the assembly time of the test products. Each neural network is composed of 189 architectures and each architecture has 100 repetitions resulting in 18,900 predicted assembly time data points for each product. The average time of all of the results of a neural network is the average predicted assembly time (see Table 5.1). The number of architectures as well as repetitions for each architecture may be reduced to decrease computational effort.

Table 5.1: Predicted Assembly Times of Test Products

	Target Time	AMM Average Predicted Time	IDM Average Predicted Time
Stapler	123.51	115.84	89.98
Flashlight	75.40	107.65	65.96
Blender	263.21	290.40	352.09

To compare the predictive ability of each of the graph generation methods, the mean percentage error (MPE) was calculated for each neural network. The MPE is calculated as the following:

$$\mathrm{MPE} = \frac{1}{n}\sum_{1}^{n}\frac{P-T}{T} \qquad (1)$$

Where:

n = Number of Observations

T: Target Time

P: Predicted Time

The MPE of each of the test cases is calculated, and all of the MPE values are less than 45% (Figure 5.1). Graphically, neither method has a clear advantage based on MPE. The IDM has a lower MPE for the stapler and blender, but the AMM has a lower MPE for the maglight.

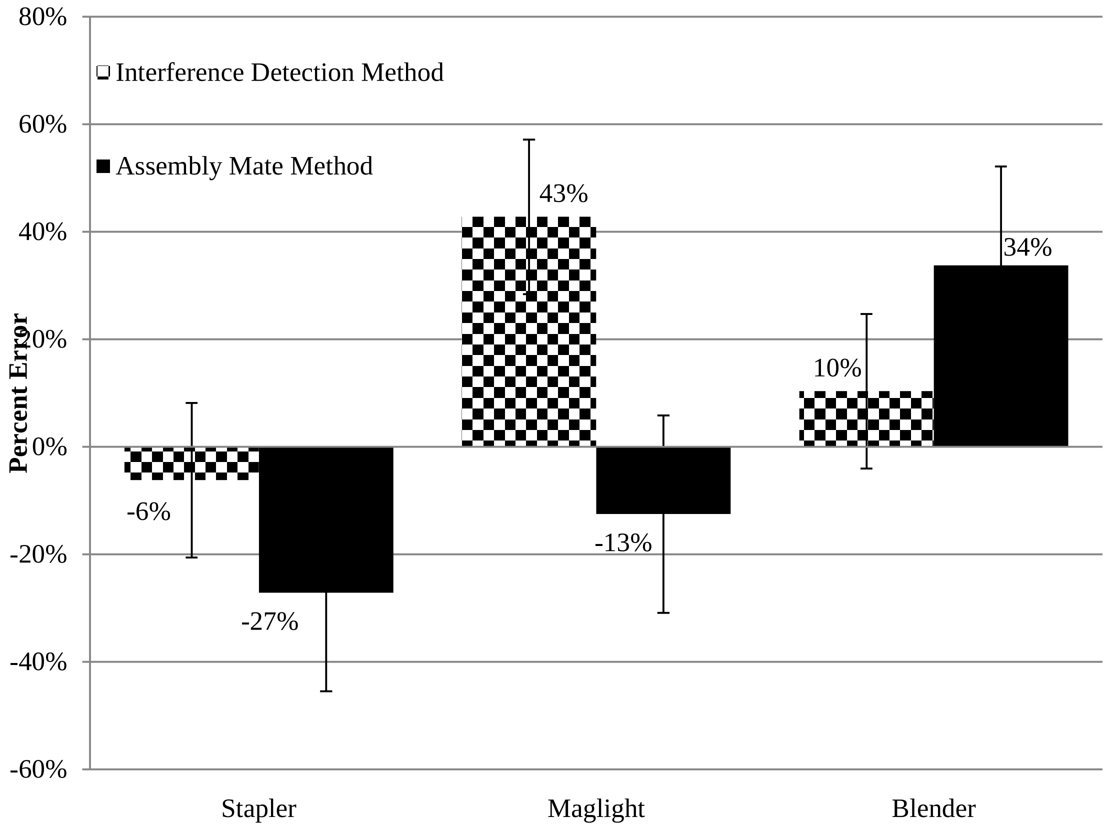


Figure 5.1: Mean Percent Error of Test Products

To compare the mean percent error values a 2 sample t-test was conducted. Based on the central limit theorem, the sample size is large enough to assume a normal distribution and therefore a two sample t-test with unknown variances is appropriate.

The hypothesis test was used to test if the mean average error of the IDM was statically different than the mean of the AMM. The confidence interval used for this test was 95%.

$$H_0 : \mu_0 = \mu_1$$

$$H_1 : \mu_0 \neq \mu_1$$

The results indicate a p-value less than 0.05 providing evidence to reject the null hypothesis. The t-test suggests that the mean percent error values of assembly time are not equal. While there is statistically significant evidence that the means are not equal, the practical difference in the means are not different. Graphically, the mean percentage error of the IDM and the AMM are similar (see Figure 5.2). The graphical depiction, however, does suggest that, while the means are similar, the variance observed with the AMM method is greater than that observed with the IDM. The graphical evidence supports that both methods are relatively accurate in estimating assembly time, but the IDM method produces less variance. The results of the three test cases suggest that the AMM is more accurate in estimating assembly times however has a greater variance, indicating the time estimates are centered about the mean.

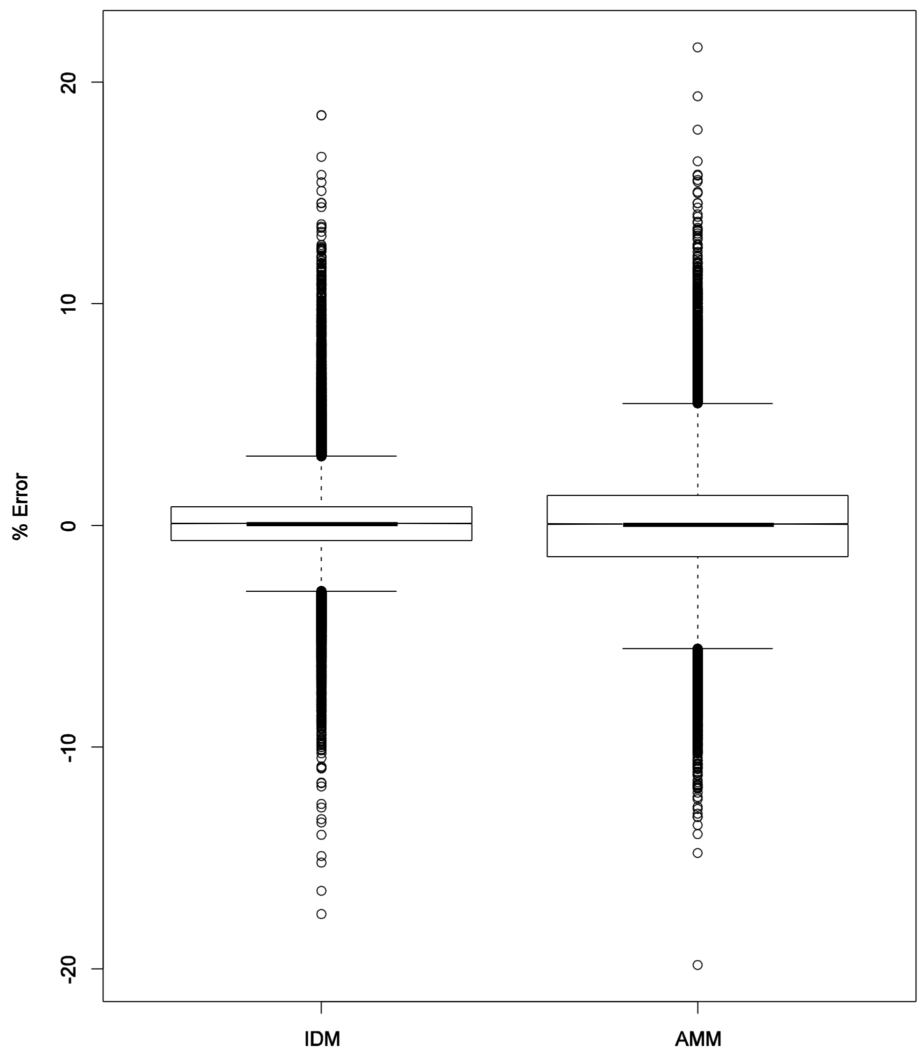


Figure 5.2: Comparison of IDM and AMM Assembly Time Estimates

5.2 External Testing – Original Equipment Manufacturer Products

The IDM has been shown to be able to estimate the assembly time of products that were reverse engineered, and the target assembly time was calculated using the manual Boothroyd and Dewhurst assembly time estimation method. While previous literature has shown that the Boothroyd and Dewhurst assembly time estimation method is well accepted in academia and industry, this research is further validated by applying the IDM to products currently in production with known assembly times.

A local power tool manufacturing company provided CAD assembly models of fourteen products along with the actual in plant assembly times of each product. The phrase “actual assembly time” is used to describe the measured assembly time required to assemble a product. The times for each product are acquired directly from the manufacturing plant where the product is being assembled.

This experiment was conducted to determine if a neural network trained with products from the same product family and manufactured by the same company would improve the methods ability to estimate the assembly time. Previous work on manually constructed connectivity graphs for automotive sub-systems demonstrated that the method performed better to predict assembly times for products drawn from the same portion of the OEM’s assembly line [29]. This informs the following hypothesis for this experiment.

Hypothesis	Training an artificial neural network using products from within the same product family as the products for which be estimating the assembly time of will improve the overall accuracy of the time estimate.

5.2.1 Artificial Neural Network Training

To conduct this experiment, two previously trained ANNs are used to estimate the assembly time of the products provided by a local power tool OEM. This OEM is a major competitor in the design and manufacturing of power tools, outdoor equipment, and floor care. Tools from the power tools division will be used for external testing in this research.

The first ANN was trained using the CEDAR products (see Table 3.3). The CEDAR products include consumer products mostly composed of electromechanical devices (see Table 5.2), for additional information and pictures see Chapter 3.3).

Table 5.2: CEDAR Training Products

Ink Pen
Pencil Compass
Indoor Electric Grill Model
Solar Yard Light
Pony Vise
Electric Drill
Shift Frame
One Touch Chopper
Computer Mouse
Piston Assembly
Three Hole Punch

The second ANN is the OEM ANN which was trained using product models and actual assembly times provided by OEM. The OEM ANN products are tools that are from the handheld power tools product family (Table 5.3). To train the ANN, eleven of the fourteen products provided by the OEM were used as the training set (See Table 5.3). The remaining set of products will be used to test the ability of the ANN to estimate the assembly times once training is completed. A supervised back propagation network is used to find a relationship between the complexity metrics of the training set and the respective assembly time.

Table 5.3: Training and Testing Products for OEM ANN

Product Name	Training/Testing	Image
Circular Saw	Training	
Laminate Trimmer	Training	
Reciprocating saw	Training	
Compact reciprocating saw	Training	

Compact Jigsaw	Training	
Drill	Training	
Impact	Training	
Angle Drill	Training	

Ratchet head	Training	
Multitool	Training	
Hammer Head	Training	
Recip Saw Head	Testing	

Jig Saw Head	Testing	
Hammer Drill	Testing	

5.2.2 Assembly Time Estimation

The trained ANN was used to estimate the assembly time of the three products that were withheld from the training set. The estimated time is the average time of the result of 18,900 time estimates resulting from the ANN design of 189 architectures with 100 repetitions.

Table 5.4: OEM ANN Assembly Time Estimation Results

	Recip Head	**Jigsaw Head**	**Hammer Drill**
Target Time [s]	~1200	~1100	~1400
OEM ANN Average Estimated Time [s]	825	778	410
Percent Error	-35%	-30%	-70%

The results of time estimate show that the mean percent error (MPE) of the reciprocating head, the jigsaw head, and the hammer drill was -35%, -30%, and -70% respectively. The negative MPE indicates that the estimated time was less than the target time. Boothroyd and Dewhurst indicate that the user should expect an error of approximately 50%, and two of the three test products analyzed using the IDM fell within this expected error.

To test the ability of the OEM ANN to estimate the assembly times of a the test products in comparison to the CEDAR ANN, a non-parametric test (Mann Whitney) test was used to compare the medians. Each ANN results in 18,900 time estimates for each product. The percent error for each of 18,900 per product per ANN was calculated and used as the basis of comparison. The percent error was calculated as the difference between the predicted time and the target time, and normalized by the target time (see equation (2)).

$$\%\ \text{Error} = \frac{P - T}{T}\ x\ 100 \tag{2}$$

Where:
P: Predicted Time
T: Target Time

The alpha value used for this study is 0.05 and null and alternative hypothesis are the following:

H_0 : The population medians are equal

H_1 : The population medians are not equal

The Mann-Whitney statistical test suggests that there is sufficient evidence to reject the null hypothesis of equal medians between the two ANNs with a p = 0.000 < alpha = 0.05. The median percent error of the OEM ANN is -40.11 and the median percent error for the CEDAR ANN is -59.93. The 95% confidence interval for the percent error difference between the OEM ANN and the CEDAR ANN (OEM ANN - CEDAR ANN) is approximately between -18 and -13. The median of the OEM ANN is less than the median of the CEDAR ANN. *The results suggest that the neural network trained on products from a specific company from within the same product genre results in a lower percent error when estimating the assembly time using the Interference Detection Method Assembly Time Estimation Method.*

CHAPTER SIX
SPLIT INTERFERENCE DETECTION METHOD

In an effort to further improve the accuracy of the Interference Detection Method and the Boothroyd and Dewhurst assembly time estimation method, the handling codes and insertion codes are automated using separate techniques. While prior research indicated that there is an opportunity to reduce subjectivity by statistical means [60], the resulting range from using statistics alone for the insertion would leave the method inaccurate by producing a range too large to be useful (see Section 2.3.2). The handling codes are composed of mainly objective questions, and based on part geometry and part properties can be determined from the solid model (see Section 2.3.3). The insertion codes on the other hand are composed of a majority of subjective questions (see Section 2.3.3). Therefore, the insertion times are determined using a modified complexity connectivity method. The modified complexity method used part connection information within the assembly model to calculate a complexity vector. The complexity vector is then used as the input into the ANN to estimate an assembly time. This method potentially eliminates the need of a human inputting subjective information by using the modified complexity connectivity as a surrogate to the insertion time. The sum of the handling time and the insertion time for each part then results in the total assembly time of the product. This chapter will describe the approaches to estimate the handling and insertion codes respectively.

6.1 Handling Codes - Objective Questions

Eighteen of the twenty options from the first chart of handling questions required in the Boothroyd and Dewhurst assembly time estimate consist of objective questions [27]. The Split Interference Detection Method (SIDM) will retrieve handling codes and times based on objective information gathered from CAD software such as part size, part weight, and material. This portion of the method will need to calculate the following information related to each part:

- Symmetry (Alpha + Beta Symmetry)
- Size – longest bounding box edge length
- Thickness – shortest bounding box edge length
- Volume – related to weight by the mass density relationship

Information retrieved from the part CAD model will be used to determine the handling time associated Boothroyd and Dewhurst estimated assembly time (see Figure 6.1).

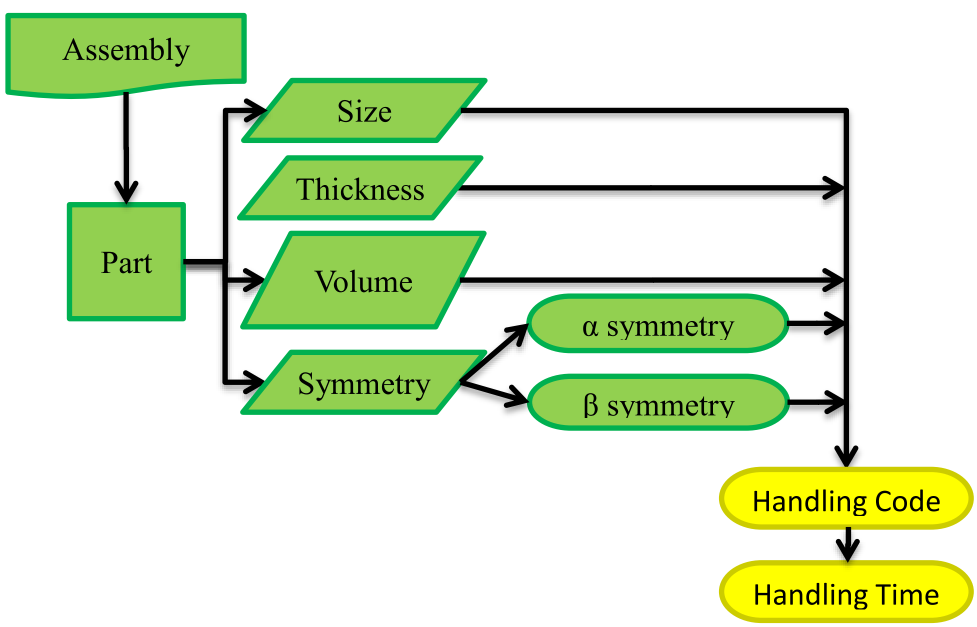


Figure 6.1: Handling Code Flow Chart

To determine the handling code and handling time of a part, the size and thickness of the part need to be determined. The envelope is the smallest rectangular box that can completely enclose the part (see Figure 6.2). The smallest box that can enclose the part and is aligned with the part global coordinate system is known as the bounding box (see Figure 6.2). The faces of the bounding box are aligned with the front, right, and top plane. The front, top, and right planes are the global part planes and are aligned with the global coordinate system shown by the triad showing the X, Y, and Z directions (see Figure 6.2). A survey of parts created by students indicated that generally parts are created by starting on one of the pre-created front, top, or right plane. From 100 parts examined, 97 of the parts were started on one of the pre-created planes.

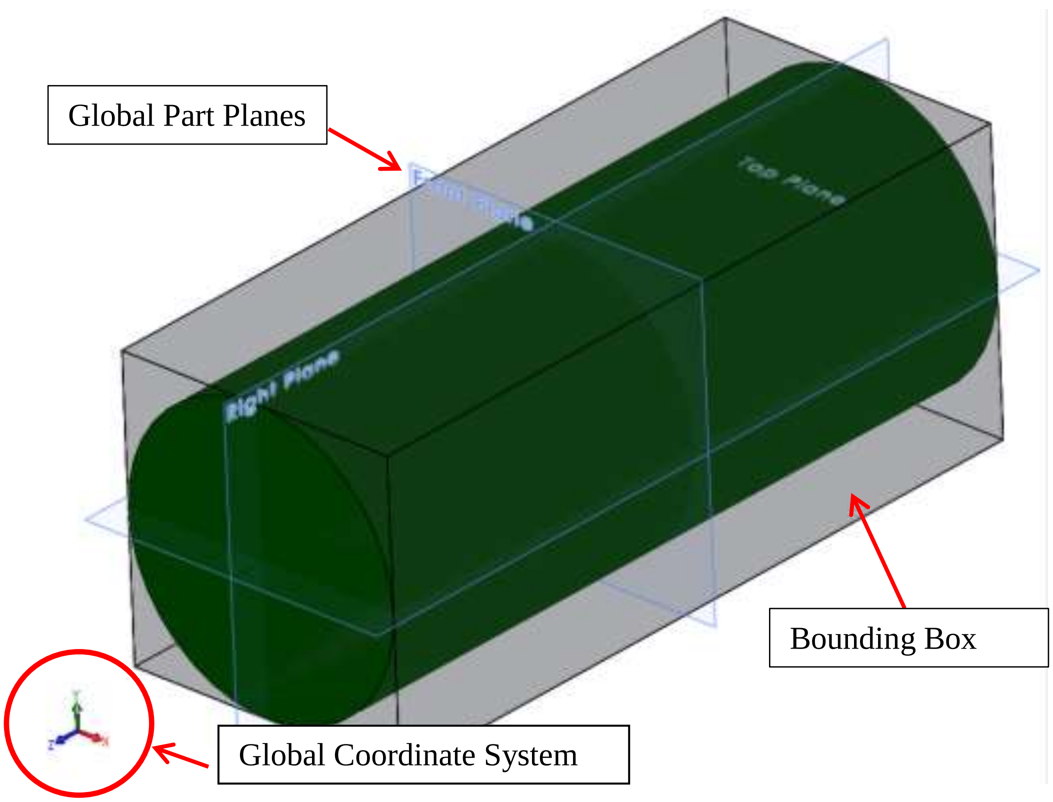


Figure 6.2: Bounding Box Aligned to Part Global Coordinate System

The size of the part is determined by the length of the longest edge of the bounding box. The length of the shortest edge of the bounding box is the thickness of the part (see Figure 6.3). In the case where the bounding box has all edges with equal lengths (cube), then the size and thickness have the same value. A call is then made to the mass properties function of the application protocol interface (API) to find the volume of the part. The API is an interface that allows a programmer to make calls from a standard programming language (C++ is used for this research) to the SW program. The function calls are specific to each commercial CAD system, but are common function calls found in all systems [74].

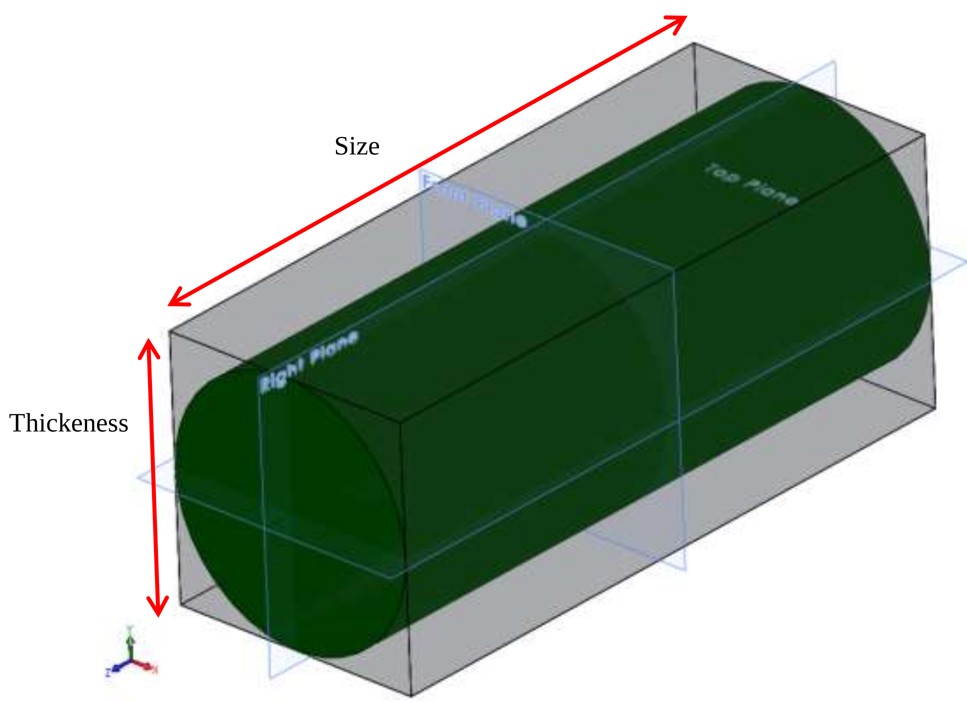


Figure 6.3: Bounding Box Aligned to Part Global Coordinate System

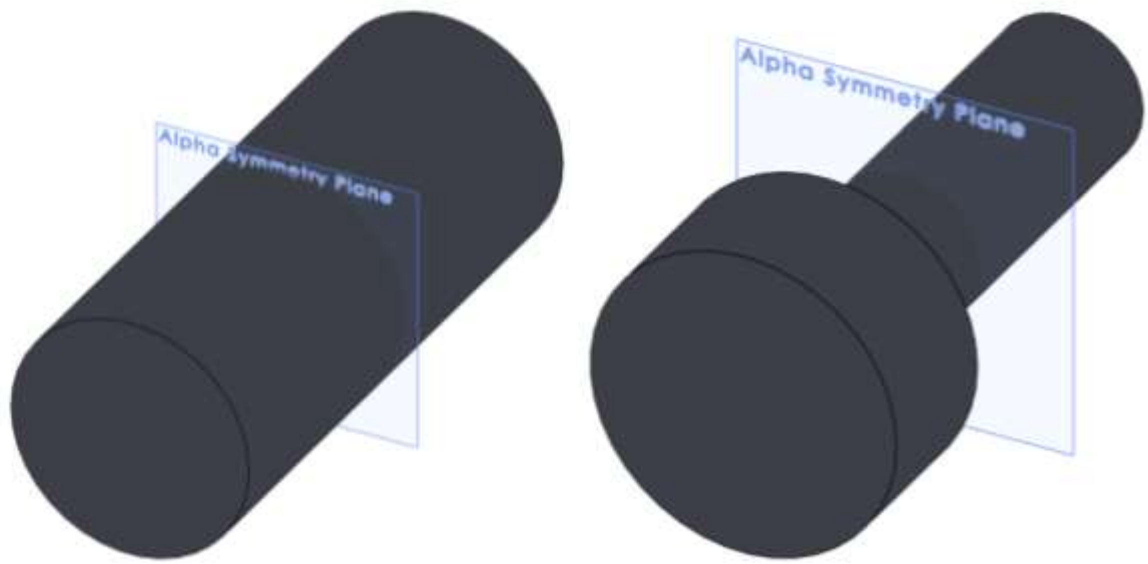
To determine the symmetry of each part, an algorithm was developed that that creates multiple cuts on the part and compares the volume before and after the operations.

This symmetry algorithm was specifically developed for the determining the handling code for Boothroyd and Dewhurst assembly time estimation method [4]. The handling portion of the Boothroyd and Dewhurst assembly time estimation method focuses on determining the alpha and beta symmetry of a part.

This symmetry algorithm is designed to determine a range of symmetry instead of an exact symmetry. This allows for the algorithm to operate on part geometry as opposed to previous symmetry detection methods which implement more computationally demanding techniques [26,75–77]. Previous methods often focus on topological features for comparison such as face loops or vertices, while others compare the arc lengths at different sectional views of a part [26,75–77]. While previous more computational expensive techniques return exact symmetry, the symmetry needed in estimating assembly times is more approximate and is based on symmetry ranges as described by Boothroyd and Dewhurst (see Chapter 6.1.3).The algorithm determines the two symmetry values: alpha and beta symmetry.

6.1.1 Alpha Symmetry Algorithm

The alpha symmetry is the symmetry along a plane perpendicular to the axis of insertion[4,78]. The axis of insertion is the axis parallel to the insertion direction of one part into another [4]. The alpha symmetry indicates if a part can be inserted either end first, or if there is a specific orientation for the part. For instance, a long slender cylinder (length > diameter) has an alpha symmetry of 180 degrees (see Figure 6.4). This indicates that it is possible to insert the part at every 180 degree rotation of the part. A bolt on the other hand is not symmetric about the alpha symmetry plane and can only be inserted every 360° rotation of the part (see Figure 6.4).



$\alpha = 180$ $\alpha = 360°$

Figure 6.4: Example of Alpha Symmetry

The alpha symmetry algorithm is a multi-step approach focused on volume comparison (see Figure 6.5).

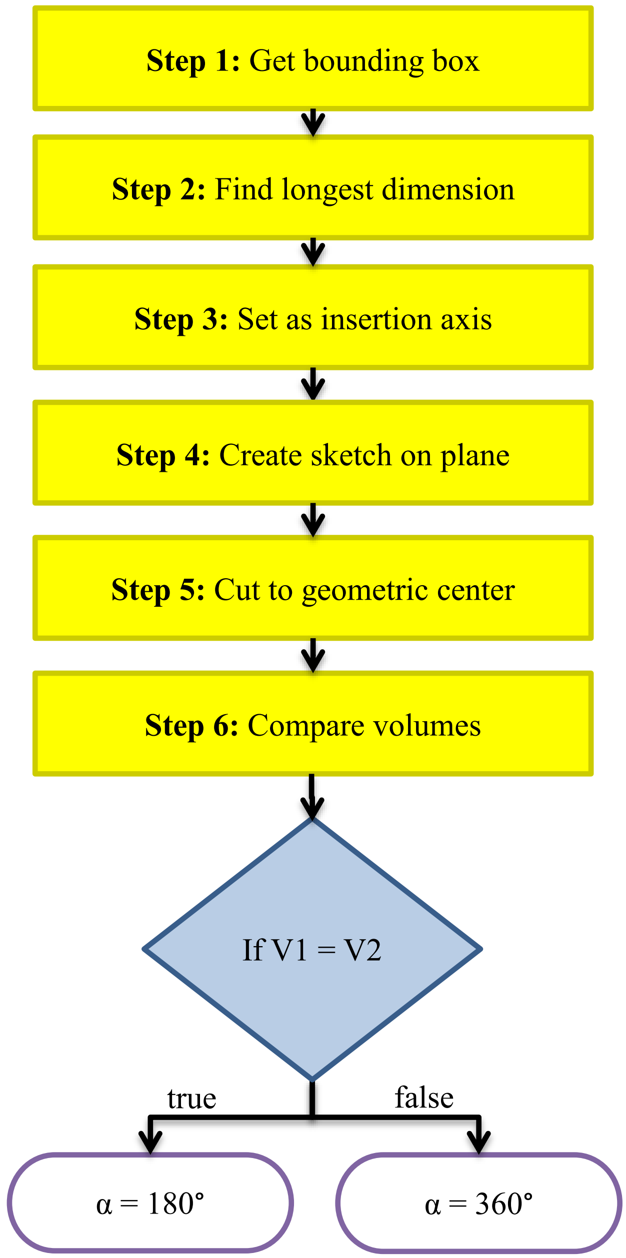


Figure 6.5: Alpha Symmery Algorithm Flow Chart

To demonstrate the alpha symmetry algorithm, an abstract model of a bolt is used (see Figure 6.6). The first step in the finding the alpha symmetry is to find the bounding box of the part. The bounding box of the part is the smallest axially aligned orthogonal box that can enclose the part. The term orthogonal is used in this case to indicate that each face of the bounding box is aligned with the global part planes (see Figure 6.6). The bounding box is the rectangular box outlined in black, and the part planes (“Front Plane”, “Right Plane”, and “Top Plane”) are the planes created by SolidWorks for every part. Each face of the bounding box is aligned (parallel) with the part planes. In this step the size, volume, and thickness are captured by a call to the API.

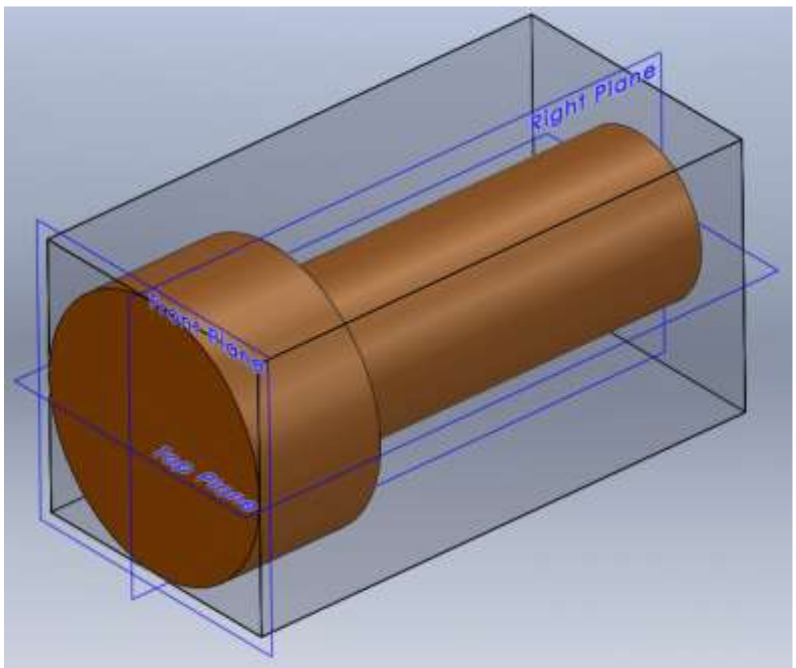


Figure 6.6: Bounding Box

The second step in the alpha symmetry algorithm is to find the axis of insertion. The axis of insertion indicates the direction that the part will be inserted during assembly. The assumption that is used for this research and has been used in previous research is the axis of insertion is assumed to coincide with the longest dimension of the part [78]. The axis of insertion is created from the center point of the plane that is normal to the longest dimension of the bounding box of the part. For the bolt, the longest dimension of the bounding box is along the right plane or top plane, therefore the front plane is the plane normal to the right plane (see Figure 6.6 and Figure 6.7). The axis is then created from the center point of the face of the bounding box that is aligned with the front plane. The axis is then created to the opposite side of the bounding box (see Figure 6.7). The axis of insertion is shown only to clarify the approach taken. The actual axis feature is not needed by the algorithm, but only the direction of the axis.

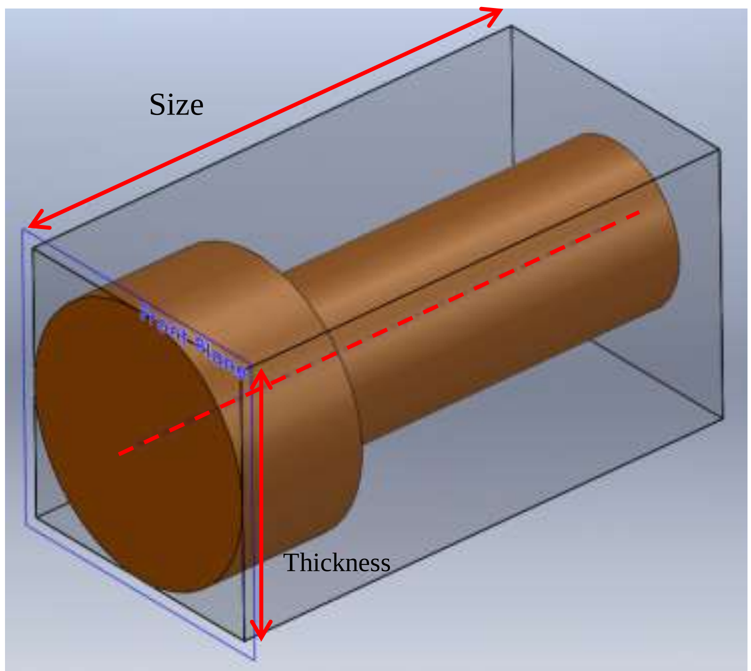


Figure 6.7: Axis of Insertion

After the axis of insertion and the associated normal plane are determined, creating a sketch is the fourth step in determining the alpha symmetry. A sketch is created on the plane normal to the axis of insertion. This algorithm sketches a circle (see Figure 6.8) on the normal plane, with two main conditions:

1. The diameter of the circle must be such that it encompasses the entire part
2. The center of the circle must be centered at the center of the normal plane. This is the point where the axis of insertion and normal plane intersect.

The radius of the circle is determined using the dimensions of the bounding box aligned with the normal plane. The diameter of the circle is set to be twice the length of the edge of the bounding box aligned with the normal plane. For instance, if the bounding box results in a dimension of 2x2x4 inches (height x width x length) and the normal plane is aligned with the height and width, the radius of the circle would be equal to four inches (2*2 inches). This value is used to ensure that the circle drawn will encompass the entire part when used for the cut. A circle is chosen for this research to reduce the number of input parameters needed. A circle is defined by a center point and a radius/diameter. A different shape such as a rectangle can be used with the same technique; however it would require additional parameters to define the shape and would therefore decrease the efficiency of the algorithm.

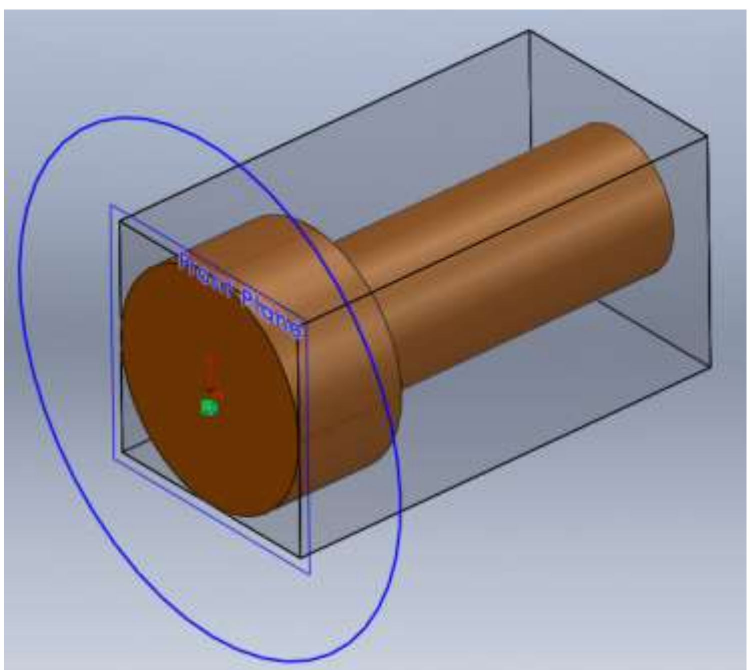


Figure 6.8: Sketch to Create Cut

Once a sketch is created on the plane normal to the axis of insertion, the sketch is then used to create an extrude cut (see Figure 6.9). The extruded cut is specified to a distance of half of the length of the longest edge of the bounding box of the part.

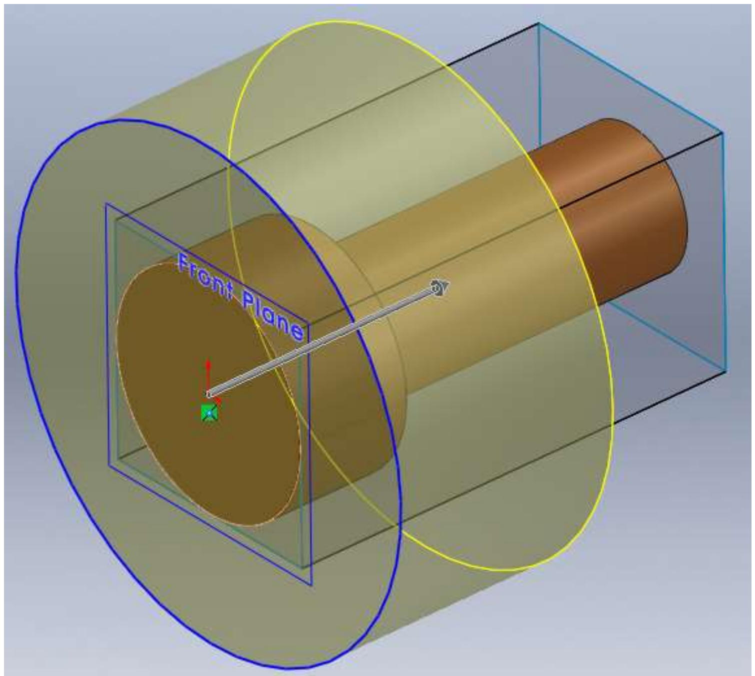


Figure 6.9: Cut to Geometric Center

The part can now be viewed as two separate bodies. The volume of the part that is being cut away is referred to as V1 and the remaining part after the cut is completed is referred to as V2 (see Figure 6.10).

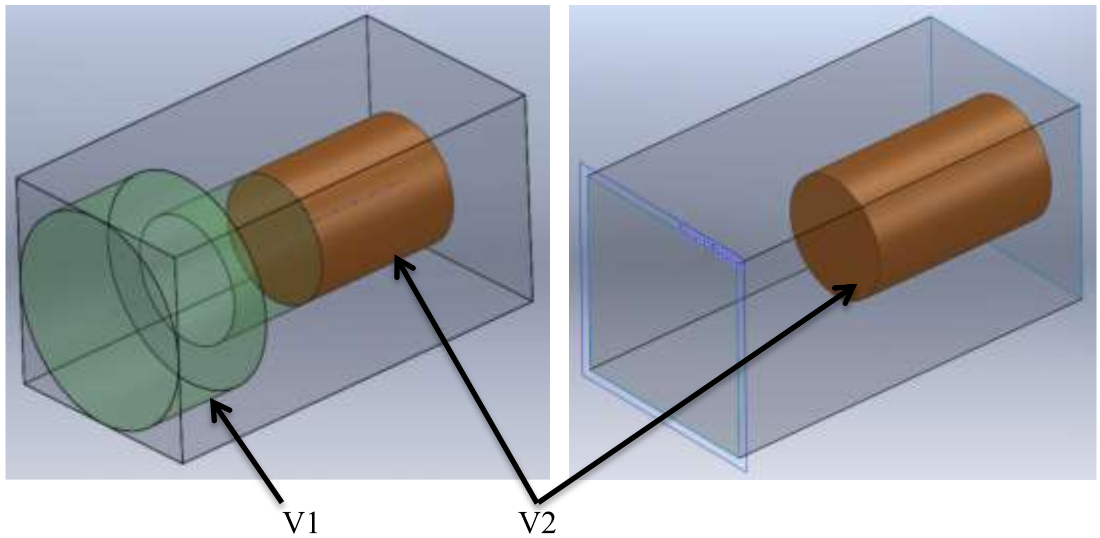


Figure 6.10: Compare Volumes

The volume of the two bodies are then compared to determine if they are equal with a 0.2% tolerance in order to account for numerical rounding errors. If the volumes are equal, then the part is considered to 180° or less in terms of alpha symmetry. For instance, a sphere would have an alpha value of 0^0, but this level of granularity is not necessary and is not captured in this symmetry algorithm. However, if M1 and M2 are not equal, then the part would be considered to have an alpha value of 360^0 (see Table 6.1).

Table 6.1: Alpha Values based on Part Volume

If:	Alpha Value (α)
V1= V2	180° or less
V1 ≠ V2	360°

While this method provides a quick estimate of the alpha symmetry of a part, there are certain cases that result in an inaccurate alpha value prediction. One example of a part that could return an inaccurate alpha value is a “dumbbell” that has different geometry but equal volume on each end (see Figure 6.11).

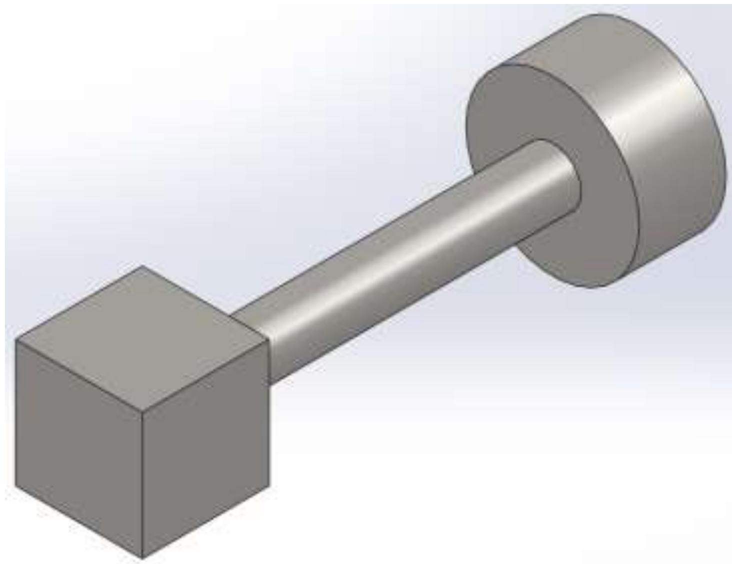


Figure 6.11: Dumbell Alpha Example

The algorithm to determine part symmetry is based on the part volumes, so unique cases exist such that the symmetry of the part is not correctly captured by the algorithm. When the dumbbell is cut to the geometric center, the volume V1 = V2, resulting in an alpha value of 180°. Visual inspection reveals that the dumbbell is only symmetric at angle of 360°, indicating that the dumbbell can only be inserted in one way. This increase the row of the Boothroyd and Dewhurst assembly time estimation method from row 1 to row 3 ((see Table 6.2). While unique cases exist that result in an incorrect alpha value, the effect this has on the handling time estimation is minimal when considering a full assembly model. An incorrect alpha value may result in a maximum handling time error of one second per part that is incorrectly evaluated (see Table 6.2).

Table 6.2: Exerpt of One Hand Handling Chart [4]

		Parts are easy to grasp and manipulate				
		T > 2mm			T ≤ 2 mm	
		S > 15 mm	6 mm ≤ S ≤ 15mm	S < 6 mm	S > 6 mm	S ≤ 6 mm
		0	1	2	3	4
(α+β) < 360	0	1.13	1.43	1.88	1.69	2.18
360 ≤ (α+β) < 540	1	1.5	1.8	2.25	2.06	2.55
540 ≤ (α+β) < 720	2	1.8	2.1	2.55	2.36	2.85
(α+β) = 720	3	1.95	2.25	2.7	2.51	3

The alpha symmetry algorithm is generally able to determine the symmetry of the part and will be demonstrated against a set of test cases drawn directly from the literature in Section 6.1.3.

6.1.2 Beta Symmetry Algorithm

The beta symmetry determines the rotational symmetry of a part about its axis of insertion [4]. The beta symmetry algorithm uses a similar approach as alpha symmetry, requiring three additional steps (see Figure 6.12).

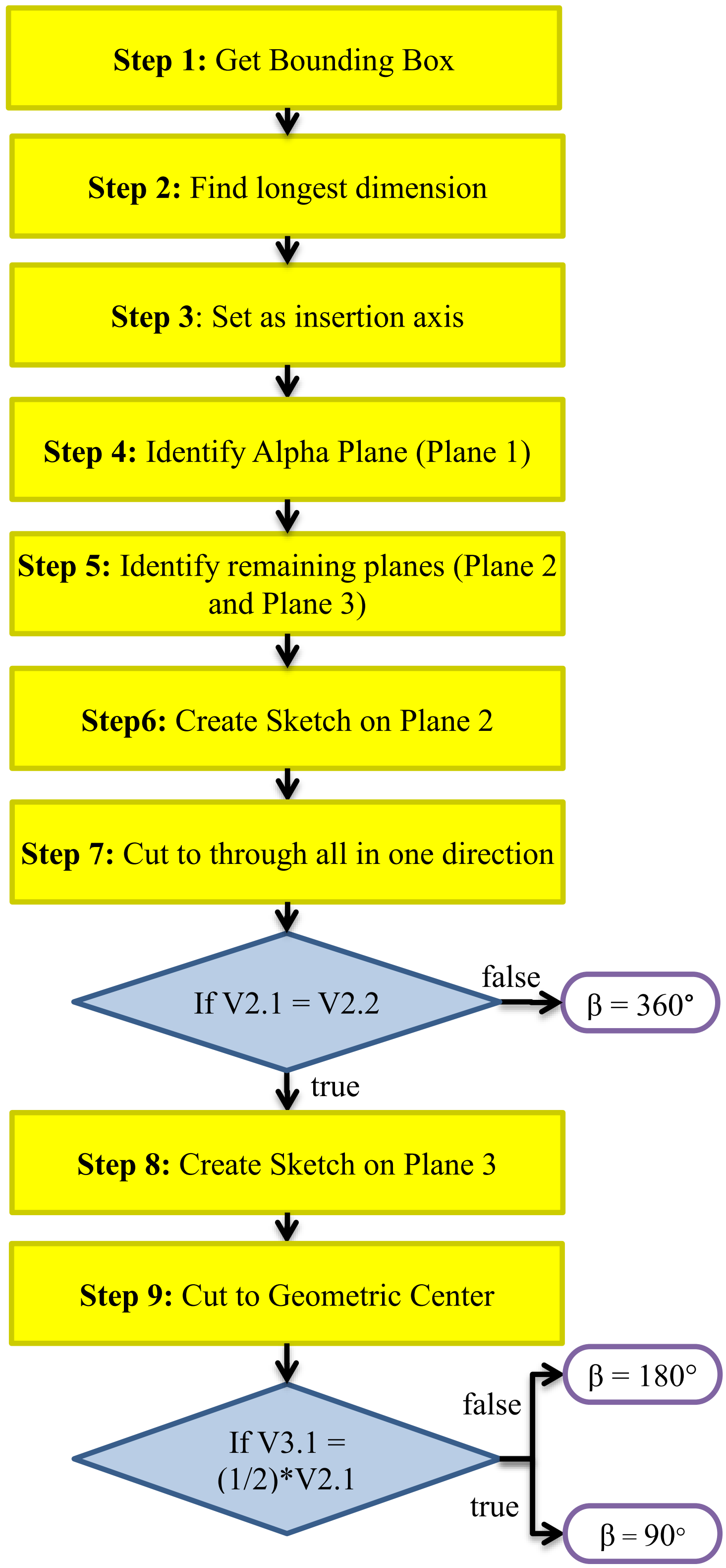


Figure 6.12: Beta Symmery Algorithm Flow Chart

The front plane was determined to be the plane normal to the axis of insertion from the alpha symmetry algorithm (see Figure 6.13). For the beta symmetry, the remaining two planes (the right plane and the top plane in this case) that are normal to the plane used for alpha symmetry are used to create the sketches for cutting (see Figure 6.13).

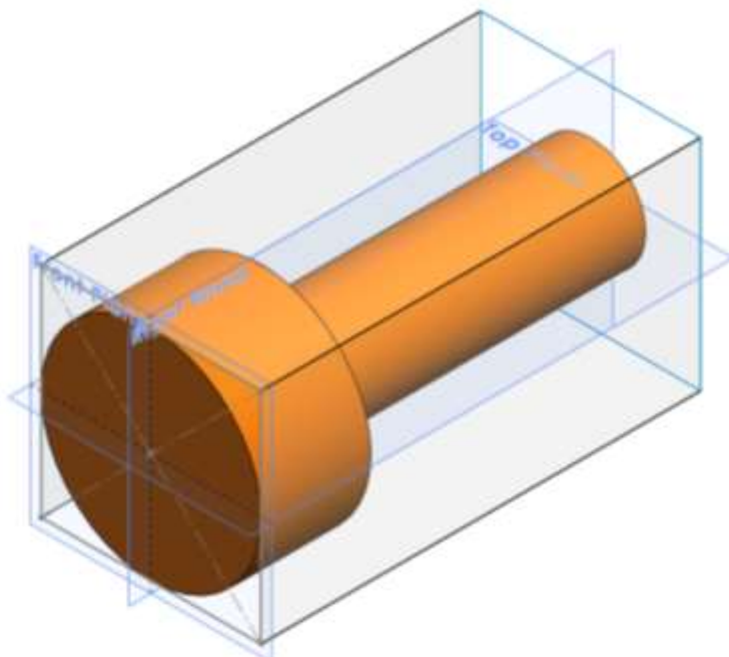


Figure 6.13: Bounding Box for Beta Symmetry

The first cut in determining the beta symmetry is created on the right plane (see Figure 6.14). A circle sketch is created that is centered at the intersection of the right and front planes, which is also the center of the respective face of the bounding box. The radius of the circle is determined based on the size of the bounding box measure for the part. The radius is set as the minimum diameter to encompass the longest edge of the bounding box. The circle is again chosen as the cutting shape to minimize the number of parameters and to simplify the subtraction volume construction.

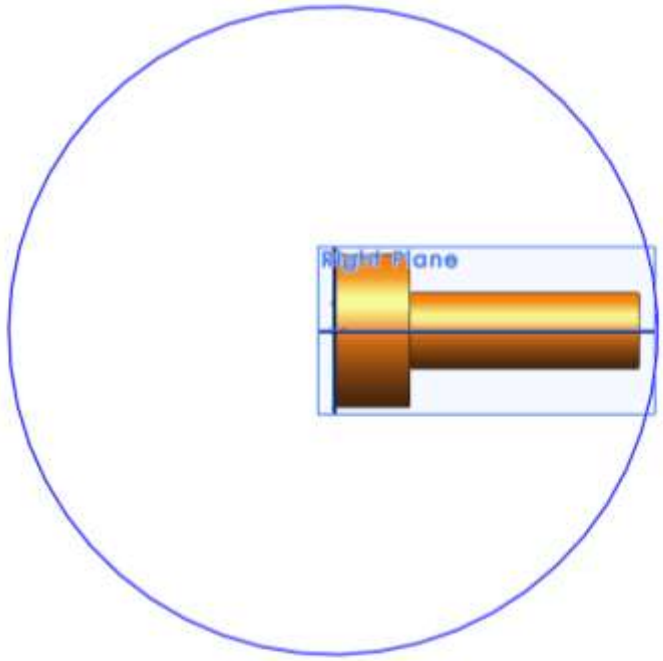


Figure 6.14: Circle Sketch for First Cut for Beta Symmetry

The circle sketch created is then used to cut through all in one direction of the part. This cut will remove half of the part based on the location of the bounding box enclosing the part. The volume of the remaining body (V2.1) is compared to the volume of the cut body (V2.2) (see Figure 6.15).

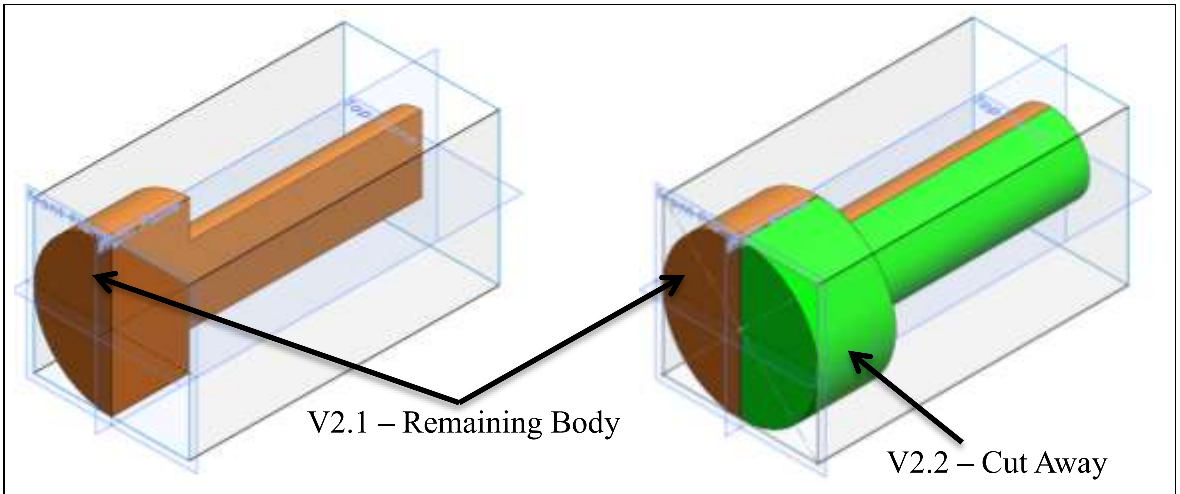


Figure 6.15: First Cut for Beta Symmetry

If the volume of V2.1 is not equal to the volume of V2.2 then the part has beta symmetry of 360°. If the volume of V2.1 is equal to the volume of V2.2, then additional steps are taken to determine the beta symmetry. Continuing to operate on the remaining body (V2.1) a circle is sketched on the third and final part plane, the top plane (see Figure 6.16). Once again the diameter of the circle is determined from the size of the bounding box as discussed for first cut for beta symmetry.

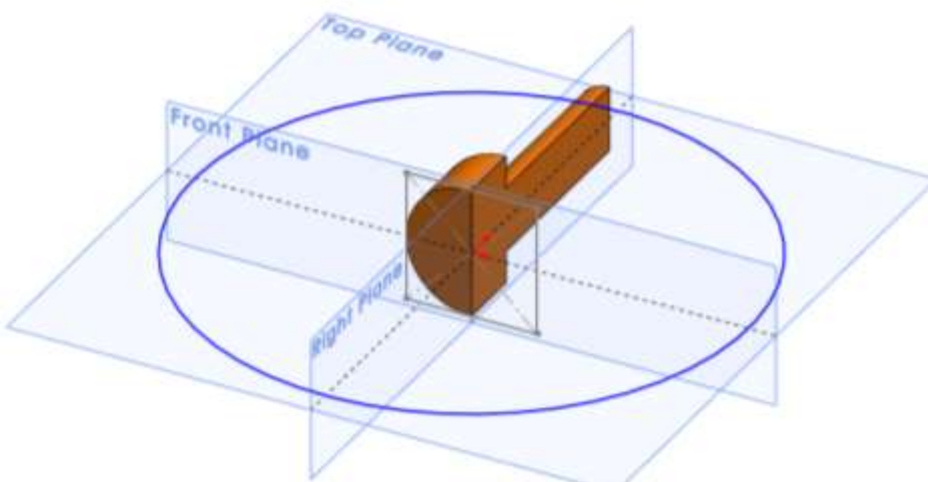


Figure 6.16: Circle Sketch for Second Cut for Beta Symmetry

This circle is cut 'through all' in one direction to leave the body V3.1 (see Figure 6.17). The volume V3.1 is compared to the volume V2.1.

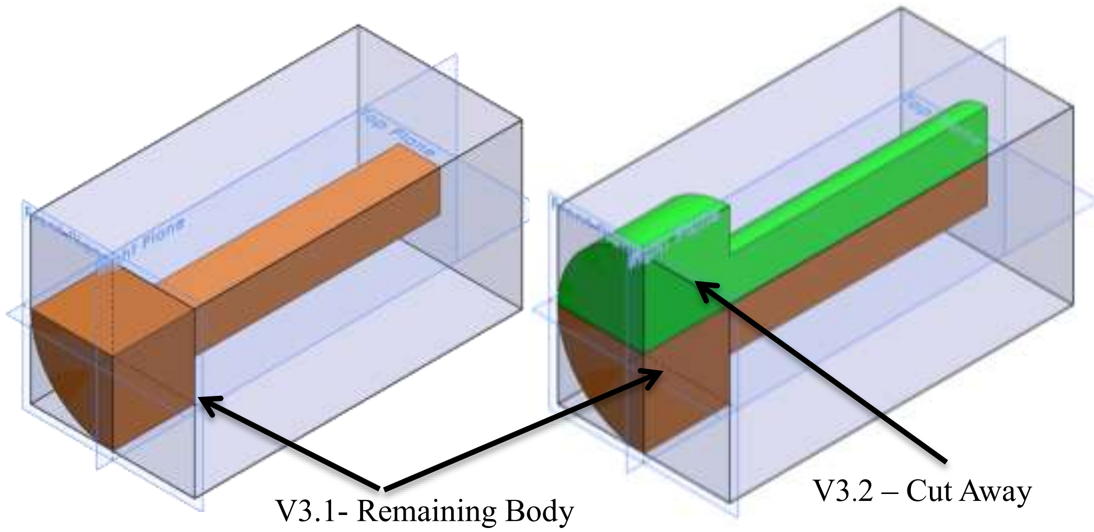


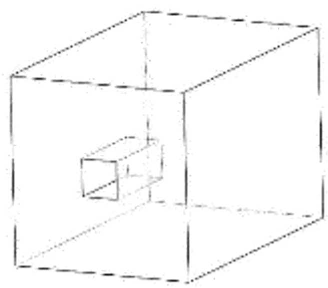
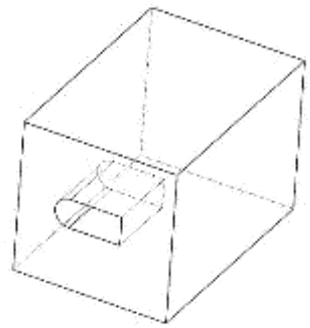
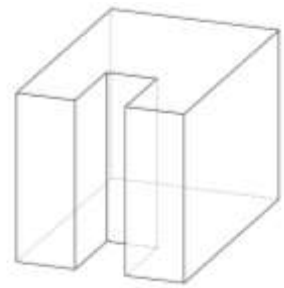
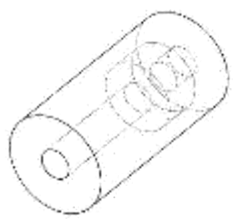
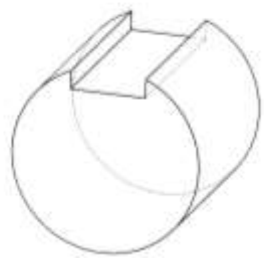
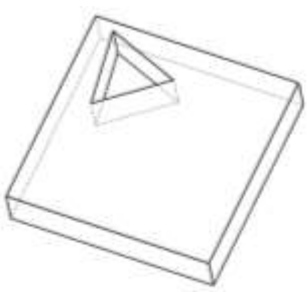
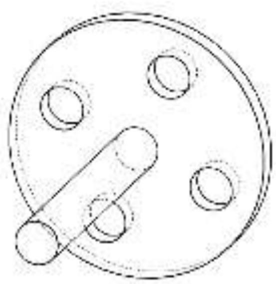
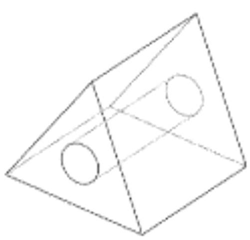
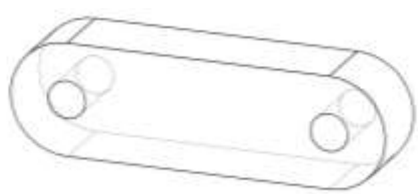
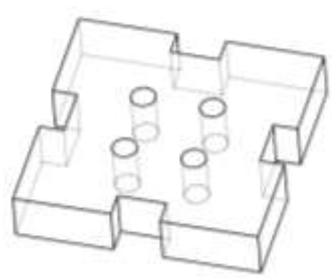
Figure 6.17: Second Cut for Beta Symmetry

If V3.1 is equal to half of V2.1 (or a quarter of the entire volume of the part), then the part has a beta symmetry of 90°. If V3.1 is not equal to half of V2.1, or a quarter of the volume of the original part, then the part is only has beta symmetry of 180°. Similar to the alpha algorithm, the beta algorithm only determines symmetry to a level of granularity of 90° increments. A long slender cylinder should result in beta symmetry of 0° since it can be inserted at any rotational angle; however this algorithm returns a value of 90°. While this is a limitation of the algorithm in general, for this application in determining handling codes for the Boothroyd and Dewhurst assembly time estimation method, this level of granularity is sufficient because the row groupings are distinguished by 180°. This limitation is further discussed in the next section as the symmetry algorithm performance in determining the value of alpha and beta is compared to previous literature.

6.1.3 Symmetry Test Cases from Literature

To test the performance of the symmetry algorithm, a set of test cases from literature is used for evaluation purposes [78]. Twelve parts (see Table 6.3) are used to compare the performance of this symmetry algorithm to another symmetry algorithm found in research literature [78]. The benchmark parts are chosen from research literature source and are defined external of this research to ensure an objective demonstration of the symmetry method in comparison to previous research.

Table 6.3: Symmetry Test Parts (Adapted from [78])

		
Test Part 1	Test Part 2	Test Part 3
		
Test Part 4	Test Part 5	Test Part 6
		
Test Part 7	Test Part 8	Test Part 9
		
Test Part 10	Test Part 11	Test Part 12

The specific ranges of symmetry that are required to determine the handling code from the Boothroyd and Dewhurst assembly time estimation method have been categorized for discussion in this research (see Table 6.4). Each range of symmetry is directly linked to a row from the Boothroyd and Dewhurst assembly time estimation method.

Table 6.4: Symmetry Ranges and Associated Boothroyd and Dewhurst Row Number

Symmetry Range	Symmetry Category	Boothroyd and Dewhurst Row Code
$\alpha+\beta < 360^o$	1	1
$360^o \leq \alpha+\beta < 540^o$	2	2
$540^o \leq \alpha+\beta < 720^o$	3	3
$\alpha+\beta = 720^o$	4	4

The alpha and beta values using the Ong algorithm are compared to the symmetry results from the SIDM symmetry algorithm. Specifically, the total value of the alpha plus beta determines the symmetry category (see Table 6.4).

Table 6.5: Symmetry Test Case Results

#	Ong Alpha [78]	Ong Beta [78]	SIDM Alpha	SIDM Beta	Ong Total	SIDM Total	Ong Boothroyd and Dewhurst Row Code	SIDM Boothroyd and Dewhurst Row Code
1	360°	90°	360°	90°	450°	450°	**2**	**2**
2	360°	360°	360°	360°	720°	720°	**4**	**4**
3	360°	180°	360°	180°	540°	540°	**3**	**3**
4	360°	0°	360°	90°	360°	450°	**2**	**2**
5	360°	0°	360°	90°	360°	450°	**2**	**2**
6	180°	360°	180°	360°	540°	540°	**3**	**3**
7	360°	360°	360°	360°	720°	720°	**4**	**4**
8	360°	90°	360°	90°	450°	450°	**2**	**2**
9	360°	360°	360°	360°	720°	720°	**4**	**4**
10	180°	120°	180°	360°	300°	540°	**1**	**3**
11	180°	180°	180°	180°	360°	360°	**2**	**2**
12	180°	90°	180°	90°	270°	270°	**1**	**1**

Of the twelve parts tested, only part 10 symmetry row code did not match between SIDM and Ong. The possible symmetry values that can be returned using the SIDM are 90°, 180°, and 360°. Since the first cut created to determine the beta symmetry for part 10 results in two bodies that do not have an equal volume, the SIDM algorithm results in symmetry value of 360° for beta. The correct beta value as determined from Ong is 120°. This part serves as an example of a unique part in which the symmetry is not correctly determined using the SIDM. As discussed earlier, an incorrect symmetry estimate results in a maximum of one second time estimation difference.

One limitation of the SIDM is the symmetry algorithm was specifically designed to determine the symmetry for use in finding the handling code/time of a product. The handling code does not require granularity for the individual alpha and beta values. For instance, the SIDM cannot determine the beta granularity of a part that is completely symmetrical resulting in a beta value of 0. Instead, the minimum value returned by the SIDM is 90°. Part 4 for example is completely symmetric about its axis of insertion (beta symmetry), and therefore has a beta value of 0°. The SIDM algorithm returns a value of 90° for beta, but due to the range of alpha plus beta values to remain within category 2, the distinction between 0° and 90° is not necessary for table value look-up within the Boothroyd and Dewhurst database. For this research, the symmetry accuracy is sufficient to quickly extract symmetry values. Further refinement of the method is possible by introducing additional slicing volumes for additional symmetry granularity.

6.2 Insertion Codes - Subjective Questions

Unlike the handling codes, all of the insertions questions needed to determine the insertion codes are subjective (see Section 2.3.2)[27]. For humans, this may not seem problematic, but, as seen in the pen study, the insertion estimates resulted in a large variation in the time estimate, reducing the confidence in the estimated assembly time (see Section 2.3.3). Therefore, the insertion times will be determined using a modified connectivity complexity method that is objectively calculated and based on historical data that can be updated to improve the accuracy.

The eleven CEDAR products used to train the ANN for earlier research (see Chapter 5.1 and Table 6.6) will once again be used to train an ANN for the insertion times. When the Boothroyd and Dewhurst assembly time estimation method was manually conducted for the CEDAR products, each product required a handling time and insertion time. To train an insertion only ANN, the complexity vector for each product will be used as the input and the insertion portion of the assembly time will be used as the target time. The eleven products and the respective insertion times (see Table 6.6) are used to train an ANN and will referred to as ‘Insertion Only ANN’. The same ANN design used earlier in this research will once again be used (see Chapter 4.3). Three of the fourteen products will be withheld for testing purposes, and the remaining eleven products will be used to train the ANN.

Table 6.6: Seperated Handling and Insertion Times of CEDAR Products

Product Name	Training / Testing	Handling Time	Insertion Time	Total Assembly Time
Stapler	Testing	39.01	84.50	123.51
Flashlight	Testing	24.40	51.00	75.40
Blender	Testing	88.76	166.00	254.76
Ink Pen	Training	13.40	21.00	34.40
Pencil Compass	Training	22.83	46.50	69.33
Electric Grill	Training	44.08	77.00	121.08
Solar Yard Light	Training	32.29	96.50	128.79
Bench Vise	Training	32.69	111.00	143.69
Electric Drill	Training	45.65	144.00	189.65
Shift Frame	Training	65.70	248.00	313.70
Food Chopper	Training	88.12	228.50	316.62
Computer Mouse	Training	25.65	56.50	82.15
Piston	Training	15.01	33.00	48.01
3- Hole Punch	Training	42.38	103.00	145.38

This portion of the research will implement a modified connectivity method to estimate only the insertion portion of the Boothroyd and Dewhurst assembly time estimation method. The ANN trained only to predict the insertion time of a product will be tested by adding the predicted insertion time to the calculated handling time, and compared to the overall assembly time of the product as determined by the IDM.

6.3 Comparison of Split Interference Detection Method to Interference Detection Method

To test the complete SIDM, the handling time and insertion time for each of the three test products (stapler, flashlight, and blender) was calculated. The total assembly time estimate for each part is the sum of the handling time and the insertion time. A modified complexity connectivity method uses an ANN to estimate the insertion time of each part. The ANN returns 18900 insertion time estimates for each product. Each of these 18900 time estimates is added to the single handling time objectively determined from the CAD model (see Figure 6.6).

Table 6.7: Example SIDM Results for Stapler

Estimate #	Predicted Handling Time [s]	Predicted Insertion Time [s]	Predicted Total Time [s]	Target Handling Time [s]	Target Insertion Time [s]	Target Total Time [s]
1	37.72	27.56	55.22	39.01	84.50	123.51
2		33.40	61.06			
3		99.61	127.27			
⋮		⋮	⋮			
18899		88.42	116.08			
18900		119.08	145.74			

To determine if there is a statistical difference between the total predicted assembly time from the SIDM and IDM methods, a Mann Whitney test (Wilcoxon rank-sum test) will be used to compare the percent error of each of the methods[79–81]. The two ANNs were compared using the rank-sum test with a 0.95 confidence interval. Due to a small sample size (three test products and eleven training products), a wider confidence interval is used. The null hypothesis for this test is that there is no difference in percent error from target time when predicting the assembly times of products using the Full ANN and the Reduced ANN. To calculate the percent error for each ANN, the difference between the predicted time and the target time was normalized using the target time (see equation (3)).

$$\% \text{ Error} = \frac{P - T}{T} \; x \; 100 \qquad (3)$$

Where:

P: Predicted Time
T: Target Time

The percent error for each of the 18900 assembly time estimates for each of the three test products is calculated for the IDM and the SIDM. The results of the Mann Whitney test provide sufficient evidence that the medians of the IDM and the SIDM are not equal with a p-value of less than 0.0000. The median percent error value of the IDM and SIDM are 11.72 and -35.12 respectively (see Table 6.8).

Table 6.8: Median Values of IDM and SIDM for CEDAR Test Products

Method	Number of Estimates for Three Test Products	Median Percent Error of Three Test Products
IDM	56700	11.72
SIDM	56700	-35.12

The results of the statistical comparison of the IDM and SIDM suggest that the medians of the two methods are not equal. The negative sign in the median error for the SIDM indicates that value of the predicted time is less than the value of the target time. Therefore, the absolute value of the error the IDM is less than the absolute value of the error of the SIDM. These results indicate that the IDM can predict the assembly time with a lower percent error than the SIDM.

CHAPTER SEVEN
STATISTICAL ANALYSIS OF COMPLEXITY METRICS

This research thus far has focused on exploring the extent to which the IDM and the SIDM can be used to predict assembly times. This chapter focuses on the complexity vector that is used to represent the assembly model in the IDM and SIDM. The goal of this chapter is to understand which of the twenty nine complexity metrics are most significant in estimating the assembly time of three test products.

The complexity connectivity method is based on a complexity vector of twenty nine complexity metrics. The complexity vector has been used as the input vector of information as the complexity connectivity method has evolved from a linear regression time estimate [66], to the use of an ANN [31] for the AMM and IDM/SIDM (Chapter Three and Chapter Six). However, the complexity vector itself has not been evaluated to determine the necessity of all of the complexity metrics.

A statistical study is conducted to determine if all twenty nine of the current complexity metrics are significant and needed in the automated assembly time estimation method. Reduction of the number of complexity metrics being used can reduce the computation effort required by the method, and may provide an additional benefit of improving the accuracy and or the repeatability of the assembly time estimate [28]. This portion of the research will help determine the necessity of the current twenty nine metrics, and also provide a process for evaluating necessary metrics for future application of the complexity connectivity method. This chapter uses a multistep approach to determine the significant complexity metrics and test the reduced set to ensure that errors in time estimates are less than or equal to the full set of complexity metrics.

A full linear regression analysis is used to determine the significant complexity metrics. The reduced set of complexity metrics is then be used to train a new ANN (Reduced ANN). The reduced ANN is next used to predict the assembly time of the test products. The estimated assembly times from the same training products and test products are compared for the full ANN and the reduced ANN. This multistep approach (see Figure 7.1) is used to determine significant factors and if the reduced set can estimate assembly times with equal or lesser error. The development of this process is not the focus of this research, but is a necessary step in the improvement and development of the complexity connectivity method. The study of the complexity metrics are used to answer RQ2.2 and can also be applied to improve previous and future research involving the use of the complexity metrics [23,28,30,63].

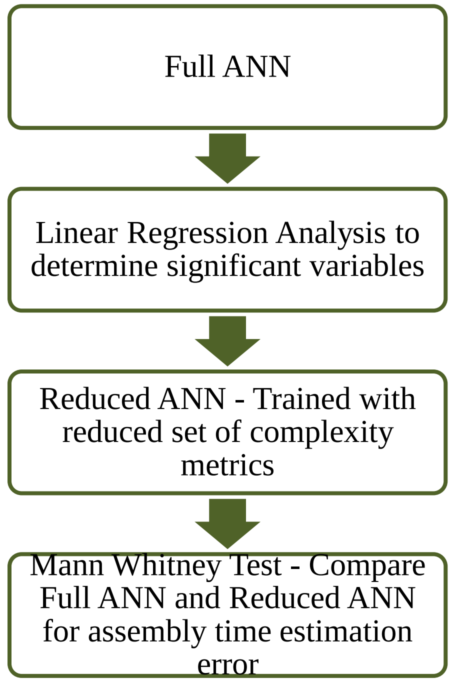


Figure 7.1: Approach for Reduction of Complexity Metrics

7.1 Regression Analysis

The ANN design used in this research consists of 189 architectures with 100 repetitions each. This results in a total of 18900 assembly time point estimates. A linear regression analysis is used to find a relationship between the assembly times and the complexity vector for each of these estimates. The 18900 time estimates are used as the response variable for the regression analysis and each of the 29 complexity metrics are used as the dependent variables.

The results of the regression analysis suggest that of the twenty nine complexity metrics, fifteen of the metrics are linear transformation of the others. This is indicated by the "---" in the p-value column (see Table 7.1). From the remaining fourteen complexity metrics, two of the complexity metrics (x17 and x24) are not statistically significant variables ($p > alpha = 0.05$) in predicting assembly time.

Table 7.1: Regression Analysis of Complexity Metrics

Complexity Metric:	Coefficient	pValue
(Intercept)	0	---
x1	24.40	1.28e-217
x2	0	---
x3	0	---
x4	-3.70	3.35e-128
x5	0.75	9.17e-40
x6	-10.10	2.38e-14
x7	0	---
x8	0	---
x9	0.04	9.77 e-4
x10	1.57	6.88e-22
x11	24.35	7.46e-22
x12	0	---
x13	-1.39	2.00e-59
x14	0.49	4.39e-87
x15	6.45	1.50e-64
x16	0	---
x17	0.40	0.62
x18	0	---
x19	0	---
x20	0	---
x21	0.54	5.46e-81
x22	-9.83	9.11e-51
x23	0	---
x24	1.51	0.74
x25	0	---
x26	0	---
x27	0	---
x28	0	---
x29	0	---

The resulting linear model for this data set is represented by the following equation:

$$Assembly\ Time = 24.401 * x1 - 3.7018 * x4 + 0.7516 * x5 - 10.107 * x6 + 0.044927 * x9 + 1.5794 * x10 + 24.353 * x11 - 1.393 * x13 + 0.49322 * x14 + 6.4558 * x15 + 0.54954 * x21 - 9.8392 * x22 \tag{4}$$

Each x-value (see Equation (4)) represents one of the twenty nine complexity metrics. The significant factors determined from the regression analysis are highlighted (see Table 7.2). These significant dependent variables are used to train a new neural network to test the predictive ability using the reduced complexity vector. One observation of interest is the regression analysis resulted in at least one significant metric form each of metric groupings: Decomposition, Centrality, Interconnections, and Size. Specifically, five of the fifteen identified significant metrics belong to the interconnections grouping. This follows closely with the fact that the graphs are generated based connections between parts within the assembly.

Table 7.2: Statistically Significant Complexity Metrics

Complexity Metrics	Size	Dim		elements	x1
				relations	x2
		Conn		DOF	x3
				connections	x4
	Interconnection	Shortest Path		Sum	x5
				Max	x6
				mean	x7
				density	x8
		Flow Rate		Sum	x9
				Max	x10
				mean	x11
				density	x12
	Centrality	Betweenness		Sum	x13
				Max	x14
				mean	x15
				density	x16
		Clustering Coefficient		Sum	x17
				Max	x18
				mean	x19
				density	x20
	Decomposition	Ameri Summers			x21
		Core Numbers	In	Sum	x22
				Max	x23
				mean	x24
				density	x25
			Out	Sum	x26
				Max	x27
				mean	x28
				density	x29

7.2 Reduced ANN Comparison to Full ANN

The reduced set of complexity metrics were used to train a new ANN (named Reduced ANN) for comparison with the original ANN (named Full ANN) that included all twenty nine of the complexity metrics. The Full ANN (as discussed in section 4.3) and the Reduced ANN were both trained and tested using the set of fourteen electromechanical products. For each ANN, the same eleven products were used for training and three products were reserved for testing. To compare the performance of the Full ANN and the Reduced ANN the percent error from the target times were evaluated using the Mann Whitney test (Wilcoxon rank-sum test).

The two ANNs were compared using the rank-sum test with a 0.95 confidence interval. The null hypothesis for this test is that there is no difference in percent error from target time when predicting the assembly times of products using the Full ANN and the Reduced ANN. To calculate the percent error for each ANN, the difference between the predicted time and the target time was normalized using the target time (see equation (5)).

$$\%\ \text{Error} = \frac{P - T}{T}\ x\ 100 \tag{5}$$

The percent error for each of the 18900 predicted time estimates from each ANN was calculated. The results provide sufficient evidence to reject the null hypothesis of the two ANNs having equal percent error in predicting assembly times. The median percent error for the Full ANN and the Reduced Set is 9.5% and 5.5% respectively. The 95% confidence interval suggests that the difference percent error of the original set and the reduced set is [1.6, 3.9]. The statistical test provides evidence that the mean error of the Full ANN and the Reduced ANN are not equal. The Reduced ANN has a mean error of 5.5% which is less than the error of the Full ANN suggesting that the Reduced ANN can estimate the assembly time with less percent error than the Full ANN. However, this reduced set was only determined based on the exploration of the initial twenty nine metric vector. Without additional testing, the justification of the use of the reduced complexity vector is limited to this data set. This portion of the research does provide a method that can be reapplied to other complexity based modeling schemes. This results suggest that the full complexity vector should initially be used and the process used in this chapter can be applied to the data set at hand.

7.3 Conclusions on Statistical Analysis of Complexity Metrics

This study used a linear regression analysis to form a model representing the complexity vector composed of twenty nine metrics and the predicted assembly times resulting from the Full ANN. The model was then used to reduce the complexity vector from twenty nine metrics to twelve metrics. To test the performance of the ANN with the reduced set of complexity metrics a new ANN (Reduced ANN) was trained with the reduced complexity vector and was compared to the Full ANN using the Mann Whitney test. The results indicate that error between the predicted times output by the Full ANN and the Reduced ANN are not equal and that the Reduced ANN has a lower percent error in the predicted assembly times. The results of this study indicate that there is an opportunity to reduce the computational effort required in computing the complexity vector by using a reduced complexity vector. Furthermore, anecdotal evidence suggests an opportunity to eliminate the need for the computationally expensive ANN, and replace it with a regression model to predict assembly times. To obtain this type of relationship a larger product set would be required and statistical validation of using a linear model as opposed to an ANN, however this is out of the scope of this research. With low sample size, the neural network provides a stochastic modeling approach for estimating assembly times.

CHAPTER EIGHT
CONCLUSIONS AND FUTURE WORK

The research presented in this dissertation focused on designing an automated assembly time estimation method that was accurate, repeatable, and minimized the amount and detail of information needed from designers. Opportunities exist in academia and industry to apply this assembly time estimation tool, improve the tool, and also use the fundamental design of the method to improve other aspects of engineering design.

8.1 Intellectual Merit

The proposed research demonstrates an automated assembly time estimation tool that can support designers throughout the design process. With increasing product costs, industry is looking for ways to maximize profit by decreasing manufacturing costs [3,4,13,15]. Previous research has shown that early stages of the design process account for approximately 50-70% of product cost [4,37,50]. However, one general limitation of design for assembly methods is the tools and methods are generally reserved for detailed design stage due to the amount of detail required about the parts and the time required to apply [82]. This research aims to provide a design for assembly time estimation tool that can be used iteratively throughout the design process by reducing the time required for analysis and the amount and level of detail of information required to perform the analysis. This tool will allow the manufacturing industry gain the benefits of improving product design in the early stages of the design process, and in turn reduce time to market of products as well as providing customers with an increase in product quality [1,4,7,10,11,82,83].

8.2 Broader Impact

The focus of this research is designing an automated assembly time estimation method, but the core contributions of this research provide a basis for a variety of applications. This research can be applied to other areas of academia and manufacturing. An example of each of these includes the use of a similar method to predict the amount of credit that individual questions on a test should be worth [84]or for manufacturing to predict which design for assembly or design for manufacturing guidelines should be applied to a product.

For example, a complexity vector can be created to represent the difficulty of problems on an engineering exam. The complexity metrics for this type of application would be substantially different however and an opportunity exists to define a set of metrics to represent the difficulty of the problem. These metrics may include factors such as the amount of time needed for the instructor to solve the problem, the college years standing (freshman, sophomore, junior, or senior) of the students taking the test, and the total number of problems on the test. A neural network would have to be trained on the input data set with provided problem difficulties, and may be applied to quickly distribute test points on future exams.

One area of extreme interest is using a similar approach in predicting which design for assembly guidelines or design for manufacturing guidelines should be applied to a product. For instance, based on the connectivity graph of a product, certain guidelines could be suggested for implementation. A simple example can be perceived between the number of parts and the number of relations, to suggest implementing a design for assembly guidelines of reducing the number of parts. By training a neural network on many products and the design for assembly or design for manufacturing efforts implemented to improve the design, a tool could be envisioned to help guide designers on how to improve a product.

8.3 Future Work

Based on the research conducted and presented in this dissertation, additional research has been identified to further improve the effort in designing an automated assembly time estimation method. This research has identified areas of interest to further improve this method itself and motivate future research in this area.

8.3.1 Training and Testing Sets

One of the limitations of this research is the limited sample size for training and testing of the automated assembly time estimation method. This research can provide anecdotal evidence of the power of an automated assembly time estimation tool, and can be used as motivation to gather additional data from local manufacturers. Increasing the sample size will help to further refine and validate the method, creating a better understanding of the ability of the tool to predict actual assembly times as seen in industry.

8.3.2 Software Independence

Another area of future work is the separation of the automated assembly time estimation method from the current implementation within Matlab. Currently, Matlab is used to analyze the connectivity graphs and calculate the connectivity vector. The metric values are then used to first train an artificial neural network, run through Matlab, and then for the prediction of assembly times. Ideally this portion of the method will be transferred into a standard programming language, such as C++, and integrated into the SolidWorks API code. This would fully automate and integrate the assembly time estimation process, allowing the add-in to find and create connectivity graphs, use the connectivity graph to calculate the complexity vector, and use the complexity vector(s) to train an ANN or to calculate the assembly time using the vector as the input into a previously trained ANN. This level of integration and automation is appropriate for the potential commercialization of the solution.

8.3.3 Neural Network Design

Additional investigation can be conducted on the operation and use of artificial neural networks. Many different types of artificial neural networks can be used as prediction and data mining tools [67–70,85]. This portion of the research is currently limited to a supervised back propagation network with one hidden layer as suggested in previous literature [23,28,29,31,68,85]. This research also used a “brute force” method in which each neural network was made of 189 architectures with 100 repetitions each in order to avoid the challenges of ANN architecture design while addressing the low training size hurdle. Therefore, every product that was analyzed resulted in 18,900 individual time estimates. Further research can be conducted to improve the neural network design in terms of neural network type, the number of neurons and hidden layers required, and the number or repetitions needed. Since the neural network returns multiple time estimates, based on the network design, work is also needed on how to aggregate the data to arrive at a single point estimate. Nonetheless, a large opportunity exists in further improving the artificial neural network design and training in predicting an output, assembly time or otherwise.

8.3.4 Clearance Verification

The IDM was limited in finding connections between parts that were in physical contact with one another or with faces or edges that were coincident (see Chapter Three). The IDM did not have an option to find additional connections between parts that were within a designated distance. This is specifically important as products are often modeled with designed tolerances in mind. The example discussed in Chapter Three presents the design of a shaft (pin) into a block with a hole in the center. A designer creating the shaft and block assembly may model the shaft with a diameter of 1.000 inch and model the hole in the block with a diameter of 1.002 inch (see Figure 8.1).

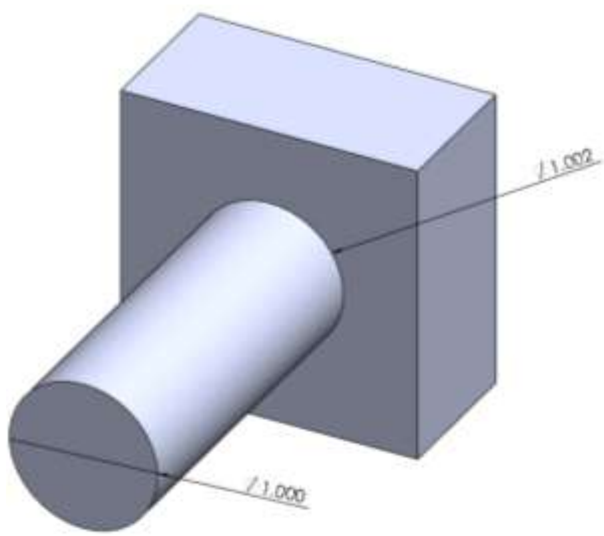


Figure 8.1: Shaft and Hole Modeled with a Tolerance

The IDM would not detect a connection between the shaft and the hole due to the 0.002 inch size difference. Similar to the interference detection tool used in the IDM, SW also includes a 'clearance verification' tool. The clearance verification tool is used to verify if all the parts in an assembly model are correctly designed so that there is an acceptable clearance between parts. The user of the clearance verification tool inputs the desired clearance (i.e. 0.001 inch), and SW finds all pairs of parts that are within that distance from one another. The clearance verification tool can be used to in place of the interference detection tool to find the connections between parts in the assembly, while adding the flexibility of finding parts that are within a user specified distance of one another. The performance of the clearance verification tool needs to be compared to the interference detection tool with regards to time for analysis and ability to detection connections between parts such as face to face and vertex to vertex.

8.3.5 Graph Modeling Refinement

The IDM uses the interference detection tool, which is a function, built into SolidWorks. One output of the interference detection tool that is not currently used is a volume overlap between parts. The volume overlap between parts could potentially be useful information in the connectivity graphs of the assembly [86]. The volume overlap could additional insight in the interconnectedness of the assembly that is not currently captured.

8.3.6 Complexity Vector Metrics

The twenty nine complexity metrics that form the current complexity vector are all used to represent the assembly model and used as a surrogate for assembly time. This dissertation presented the results of statistical analysis in reducing the current set of complexity metrics, however did not explore the need of possible additional metrics. Furthermore the current metrics are developed to represent an assembly for assembly time estimation, additional complexity metrics can be developed to represent other areas of interest: product cost prediction, manufacturing processes, time to products, and design time needed. Additional research is needed in the justification of the twenty nine complexity metrics and if these are sufficient to fully represent an assembly model and used as a surrogate for assembly time estimation.

8.3.7 Automated Assembly Instruction

Assembly instructions are authored by assembly planners and are manually created after a product has entered detailed design and production phase. Recent research has strived to standardize the work instruction authorship to a predefined list of verbs and nouns to assist in automating the authorship process [87]. The connectivity graphs found by the IDM could potentially be used to predict the assembly verbs. If sub-graph patters can be found between part connectivity and assembly verbs, an opportunity exists to automate the work instructions based on the assembly model.

8.4 Research Contribution

This research dissertation has developed, presented, and demonstrated a new graph generation (Interference Detection) method that can be used with SolidWorks to generate the connectivity graphs needed to apply the Complexity Connectivity Time Estimation method. The algorithm is based on standard solid modeling operations and is therefore extendable to other commercial applications. The IDM has three major benefits relative to the previous Assembly Mate Method (AMM).

The first major contribution is the elimination of variability due to designer decision when creating the assembly model. The AMM operated on the assembly mates that the designer chose to use in assembling the parts in the SW assembly file. While preliminary studies with the AMM demonstrated that this was a minor issue in prediction accuracy, it was still a variance between designers [58]. The IDM finds parts that are coincident or overlapping in the assembly space to create the connectivity graph eliminating the variability that is possible due to various designers.

The second major contribution of the IDM is the support for multiple file types. The IDM can operate on the bodies imported by SW from a number of different file types. The AMM is limited to SW assembly files because it requires the mate list which is specific to the SW software. With increasing globalization in industry, organizations across the design chain are using different software and modeling environments [65,88,89]. This contribution allows the separate organizations to share the geometry modeled in different environments without the need for assembly constraints.

The third benefit is a reduction of variance in the assembly time estimate while maintaining relatively the same accuracy as the AMM method. This research is another step in designing a fully automated assembly time estimate to provide design engineers with an accurate and repeatable tool that does not require substantial time or effort to implement. The estimated assembly times predicted by the IDM are similar to the AMM, have a lower variance, increasing the confidence of the actual assembly time falling in a range.

The demand for design tools to support the conceptual design stage has increased in industry due to the significant portion of product cost determined early in the design process [50,90]. This research has demonstrated the application of the IDM to low fidelity conceptual models generated early in the design process. The assembly time of a product is unknown and difficult to estimate during the conceptual design phase, but the IDM can provide assembly planners an estimated assembly time based on low fidelity models for early assembly process design.

The testing of the SIDM did not demonstrate an improvement over the IDM in estimating assembly time based on the percent error, but the sample size of products was limited. The handling portion of the SIDM can however be used to calculate the estimated handling code and time of the Boothroyd and Dewhurst assembly time estimation method. The use of the handling portion of the SIDM to calculate the handling code and time reduces the time and effort needed in applying the Boothroyd and Dewhurst assembly time estimation method.

8.5 Conclusion

This dissertation presents an automated assembly time estimation method based on the Boothroyd and Dewhurst assembly time estimation method and the complexity connectivity method. The IDM is developed and demonstrated for creating the connectivity graphs needed to calculate the complexity vector as input into the

Complexity Connectivity Method. The IDM is tested on products that are reverse engineered to determine a target assembly time, and product models provided by an industry sponsor with actual assembly times. The SIDM presents the separation of the handling and insertion time to address the subjective questions inherent in the Boothroyd and Dewhurst assembly time estimation method. The outcome of this research is an automated assembly time prediction tool that can be implemented throughout the design process.

REFERENCES

[1] Edwards K., 2002, “Towards More Strategic Product Design for Manufacture and Assembly: Priorities for Concurrent Engineering,” Materials & Design, **23**(7), pp. 651–656.

[2] Boothroyd G., Knight W., and Khan Z., 2008, “Design for Assembly,” Assembly Automation, **28**(3), pp. 200–206.

[3] Huang G. Q., 1998, “A survey report on design for manufacture in the UK furniture manufacturing industry,” Assembly, pp. 383–387.

[4] Boothroyd G., Dewhurst P., and Knight W. A., 2011, Product Design for Manufacture and Assembly, CRC Press, Boca Raton.

[5] Barnes C., Dalgleish G., and Jared G., 1997, “Assembly sequence structures in design for assembly,” Assembly and Task, (August), pp. 164–169.

[6] Daabub A., 1999, “A Computer-Based Intelligent System for Design for Assembly,” Computers & Industrial Engineering, **37**(1-2), pp. 111–115.

[7] Pandit A., and Siddique Z., 2004, “A tool to integrate design for assembly during product platform design,” ASME.

[8] Tavakoli M. S., Mariappan J., and Huang J., 2003, “Design for Assembly Versus Design for Disassembly: A Comparison of Guidelines,” ASME.

[9] Maynard H., Stegemerten G. J., and Schwab J. L., 1948, Methods Time Measurement, McGraw-Hill, New York.

[10] Zha X. F., Lim S. Y. E. Y. E., and Fok S. C. C., 1998, “Integrated intelligent design and assembly planning: a survey,” The International Journal of Advanced Manufacturing Technology, **14**(9), pp. 664–685.

[11] Strategy P., Improvements P., and Cheldelin B., 2000, “DETC2000 / DFM-14020 ASSEMBLY QUALITY METHOD : A TOOL IN AID OF,” Design Engineering, pp. 1–9.

[12] Gupta R., 1997, “Prototyping and Design for Assembly Analysis Using Multimodal Virtual Environments,” Computer-Aided Design, **29**(8), pp. 585–597.

[13] Huang G., 1999, “Design for manufacture and assembly on the Internet,” Computers in Industry, **38**(1), pp. 17–30.

[14] Boothroyd G., and Alting L., 1992, “Design for Assembly and Disassembly,” CIRP Annals - Manufacturing Technology, **41**(2), pp. 625–636.

[15] Miles B., 1989, “Design for Assembly-A Key Element Within Design for Manufacture,” ARCHIVE: Proceedings of the Institution of Mechanical Engineers, Part D: Transport Engineering 1984-1988 (vols 198-202), **203**(14), pp. 29–38.

[16] Sik Oh J., O’Grady P., and Young R. E., 1995, “A Constraint Network Approach to Design for Assembly,” IIE Transactions, **27**(1), pp. 72–80.

[17] Zha X. F., Lim S. Y. E., and Fok S. C., 1999, “Integrated Knowledge-Based Approach and System for Product Design for Assembly,” International Journal of Computer Integrated Manufacturing, **12**(3), pp. 211–237.

[18] Gupta R., “Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments .pdf,” Massachusetts Institute of Technology.

[19] Sirat M., Tap M., Shaharoun M., and others, 2000, “A Survey Report on Implementation of Design for Assembly (DFA) in Malaysian Manufacturing Industries,” Jurnal Mekanikal, **1**, pp. 60–75.

[20] Dean B. V., and Salstrom R. L., 1990, “Utilization of design for manufacturing (DFM) techniques,” Engineering Management Conference, 1990. Management Through the Year 2000-Gaining the Competitive Advantage, 1990 IEEE International, IEEE, pp. 223–232.

[21] Lakshminarayana R. E., and Takai S., 2007, “Effects of Non-Geometric Features and Incentive Schemes on Manual Assembly of System Variants: An Experimental Study,” ASME.

[22] Ohashi T., Iwata M., Arimoto S., and Miyakawa S., 2002, “Extended Assemblability Evaluation Method (AEM) - (Extended quantitative assembly producibility evaluation for assembled parts and products),” JSME INTERNATIONAL JOURNAL SERIES CMECHANICAL SYSTEMS MACHINE ELEMENTS AND MANUFACTURING, **45**(2), pp. 567–574.

[23] Mathieson J. L., Wallace B. A., and Summers J. D., 2010, “Assembly Time Modeling Through Connective Complexity Metrics,” International Conference on Manufacturing Automation (ICMA), Hong Kong.

[24] Karger D. W., and Bayha F. H., 1987, Engineered work measurement: the principles, techniques, and data of methods-time measurement background and foundations of work measurement and methods-time measurement, plus other related material, Industrial Press Inc.

[25] Tate S. S. J., Jared G. E. M., Brown N. N. J., and Swift K. G. K. G., 2000, "An Introduction to the Designers' Sandpit," Structure, DETC/DFM-14014. Proceedings of the Fifth ASME Design for Manufacturing Conference, Baltimore, Maryland, pp. 1–7.

[26] Barnes C. J., Dalgleish G. F., Jared G. E. M., Swift K. G., and Tate S. J., "Assembly sequence structures in design for assembly," Proceedings of the 1997 IEEE International Symposium on Assembly and Task Planning (ISATP'97) - Towards Flexible and Agile Assembly and Manufacturing -, pp. 164–169.

[27] Owensby E., Shanthakumar A., Rayate V., Namouz E. Z., Summers J. D. D., and Owensby J. E., "Evaluation and Comparison of Two Design for Assembly Methods: Subjectivity of Information Inputs," ASME Internation Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Washington, DC, pp. 1–11.

[28] Mathieson J. L., and Summers J. D., 2010, "Complexity Metrics for Directional Node-Link System Representations: Theory and Applications," Proceedings of the ASME IDETC/CIE 2010.

[29] Miller M. G. G., 2011, "Product and Process Based Assembly Time Estimation in Engineering Design," Clemson University.

[30] Owensby J. E., Namouz E. Z., Shanthakumar A., Summers J. D., and Owensby E., 2012, "Representation: Extracting Mate Complexity from Assembly Models to Automatically Predict Assembly Times," International Design Engineering Technical Conferences and Computers and information in Engineering Conference, Chicago, pp. 1–10.

[31] Miller M., Mathieson J., Summers J. D., and Mocko G. M., 2012, "Representation: Structural Complexity of Assemblies to Create Neural Network Based Assembly Time Estimation Models," International Design Engineering Technical Conferences and Computers and information in Engineering Conference, Chicago, IL.

[32] Nair V. N., Abraham B., Mackay J., Nelder J. A., Box G., Phadke M. S., Kacker R. N., Sacks J., Welch W. J., Lorenzen T. J., Shoemaker A. C., Tsui K. L., Lucas J. M., Taguchi S., Myers R. H., Vining G., Wu C. F. J., and Hill M., 2011, "Taguchi ' s Parameter Design : Panel Discussion," Quality.

[33] Mital A. A., Desai A., and Subramanian A., 2008, Product Development: A Structured Approach to Consumer Product Development, Design, and Manufacture, Elsevier, Boston.

[34] Renu R., Peterson M., Mocko G. M., and Summers J. D., 2013, “Automated Navigation of Method Time Measurement (MTM) Tables for Automotive Assembly Line Planning,” ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, Portland, OR, pp. DETC2013–13325.

[35] Peterson M., Mocko G., and Summers J. D., 2012, “Evaluating and Comparing Functional and Geometric Complexity of Product”, ASME,” International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, pp. DETC2012–71408.

[36] Miller M., Griese D., Summers J. D., Peterson M., and Mocko G. M., 2012, “Representation: Installation Process Step Instructions as an Automated Assembly Time Estimation Tool,” International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, pp. DETC2012–70109.

[37] Whitney D. E., and Knovel, 2004, Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development, Oxford University Press, New York.

[38] Jakiela M., Papalambros P., and Ulsoy a. G., 1985, “Programming Optimal Suggestions in the Design Concept Phase: Application to the Boothroyd Assembly Charts,” Journal of Mechanisms Transmissions and Automation in Design, **107**(2), p. 285.

[39] Elmaraghy W., Elmaraghy H., Tomiyama T., and Monostori L., 2012, “Manufacturing Technology Complexity in engineering design and manufacturing,” CIRP Annals - Manufacturing Technology, **61**(2), pp. 793–814.

[40] Do S., 2008, “CIRP Annals - Manufacturing Technology Complexity-based modeling of reconfigurable collaborations in production industry,” **57**, pp. 445–450.

[41] Hu S. J., Zhu X., Wang H., and Koren Y., 2008, “Product variety and manufacturing complexity in assembly systems and supply chains,” CIRP Annals - Manufacturing Technology, **57**(1), pp. 45–48.

[42] Spicer P., and Arbor A., “Design Principles for Machining System Configurations.”

[43] ElMaraghy W. H., and Urbanic R. J., 2004, “Assessment of Manufacturing Operational Complexity,” CIRP Annals - Manufacturing Technology, **53**(1), pp. 401–406.

[44] Schuh G., 2004, “Approach to Control Rising System-Complexity 1,” CIRP Annals - Manufacturing Technology, **53**(1), pp. 167–170.

[45] ElMaraghy W. H., and Urbanic R. J., 2003, “Modelling of Manufacturing Systems Complexity,” CIRP Annals - Manufacturing Technology, **52**(1), pp. 363–366.

[46] Eimaraghy H. A., Intelligent U., and Systems M., “Manufacturing Systems Configuration Complexity I ’ h Tooling,” (l).

[47] Suh N. P., 2005, “Complexity in Engineering,” CIRP Annals - Manufacturing Technology, **54**(2), pp. 46–63.

[48] Colwell B., 2005, “Complexity in Design,” Computer-IEEE Computer Society, **10**(38.10).

[49] Poli C., 2001, Design for Manufacturing, Butterworth Heinemann, Boston.

[50] Pahl G., Beitz W., Wallace K., Lucienne Blessing, Feldhusen J., and Grote K. H., 2007, Engineering Design: A Systematic Approach, Springer, London.

[51] Jenab K., and Liu D., 2010, “A graph-based model for manufacturing complexity,” International Journal of Production Research, **48**(11), pp. 3383–3392.

[52] Ameri F., Summers J. D., Mocko G. M., and Porter M., 2008, “Engineering design complexity: an investigation of methods and measures,” Research in Engineering Design, **19**(2-3), pp. 161–179.

[53] Summers J. D., and Shah J. J., 2010, “Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability,” Journal of Mechanical Design, **132**(2), p. 021004.

[54] Sambhoos K., Koc B., and Nagi R., 2009, “Extracting Assembly Mating Graphs for Assembly Variant Design,” Journal of Computing and Information Science in Engineering, **9**(3), p. 034501.

[55] Locascio A., 2000, “Manufacturing cost modeling for product design,” International Journal of Flexible Manufacturing Systems, **12**(2), pp. 207–217.

[56] Lindemann U., Maurer M., and Braun T., 2009, Structural Complexity Management: An Approach for the Field of Product Design, Springer, Berlin.

[57] Mathieson J. L., Wallace B. A., and Summers J. D., 2012, “Assembly time modelling through connective complexity metrics,” International Journal of Computer Integrated Manufacturing, **on-line**, pp. 1–13.

[58] Owensby J. E., 2012, “Automated Assembly Time Prediction Tool Using Predefined Mates from CAD Assemblies,” Clemson University.

[59] Veisz D., Namouz E. Z., Joshi S., and Summers J. D., “The Impact of the Disappearance of Sketching: A Case Study,” pp. 1–48.

[60] Namouz E. Z., Summers J. D. J. D., and Mocko G. M. G. M., 2012, “Reasoning : Source of Variability in the Boothroyd and Dewhurst Assembly Time Estimation Method,” International Design Engineering Technical Conferences and Computers and information in Engineering Conference, Chicago, IL, pp. 1–7.

[61] Namouz E. Z., and Summers J. D., 2013, “Comparison of Graph Generation Methods for Structural Complexity Based Assembly Time Estimation,” ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, Oregon.

[62] Caldwell B., Namouz E., Richardson J., Sen C., Rotenburg T., Summers J., Mocko G., and Obieglo A., 2010, “Automotive Lightweight Engineering: A Method for Identifying Lazy Parts,” International Journal of Vehicle Design2.

[63] Namouz E. Z., and Summers J. D., 2013, “Complexity Connectivity Metrics-Predicting Assembly Times with Abstract Assembly Models,” Smart Product Engineering, M. Abramovici, and R. Stark, eds., Springer Berlin Heidelberg, Bochum, Germany, pp. 77–786.

[64] Michaels J. V., and Wood W. P., 1989, Design to Cost, Wiley-Interscience.

[65] Kalpakjian S., and Schmid S. R., 2008, Manufacturing Processes for Engineering Materials, Pearson Education.

[66] Mathieson J., Wallace B., and Summers J. D., “Estimating Assembly Time with Connective Complexity Metric Based Surrogate Models,” International Journal of Computer Integrated Manufacturing, **on-line**(in press).

[67] Francis L., 2001, “Neural Networks Demystified,” Casualty Actuarial Society Forum, pp. 253–320.

[68] Kusiak A., 2000, Computational Intelligence in Design and Manufacturing, John Wiley & Sons, INC., New York.

[69] Blackard J. a., and Dean D. J., 1999, “Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables,” Computers and Electronics in Agriculture, **24**(3), pp. 131–151.

[70] Sethi I. K., 1990, “Entropy nets: from decision trees to neural networks,” Proceedings of the IEEE, **78**(10), pp. 1605–1613.

[71] Lawrence S., Giles C. L., and Tsoi A. C., 1997, “Lessons in Neural Network Training : Overfitting May be Harder than Expected,” pp. 540–545.

[72] Wetmore III W. R., Summers J. D., and Greenstein J. S., 2010, “Experimental study of influence of group familiarity and information sharing on design review effectiveness,” Journal of Engineering Design, **21**(1), pp. 111–126.

[73] Ostergaard K. J., Wetmore W. R., Divekar A., Vitali H., and Summers J. D., 2005, “An experimental methodology for investigating communication in collaborative design review meetings,” CoDesign, **1**(3), pp. 169–185.

[74] Corney J., and Lim T., 2002, 3D Modeling with ACIS, Saxe-Coburg Publications.

[75] Tate S. J., and Jared G. E. ., 2003, “Recognising symmetry in solid models,” Computer-Aided Design, **35**(7), pp. 673–692.

[76] Soler C., Holzschuch N., and Cois F., 2006, “Accurate Detection of Symmetries in 3D Shapes,” **25**(2), pp. 439–464.

[77] Mitra N. J., Guibas L. J., Pauly M., and Eth Z., 2006, “Partial and Approximate Symmetry Detection for 3D Geometry,” Computational Geometry, pp. 560–568.

[78] Ong N., Chua C., and Lee E., 1997, “Geometric analysis of parts from a 3-D solid model for manual assembly times,” Integrated Manufacturing Systems, pp. 137–146.

[79] Walpole R. E., Myers R. H., Myers S. L., and Ye K., 2012, Probability & Statistics for Engineers and Scientists, Prentice Hall, Boston.

[80] Montgomery D. C., and Runger G. C., 2003, Applied Statistics and Probability for Engineers, John Wiley & Sons, Inc.

[81] Montgomery D. C., 1997, Introduction to Statistical Quality Control, Wiley, New York.

[82] Dalgleish G. F., Jared G. E. M., and Swift K. G., 2010, “Journal of Engineering Design for assembly : Influencing the design process,” (January 2013), pp. 37–41.

[83] Lit P. De, Delchambre A., and Henrioud J.-M., 2003, “An integrated approach for product family and assembly system design,” Robotics and Automation, IEEE Transactions on, **19**(2), pp. 324–334.

[84] Thoe S., and Summers J. D., 2013, “CORRELATING PROBLEM / PROCESS EXAM QUESTION COMPLEXITY TO ANTICIPATED EFFORT : A MODELING PROTOCOL,” ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, OR, pp. 1–9.

[85] Engelbrecht A. P., Computational Intelligence An Introduction, John Wiley & Sons, Ltd.

[86] Shanthakumar A., and Summers J. D., 2013, “Design Enabler to Recognize Duplicate Geometries in CAD Assemblies,” **10**, pp. 1–16.

[87] Peterson M., Mocko G., and Summers J., 2013, “Knowledge Management for Semi-Automated Assembly Instruction Authorship and Translation,” ASME International Design Engineering Technical Conferences.

[88] H. Todd R., P. Magleby S., and R. Parkinson A., 2005, “Experiences and observations in introducing students to design and manufacturing globalization,” Journal of Manufacturing Systems, **24**(3), pp. 162–170.

[89] Lee J., “Manufacturing globalization: new challenges for the computer-integrated manufacturing,” Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology, pp. 206–207.

[90] Improving Engineering Design: Designing for Competitive Advantage., Washington, DC.