University of Cincinnati

Date: 6/19/2013

I, Stephanie Berndt , hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled:

Manufacturing Analysis and Process Optimization of Welded Parts

Student's name: **Stephanie Berndt**

This work and its defense approved by:

Committee chair: Sundararaman Anand, Ph.D.

Committee member: Sundaram Murali Meenakshi, Ph.D.

Committee member: David Thompson, Ph.D.

UNIVERSITY OF Cincinnati

6694

Manufacturing Analysis and Process Optimization of Welded Parts

A Thesis submitted to the
Graduate School of the University of Cincinnati
In partial fulfillment of the requirements for the Degree of

MASTER OF SCIENCE

In the Department of Mechanical Engineering
of the College of Engineering and Applied Science

by

Stephanie Berndt

Bachelor of Engineering and Applied Science
Miami University, Oxford, Ohio, 2008

Committee Chair: Dr Sam Anand

ABSTRACT

This research focuses on improving the manufacturability and optimizing and estimating process time for plate cutting and welding processes. To this end, three distinct tools have been developed in this work: (1) Design for Manufacturability Anlaysis, (2) Optimal machine (cutting) time and (3) Accurate labor time estimation of welding process. This research extracts features from an AutoCAD dxf and applies sheet metal design for manufacturability rules to ensure cost effective and efficient assembly and manufacturing starting at the design phase. During the next phase of the manufacturing process, the tool path planning involved in manufacturing during the laser cutting process is optimized. An accurate time estimate is calculated by adapting a traveling salesman problem and determining the shortest laser traversal route by using a genetic algorithm. Finally, a weld time estimation program was developed to accurately estimate the weld assembly time by performing a comparative analysis of historical data. This research contributes in developing the most efficient means of welded parts manufacturing while estimating process time.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the most challenging aspects of any business is setting prices that will make a profit. Mathematically this is a very simple equation, being profit equaling the difference between customer prices and manufacturing costs. There are simply two ways to increase profit, the first being to increase the customer price, the second being to decrease the manufacturing costs. Although the equation may be simple, the methodology behind the profit equation is more complex.

There is enormous research that involves decreasing the manufacturing costs. The entire field of lean manufacturing is dedicated to evaluating costs of various manufacturing processes versus the value that set process adds. Efficiency is key when trying to reduce manufacturing costs and it is crucial when trying to operate a profitable business.

1.1 Need for Research

Skilcraft Sheetmetal, Inc. specializes in custom metal fabrication for customers covering a broad range of industries. Products range from government secure filing cabinets to intricate aerospace pieces. Because of the unique variety of products manufactured, the cost of manufacturing varies greatly which directly influences the price of the product.

Skilcraft is continually looking at areas to improve their operations and have narrowed down their least productive departments to be the welding department and the laser cutting department. One area of particular weakness is the process in which the sales team sets customer prices for products and specifically the amount of labor time estimated in the laser cutting and welding department. Inaccuracies in the amount of time estimated for the assembly versus the time the weld assembly actually took have resulted in a severe loss of profit.

Being able to accurately estimate weld assembly times are essential to the entire company. The purchasing department needs to approximate the number of machines, tools, or equipment accessories to buy. Management needs to know how many people to employ, whether they will need to hire additional help and whether it should be a temporary or a permanent position. The scheduling department needs to accurately create a schedule to deliver products to the customers on time while creating an even work flow in order to maintain customer satisfaction. Skilcrafts current methods for estimating the time it will take to weld an assembly together are based on expert judgement and reasoning.

Along with looking at ways to improve the welded assembly times, the time estimation techniques used in estimating the time to laser cut pieces is also a weak area of productivity. In addition, the laser cutting process has not been optimized for minimal machining time. Current methods involve a person reading the drawing dimensions and entering these dimensions into an excel spreadsheet along with some user inputs based on the material and geometry of the piece. The spreadsheet then calculates the estimated time based on external and internal laser traversal. This is a very time consuming process and yields results that are inaccurate often due to the demand for a quick turn around and innate human error.

Creating company wide standards for estimating welded assembly times and laser cut times are paramount in developing good business practices. This provides transparency to weaknesses in the manufacturing process whether it be worker related or process related and will result in overall increase in productivity and has the potential to lower manufacturing costs.

1.2 Outline of Thesis

This thesis paper is divided into five chapters. The first chapter is an introduction into the company and the problem with estimating welded assemblies and laser cutting times. The importance of the research and the necessity for accurate results will be addressed in this chapter as well. Chapter 2 give a historical perspective to prior research and methods that are implemented in this research and will lay the foundation for methodology used in this thesis. Design for manufacturing approach will be discussed along with guidelines for the laser cut parts, methods used in tool prediction, and techniques often used in cost prediction. Chapter 3 dives more deeply into the determined methods developed in this thesis. Algorithms implemented in the research are explained along with the methods developed. This chapter also discusses the pre-processing approaches of raw data as well as post-processing and model validation techniques. In Chapter 4 results are given for the methods used in estimation and detailed instructions on using the two programs developed. In the fifth and final chapter, recommendations and conclusions are drawn in a closing response to the research and results found possible the future work that may stem from the work presented in the paper is also included in this chapter.

Chapter 2

Literature Review

Continuous efforts are being implemented to decrease the cost of manufactured goods throughout the world. There are two aspects of decreasing manufacturing costs that are researched in this paper. The first being making improvements to the design of the product in order to in order to make manufacturing easier on machines and operators and consequently decreasing manufacturing costs. This idea is called Design for Manufacturability (DFM) and is a large area of research for engineers and industries. "Many analyses of product costs in the development cycle indicate that about 70% of the total product cost is already committed by the time the design is completed ..." [17].

The second method of reducing manufacturing costs that was researched in this thesis is related to tool path planning. Being able to accurately and most efficiently manufacture a products is crucial in decreasing the cost of a manufactured product. Specifically, this research investigates laser traversal for cutting sheet metal pieces and finding the shortest path when cutting a sheet metal part.

Finally, being able to accurately predict the effect of these approaches in cost reduction is necessary to establish accurate pricing. By knowing how long a process is expected to take, results in determining accurate labor costs that can lead to fair pricing of the product for the customer. This process outline is seen in Figure 2.1.

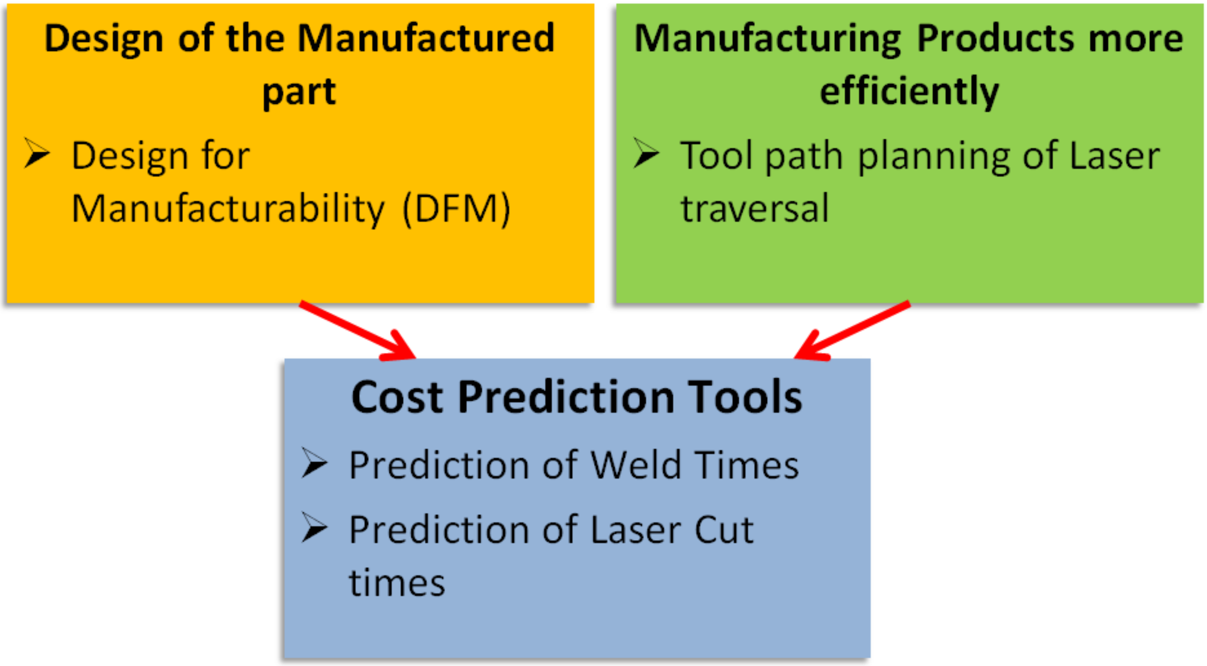


Figure 2.1: Methods of Decreasing Manufacturing Costs

2.1 Applying Design for Manufacturability (DFM) Rules

Design for manufacturability or DFM is the idea that certain features and characteristics of the design of a part make the part easier to manufacture which may result in decreasing manufacturing costs and increasing in machine and tool life. Guidelines for DFM are given for various manufacturing processes including forming, machining, casting, assembly, and finishing. These guidelines were published by Roger W. Bolz in the 1940s [2], but had been used for several year in manufacturing facilities and even by the great innovators like Eli Whitney and Henry Ford[3]. Though these ideas are not new, they are very powerful in reducing labor time and production cost, and should be considered when looking at estimation of laser cutting times.

2.1.1 Rule 1: Minimum Internal and External Radii

“Small radii, especially internal radii, are difficult to cut [3]” therefore it is best to allow for a “Generous internal and external radii [3]”. “The Corner radius on sheet metal parts should be at least 0.7 times of sheet thickness [9]”. An example is shown in Figure 2.2. This not only aids in decreasing motor wear in the laser machine, it also increases cutting speed. The laser cutting head does not need to slow down to perform the necessary cut when a generous fillet is applied to the corners.

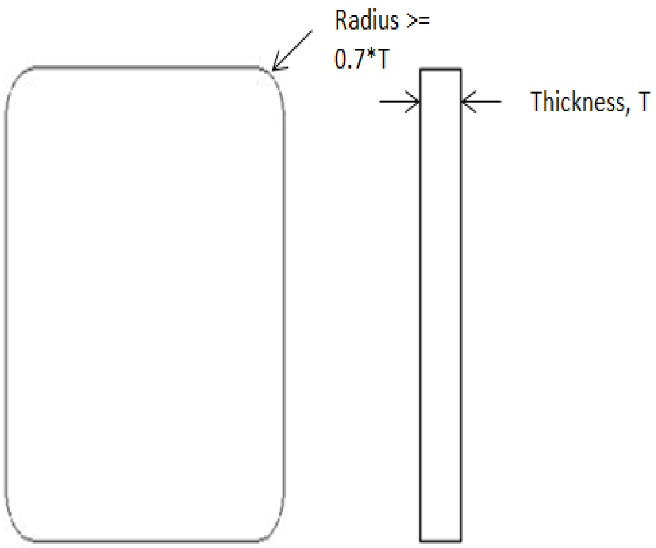


Figure 2.2: Example of Minimum Corner Radii (Adapted from [9])

2.1.2 Rule 2 & Rule 3: Minimum Slot Allowance

”Slots show allow for a width no longer than one and one half times the thickness. The length of the slot should not exceed five times the width.” [7] This is shown in Figure 2.3.

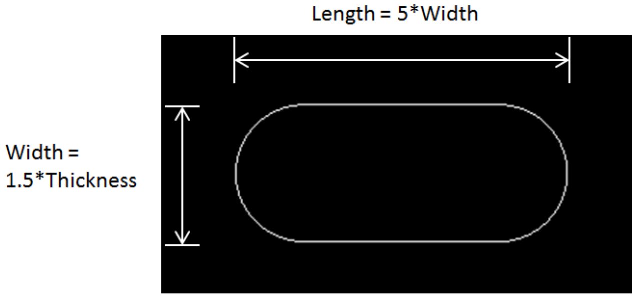


Figure 2.3: Minimum Slot Allowance (Adapted from [7])

2.1.3 Rule 4: Minimum Distance Between Inner Features and Edge

It is recommended that the distance, d, between the edge of the part and any inner features of the part be at least 1.5 to 2 times the sheet metal thickness, but not less that 0.03 in as shown in Figure 2.4. This is also a good recommendation for the distance between inner features. [3]

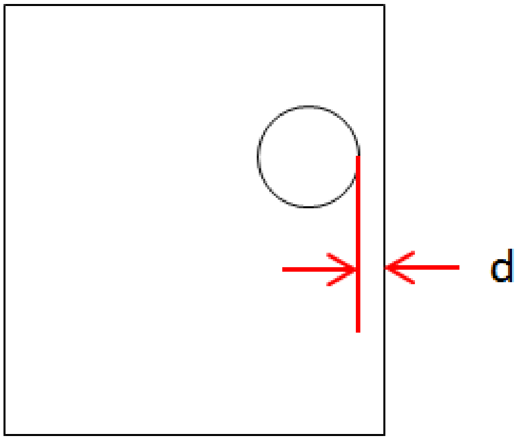


Figure 2.4: Minimum Spacing Between Inner Features and Edge of Piece (Adapted from [3])

2.1.4 Rule 5: Minimum Hole Size

"Holes can be produced by flame cutting, but because of the kerf, there is a minimum size for such opening" [3] as shown in Table 2.1.4. Smaller features will have a higher concentration of heat in that area which may cause deformation of the part. Allowing for a substantial hole diameter will alleviate this problem and will also ensure enough clearance of the laser lead in to cut the desired hole and slot.

Table 2.1: Minimum Hole Sizes (Adapted from [3])

Gauge	Minimum Slot Size (in)	Gauge	Minimum Slot Size (in)
3	0.2391	15	0.0673
4	0.242	16	0.0598
5	0.2092	17	0.0538
6	0.1943	18	0.0478
7	0.1793	19	0.0418
8	0.1644	20	0.0359
9	0.1495	21	0.0329
10	0.1345	22	0.0299
11	0.1196	23	0.0269
12	0.104	24	0.0239
13	0.0897	25	0.0209
14	0.0747	26	0.0179

2.1.5 Rule 6: Minimum Distance From Interior Features to Bend Line

"When a bend is made too close to a hole, the hole may become deformed. . . ." [16]. In order to avoid additional machining processes to maintain required hole tolerances, interior features should be at a distance of twice the material thickness plus the bend radius, as seen in Figure 2.5.

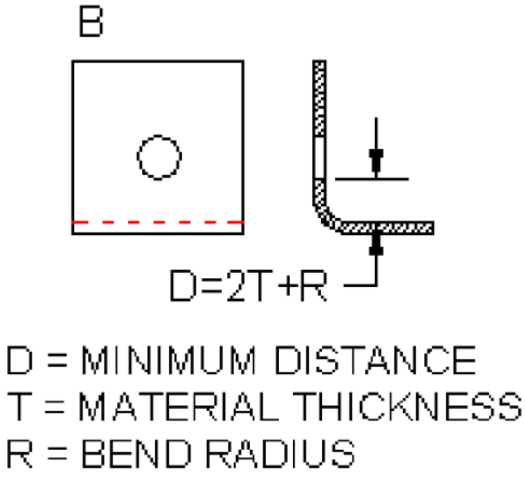


Figure 2.5: Minimum Distance from Bend line [16]

2.2 Laser Cutting Tool Path Planning

In order for the laser to cut the desired sheet metal pieces, a tool path must be generated and run in the CNC machine. The order in which these sheet metal piece features are cut is dependent on the logic of the user generating the code for the CNC. “An analysis of the total time spent by components in a manufacturing process reveals that an average of 70 % of the time is taken by movements of the jobs or of the tools” [21]. In order to find the shortest route possible, it is useful to look at the Travelling Salesman Problem (TSP) [4]. The TSP can be described as the shortest possible tour between cities such that every city is visited only once. First formulated in the 1930s, it is widely studied across varied scientific and mathematic fields. This optimization problem is much more complex than it appears at surface and only increases in complexity as more cities are added to the desired route.

The brute force approach is a commonly used method implemented in efforts to find an optimal route for the TSP problem. This method calculates the distances between all possible routes and selects the route with the shortest distance as the optimal route. Although this type of problem solving technique will result in a solution that is the global minimum if a solution exists, it is computationally intensive. Therefore this problem solving method is ideal for small problem sizes. When problem sizes get to be moderately large, the computing power necessary to generate an optimum result is just simply not feasible.

Other methods of solving the TSP are heuristic approaches and will not necessarily be able to determine a solution that is the global minimum, however they are much more computationally realistic in larger sized problems. Developed in the 1950s and applied to industrial applications in the 1980s, Genetic Algorithms (GA) have become a popular and powerful tool in all iterative cost function minimization or maximizing problems including the TSP [6]. GAs are considered as Artifical Intelligence based approaches and consists of an algorithm that mimics the biological process of evolution. Implementing a GA provides an approximation but will yield accurate results that are far less computationally intensive.

2.3 Predicting Manufacturing Cost and Establishing Standards

Improving manufacturing and assembly standards has been in the forefront of manufacturing facilities since Fredrick Taylor introduced the idea at the turn of the 19th century. Today, companies are constantly looking for ways to improve and accurately measure productivity, and are still using techniques based on Taylor's principles.

Taylor's ideas, which have been referred to as Scientific Management [5], opened the door for several renowned ideas including Henry Gantt's development of the Gantt Chart [10] and Frank Gilbreth's contribution to motion study [20]. Amalgamation of Taylors methods of time study and Gilbreths ideas of motion study became what is currently known as Methods Engineering,"a systematic examination of the methods of carrying on activities so as to improve the effective use of resources and to set up standards of performance for the activities being carried out" [11].

Although the ideas of Methods Engineering were developed over 80 years ago, it is still a widely used method to establish time and performance standards in a manufacturing environment. The methodology to create time standards are executed one of two ways. The first being performing a time study, where a person will observe a manufacturing process and record the time taken to complete the task. The second, and the one that is more relevant to this work, is reviewing historical data and developing time standards based on how long the process took in the past. With these two methods, a labor cost estimation of the product is calculated. Research performed by Niazi and Dai [14] suggests that cost estimation can categorized in two ways, qualitative and quantitative. Both are important in estimating process time and correlating with costs.

2.4 Qualitative Cost Estimation

Qualitative cost estimation is commonly used when there is a time constraint, or when just an estimate is required that can be refined further. Qualitative techniques rely exclusively on historical comparison to generation future predictions which, according to Niazi [14], is separated into intuitive techniques and analogical techniques. Intuitive techniques include methods such as Expert Judgement and Case-Based Reasoning. These techniques are utilized early in the design process when specific details of the assembly are not always known. This leads to a less accurate estimation due to the lack of information available.

2.4.1 Expert Judgement

Rush and Roy describe expert judgement as "…thinking and reasoning processes that experts use as they refer to historical data to make judgement" [18]. Benefits to using expert judgement are that it is relatively inexpensive and can produce a time estimate very quickly. Some obvious disadvantages are subjectivity and inconsistency. Customers also do not respond as well to estimates that are not mathematically derived and on occasion estimations will vary between two people. Although there are significant cons involved with using expert judgement, it is integral to the time estimating process.

2.4.2 Case-Based Reasoning

Using case-based reasoning to estimate the process time involves looking at parts or assemblies that have similar features and estimating a time based on historical completion times. One of the largest advantages to using case-based reasoning in estimating process times is that it would provide results based on the capabilities of the machinists, since they have accomplished similar tasks with such times in the past. One of the disadvantages is the lack of precision in the estimates as well as the necessity for large amounts of historical data needed for the comparison.

2.5 Quantitative Cost Estimation

Quantitative cost estimation techniques are based on specific details of the parts or assemblies and unlike qualitative methods, are more accurately used later in the design process. In the current context, quantitative cost estimation will use real user inputs regarding part or order features such as weight, volume, or gauge of the material in order to accurately estimate weld assembly time.

Using specific and related parameters to perform statistical analysis results in the determination of the weld assembly times as a function of the parameters. These quantitative parameters, can be used in a cost function that is capable of extrapolation for other parts or assemblies. The first step in developing a quantitative time estimating model is identifying key parameters that affect cost. This is a crucial step and can lead to inaccuracies if the incorrect parameters are chosen. Once the parameters are chosen, the cost function needs to be derived. This can be performed by using analytical techniques such as linear regression, non-linear regression or learning machines.

2.5.1 Regression Analysis

Regression models describe relationships between predetermined inputs related to the weld assembly and the estimated completion time. A benefit of using regression analysis is that it is fairly accurate when it is used with historically accurate data. However, regression tends to favor larger estimates leaving smaller estimates more inaccurate. There are methods of compensating for this phenomenon but it can be a tedious process.

2.5.2 Neural Network Model

One increasingly popular way to describe complex relationships between independent and dependent data is by using a neural network. Neural networks are trained by using historical data to learn how to respond to user inputs. Neural networks can produce very accurate results with data that will not respond well to regression analysis (i.e. nonlinear data, multi-collinear data). But in order to train and implement a neural network a large amount of historical data must be available. According to Zhang , “the neural network out performed regression analysis for almost all testing sets and assembly systems” [21].

2.6 Estimation Techniques Approaches Used in The Current Work

All of these techniques reviewed above have their advantages and disadvantages. Each of these are useful in different phases of the design process shown in Figure 2.6. Qualitative cost estimation techniques are best utilized in the early stages of the design process and are shown in green, while the quantitative techniques are shown in red and are employed later in the design process. If a cost estimation technique is used outside of its recommended design phase the technique will provide less accurate results.

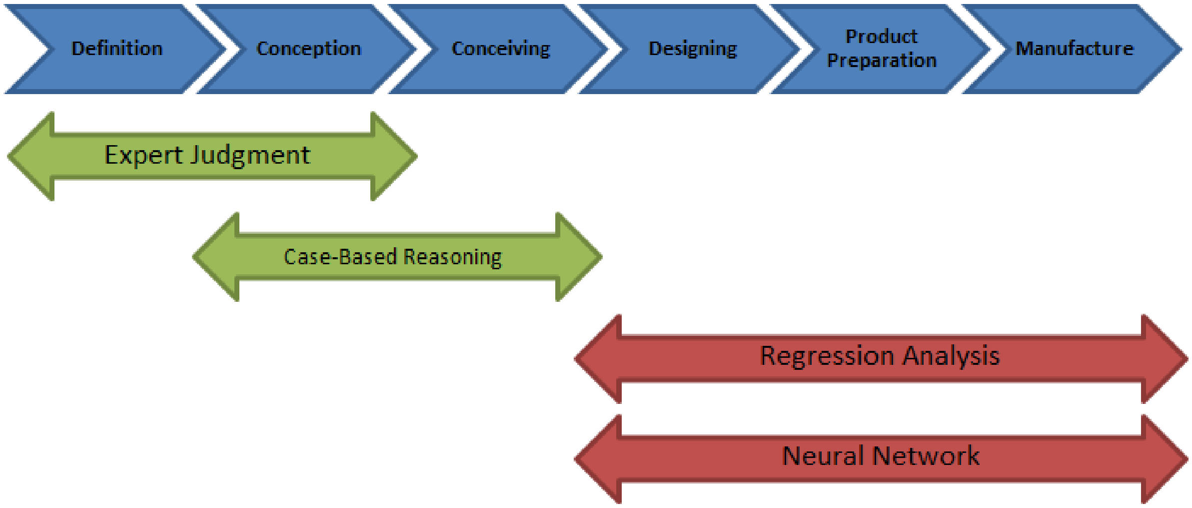


Figure 2.6: Use of Cost Estimation Techniques (Adapted from Mianaei [12])

Time and cost estimation has been an essential tool for over 100 years i the manufacturing industry. Choosing a cost estimation method depends on the information available. For early in the design process when specifics of assembly may be unknown it is best to use non-discrete techniques such as expert judgement or case-based reasoning. As the design phase progresses, more sophisticated approaches such as regression analysis or neural networks should be used to ensure more accuracy.

Chapter 3

Methodology and Results

Being cost conscious during all of the production stages of a product will have a significant effect in lowering the overall cost of manufacturing. The first instance of real cost reduction measures can occur during the design stage of the process. Based on the Design for Manufacturability Rules discussed in Chapter 2, an algorithm for extracting features and checking design violations was developed using MATLAB [19]. The MATLAB program extracts features from a AutoCAD Drawing Exchange Format or DXF and collects information regarding lines, polygons, circles and arcs that comprise the drawing. The layer that the feature is drawn on as well as location of these features is also extracted from the DXF file. Using this information a determination if any of the Design for Manufacturing (DFM) rules are being violated, can be ascertained.

Once a desired design has been chosen it must be cut via a laser CNC machine from raw sheet metal and finally welded together to create the desired assembly. It is imperative that both of these operations be accurately estimated prior to manufacturing in order to assure proper scheduling and shipping requirements.

3.1 Design for Manufacturing (DFM)

Design for manufacturing or DFM are guidelines that have been developed as a set a suggested guidelines to aid in manufacturing and assembly [17]. DFM guidelines for sheet metal fabrication suggest amending the design of features cut out of the sheet metal. The overall goal is to:

1. Decrease Wear on Machinery
2. Decrease Manufacturing time
3. Decrease Tolerance Errors
4. Maintain Design Integrity

Among the comprehensive list of sheet metal DFM rules complied by Hedge [7], the following rules are applied in this work:

Rule 1: Corners must have a fillet with radii at least 0.7 the material thickness

Rule 2: Slots must have a width of at least 1.5 times the material thickness

Rule 3: Slots must have a length of at least five times the slot width

Rule 4: The minimum distance from interior features to edge of part must be at least 1.5 times the material thickness

Rule 5: The minimum diameter of a hole must be at least twice the material thickness

Rule 6: The distance from a bend line to an interior features must be at least twice the material thickness plus the bend radius.

Detecting Rule 1 & Rule 2 violations: Rule 1 and Rule 2 both look at the relation of arcs to lines for detecting corners or slots. Since an arc can only connect to two lines, the location of those lines will determine whether the arc comprises a fillet radius or whether the arc is a part of a interior slot configuration. The algorithm used to identify if Rule 1 or Rule 2 has been violated is shown in Figure 3.1.

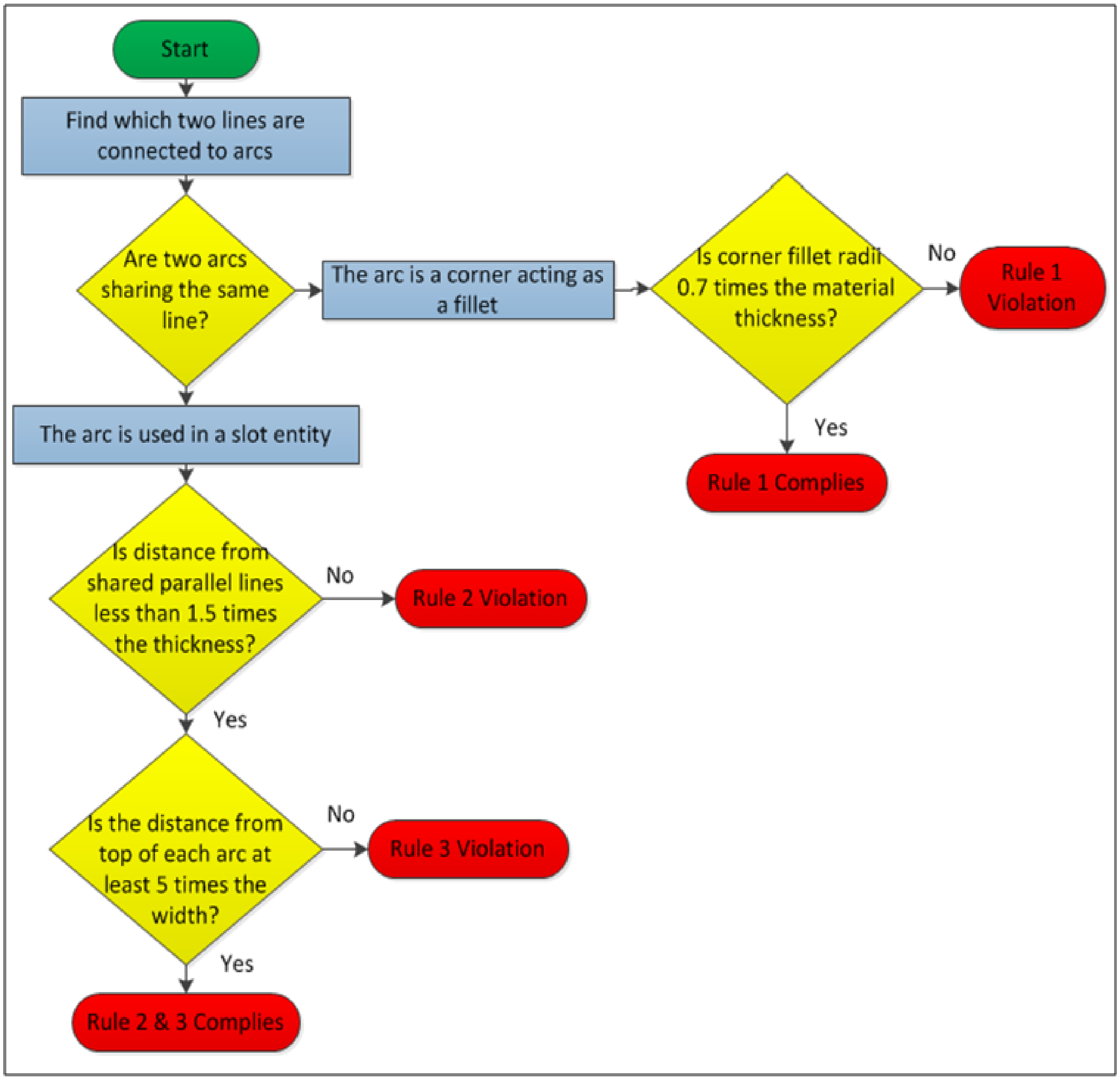


Figure 3.1: Algorithm for detecting a Rule 1 or Rule 2 violation

Detecting Rule 4 & Rule 5 violations: Rule 4 relates to the Euclidean distance from interior features to the edge of the exterior entities. In this same algorithm, violations of Rule 5 requiring a minimum diameter of circles related to the material thickness can also be checked. This algorithm is depicted in Figure 3.2.

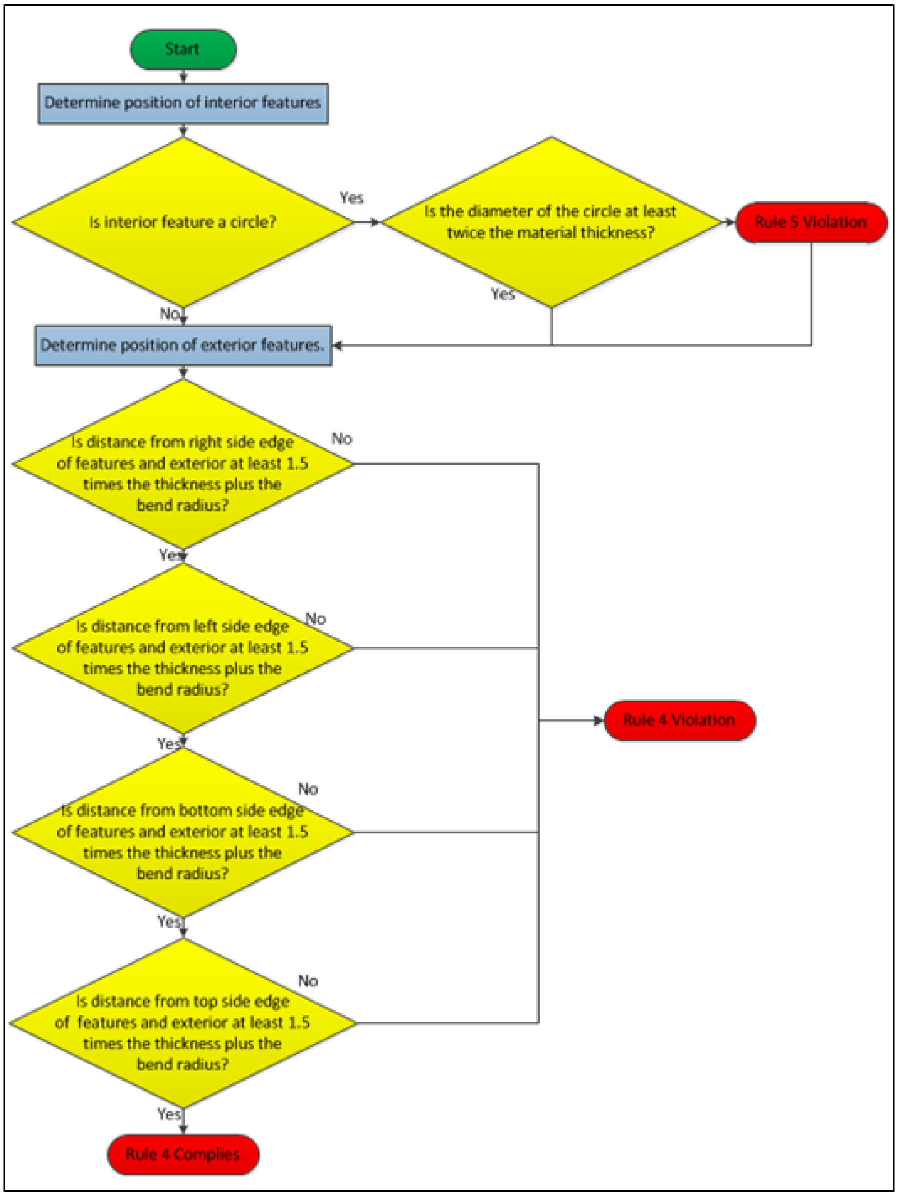


Figure 3.2: Algorithm for detecting Rule 4 and Rule 5 violations

Detecting Rule 6 violations: Rule 6 checks that interior features are not too close to any bend lines of the part. If features are too close, distortion may occur. This algorithm assumes that the bend radius is the recommended dimension based on the material thickness. Deviating from the recommended bend radius may result in fracturing of the metal. Bend lines are drawn on a different layer than other entities in the DXF file making them easier to identify. This algorithm is similar to the Rule 4 algorithm and is shown in Figure 3.3.

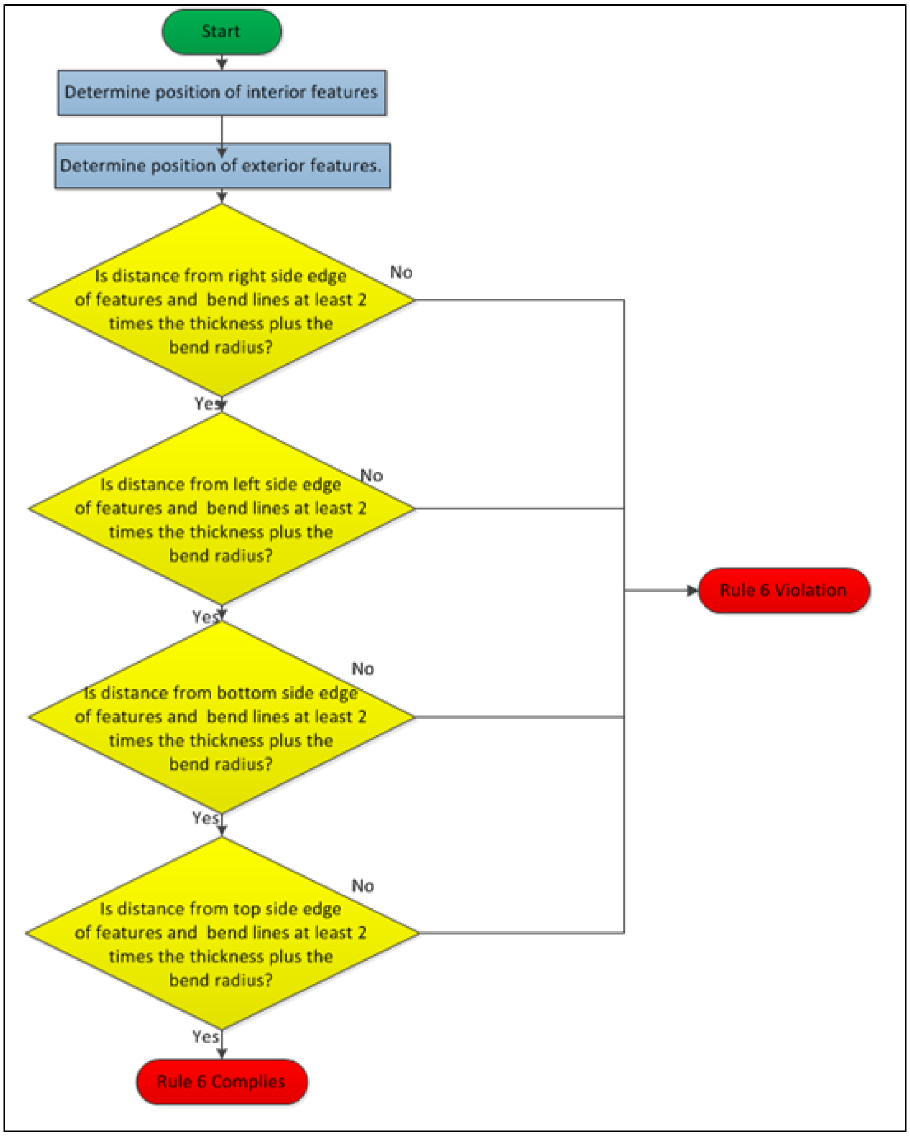
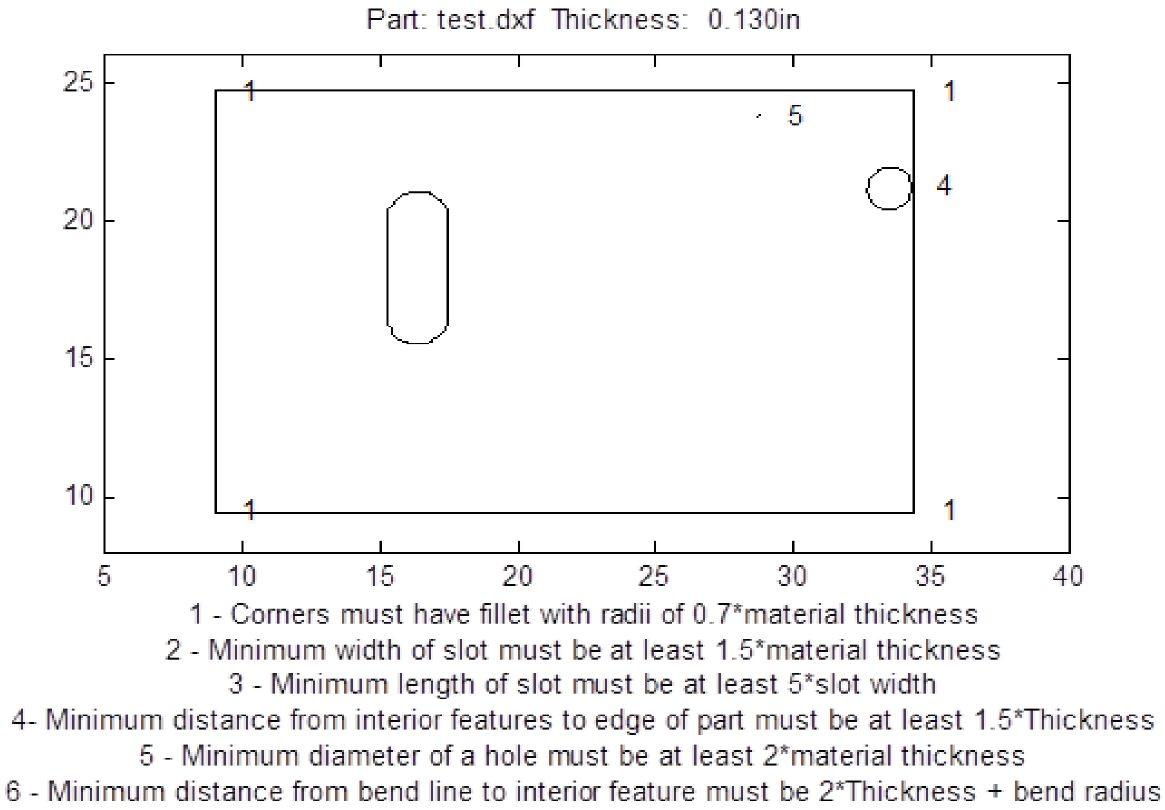


Figure 3.3: Algorithm for detecting Rule 6 violations

The algorithm calculates the distance of the edge of a feature, to the bend line and compares it to twice the material thickness.

3.2 DFM Analysis Results

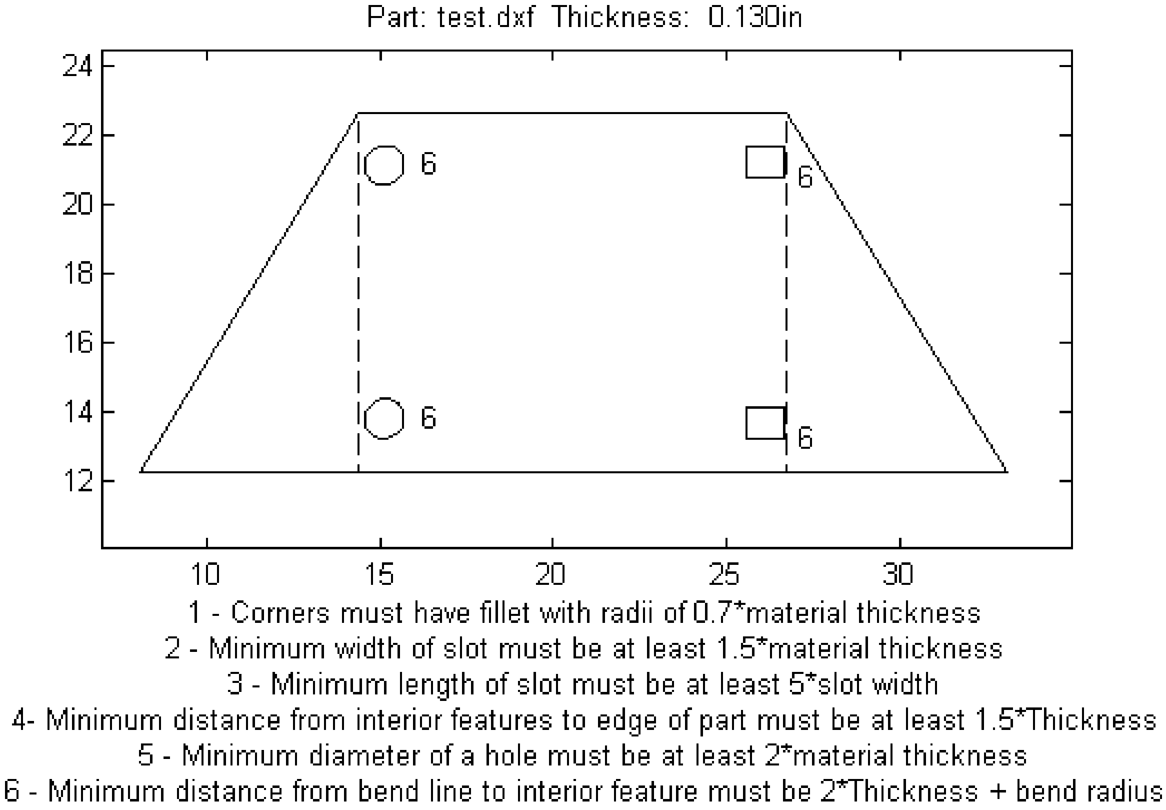
The purpose of the design for manufacturability study is to verify certain DFM rules are in compliance with design standards. An example of this application on an actual part (Part 1) is shown in Figure 3.4.



1 - Corners must have fillet with radii of 0.7*material thickness
2 - Minimum width of slot must be at least 1.5*material thickness
3 - Minimum length of slot must be at least 5*slot width
4- Minimum distance from interior features to edge of part must be at least 1.5*Thickness
5 - Minimum diameter of a hole must be at least 2*material thickness
6 - Minimum distance from bend line to interior feature must be 2*Thickness + bend radius

Figure 3.4: Example of DFM analysis output for Part 1

The MATLAB program plots a drawing of the part being analyzed and a numbering scheme indicating which rule has been violated is printed to the right of the feature in violation. An example of another part (Part 2) can be seen in Figure 3.5, where an interior entity is closer than the minimum distance from the bend line and has been called out.



1 - Corners must have fillet with radii of 0.7*material thickness
2 - Minimum width of slot must be at least 1.5*material thickness
3 - Minimum length of slot must be at least 5*slot width
4- Minimum distance from interior features to edge of part must be at least 1.5*Thickness
5 - Minimum diameter of a hole must be at least 2*material thickness
6 - Minimum distance from bend line to interior feature must be 2*Thickness + bend radius

Figure 3.5: Example of DFM analysis output for Part 2

3.3 Laser traversal Path Planning and Time Estimation

Once design modifications have been made and a machine design has been chosen it must be cut via laser CNC machine from raw sheet metal. A nesting alogorithm is used to program the laser cutter in order to cut the desired parts with minimal scrap. This program converts the path the laser traverses into G and M codes, a commonly used NC programming language. However, before the pieces are cut, it is important to know how long the process of cutting the pieces will take and also optimize the laser traversal path. As it is in many businesses, time is money and finding an accurate time estimate to the laser cutting process is crucial. Again, using the feature extraction program that was developed in MATLAB, a separate algorithm was developed in order to optimize and estimate the laser cutting times.

The laser cutting time is comprised of two distinct states of the laser, namely the time it takes the laser to cut the material $Time_{cut}$, and the time it takes the laser to traverse between cuts $Time_{trav}$ as shown in Equation 3.1.

$$Time_{total} = Time_{cut} + Time_{trav} \tag{3.1}$$

To start with, the location and the layer of all features comprising the part is extracted. These include all of the features that comprise the part, that the laser will have to travel in order to cut the part. Equation 3.2 is the equation used to estimate the time it will take the laser to cut all the features in a plate. This equation was developed by Skilcraft and is known to be fairly accurate at estimating laser cutting time.

$$Time_{cut} = \frac{Dist_{tot}}{s * 0.75} + \frac{P_{tot} * r * 1.1}{60} + \frac{L}{60} \tag{3.2}$$

In this equation, $Dist_{tot}$ is the total distance the laser travels to cut interior and exterior features of the part, s is the speed the laser travels (dependent on how size of contours of piece and gauge), P_{tot} is the total number of pierces (number of interior features plus 1), r is the pierce time (dependent on material gauge),and L is the load time. The MATLAB based program calculates the distance of all the lines and arcs, the circumference of the circles, and the perimeter of the polygons that are used in the drawing. These distances are summed and comprise the $Dist_{tot}$ variable in the $Time_{cut}$ equation. The total number of pierces is also extracted from the DXF by determining the number of interior features that are in the part. The rest of the variables that make up the $Dist_{tot}$ equation, s, r, and L, are user inputs defined when the program is initialized.

Once the $Time_{cut}$ has been calculated, the $Time_{trav}$ is determined using Equation 3.3. This is simply the travel distance between interior features over the machine speed being rated at 10182IPM.

$$Time_{trav} = \frac{Dist_{travtot}}{10182} \tag{3.3}$$

There are a few assumptions that are made when calculating the travel distance and they are as follows:

1. Interior features are cut prior to cutting the exterior perimeter. This is a common practice when laser cutting sheet metal. If the perimeter is cut before interior features, the part may inadvertently shift causing the interior feature cuts to be inaccurate.

2. The exterior perimeter cut is always started at the bottom left corner. This means the laser traversal path cutting the interior features must end at the bottom left corner

3. Starting and ending points for interior cuts is the closest position to the next feature to be cut.

The Traveling Salesman Problem (TSP) has been used in the research to calculate the shortest route $Dist_{travtot}$ between the interior features. The classic TSP aims to find the shortest possible route that visits each city exactly once, given a list of cities and the distances between each pair of cities. For the current problem, the interior features are analogous to cities and the distance between the feature is equivalent to the distance between cities. In its common linear optimization form the optimization model as shown in Equation 3.4 [13] is solved for the TSP, here c_{ij} represents the distance from city i to city j, and x_{ij} is a binary (yes/no) option of whether a route between a pair of cities was chosen or not in determining the route travelled with the minimum distance.

$$
\begin{aligned}
& \min \sum_{i \neq j} c_{ij} x_{ij} \qquad [1] \\
\text{subject to} \quad & 0 \leq x_{ij} \leq 1 \qquad \forall i, j \\
& x_{ij} \ integer \qquad \forall i, j \\
\sum_{i=0, i \neq j} & x_{ij} = 1 \qquad j = 0, \ldots, n \\
\sum_{j=0, j \neq i} & x_{ij} = 1 \qquad i = 0, \ldots, n \\
u_i - u_j + n x_{ij} & \leq -1 \qquad 1 \leq i \not+ j \leq \\
u_i & \leq 0 \qquad 0 \leq i \leq n
\end{aligned}
\tag{3.4}
$$

For problems with a smaller number of cities, a binary integer programming problem like this is very useful in determining an optimal minimum route. However, the complexity of the problem increases exponentially when additional cities are added.

In order to find the shortest possible route between interior features, a Genetic Algorithm (GA) was applied which follows the flow chart shown in Figure 3.6. The GA delivers the distance of the shortest possible route which was used as $Dist_{travtot}$.

The laser cutting estimation MATLAB program will extract data from all of the drawings that are located in the folder the program is saved it. It is necessary that parts with the same initial user inputs be run simultaneously. The limitations of this program is that the laser cutting estimator does not have capabilities to extract data about type and gauge of material.

An example of the GA for the TSP algorithm is shown in Figure 3.7. The interior features are numbered and random initial population is created in Table 3.1. to start with, the fitness function of the route with the minimum euclidean distance from the initial population is stored. From the initial population four routes are randomly chosen from the initial population as shown in Table 3.2.

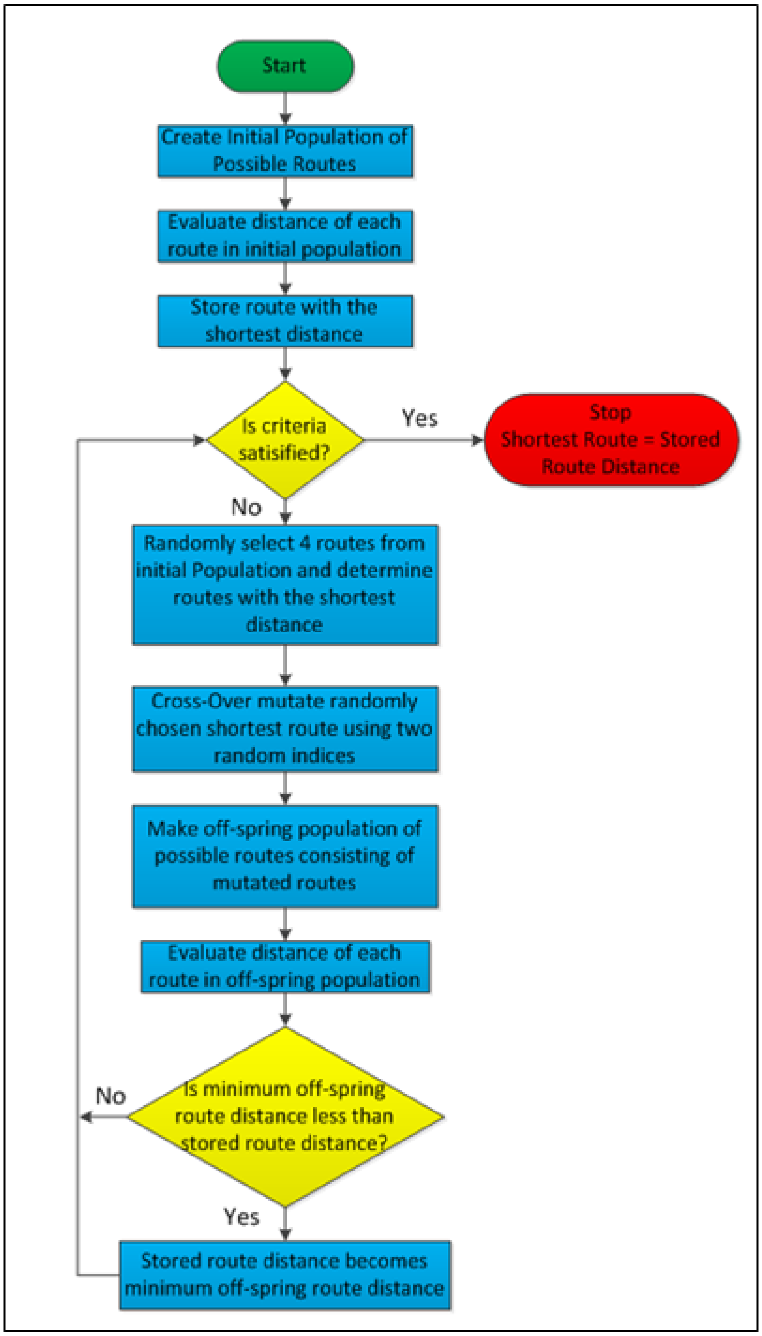


Figure 3.6: Traveling Salesman Genetic Algorithm (Adapted from Kirk [8])

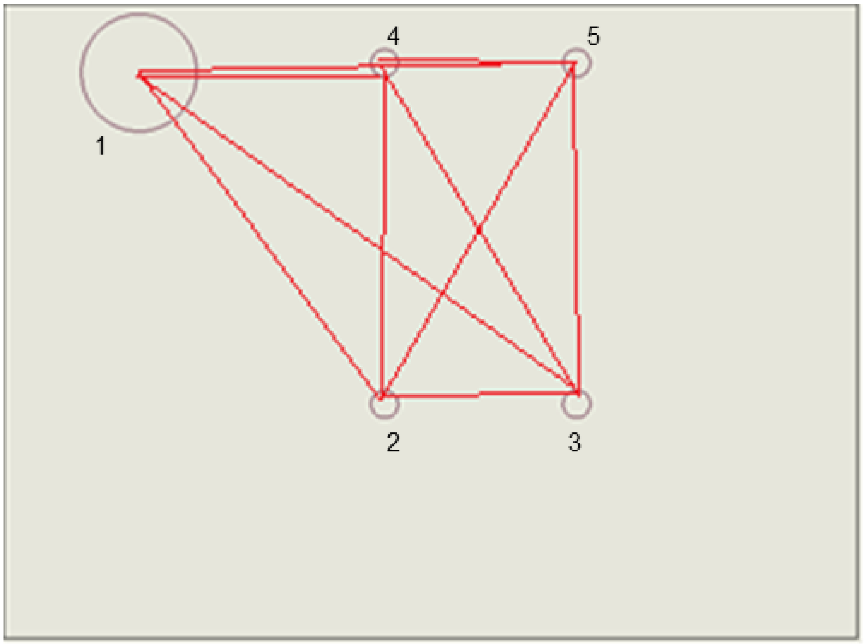


Figure 3.7: Example of Genetic Algorithm

Table 3.1: Initial Population

						Distance	Mutation
Route 1	5	1	3	4	7	7	Original
Route 2	5	2	4	3	1	11	Flip
Route 4	5	2	3	4	1	9	Swap
Route 5	5	3	4	1	2	6.5	Slide

These four randomly chosen routes are mutated using randomly chosen insertion indices which in this example are 2 & 5. The minimum euclidean distance from the cross-over mutation is compared against the minimum stored distance from the initial population. If the new route is lower than the stored initial population, the new route becomes the stored route. The process is repeated until some type of stopping criteria, in this case when the number of iterations, is met.

Table 3.2: Four Randomly Chosen Routes

						Distance
Route 1	3	4	1	2	5	10
Route 2	2	1	4	5	3	25
Route 3	1	2	3	5	4	15
Route 4	5	1	3	4	2	7
Route 5	1	5	4	3	2	20

Table 3.3: Cross-Over Mutation of Randomly Chosen Routes

						Distance
Route 1	3	4	1	2	5	10
Route 2	2	1	4	5	3	25
Route 3	1	2	3	5	4	15
Route 4	5	1	3	4	2	7
Route 5	1	5	4	3	2	20

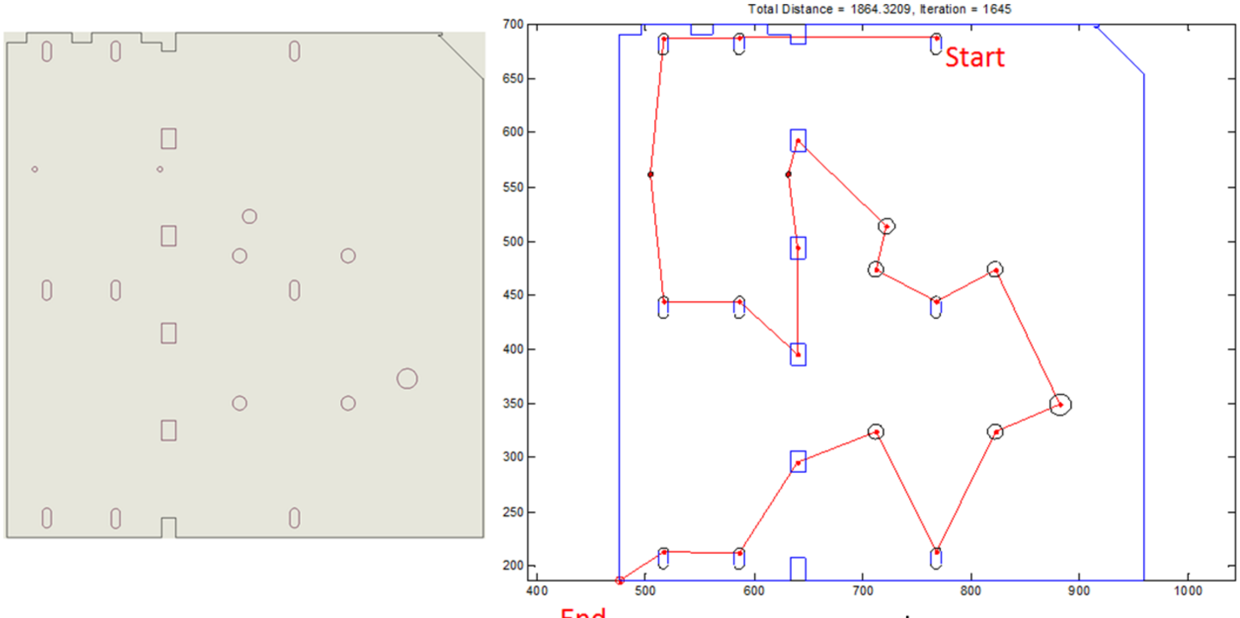
3.4 Laser Cutting Time Results

A select number of complex parts was chosen for validation to demonstrate the precision of the genetic algorithm as well as its limitations. This application of GA based TSP for laser travel is presented for four example parts.

3.4.1 Laser Estimation Example 1

A drawing of Part 1 is shown in Figure 3.8b, and has 21 interior features. The shortest laser traversal route calculated by the GA is shown in Figure 3.8a.

The travel time estimated by the GA is shown in Equation 3.5 an is 0.1831 minutes. The time estimate calculated as the cut time is 1.9942 minutes, giving a total laser cut time of 2.1773 minutes per part.



a

b

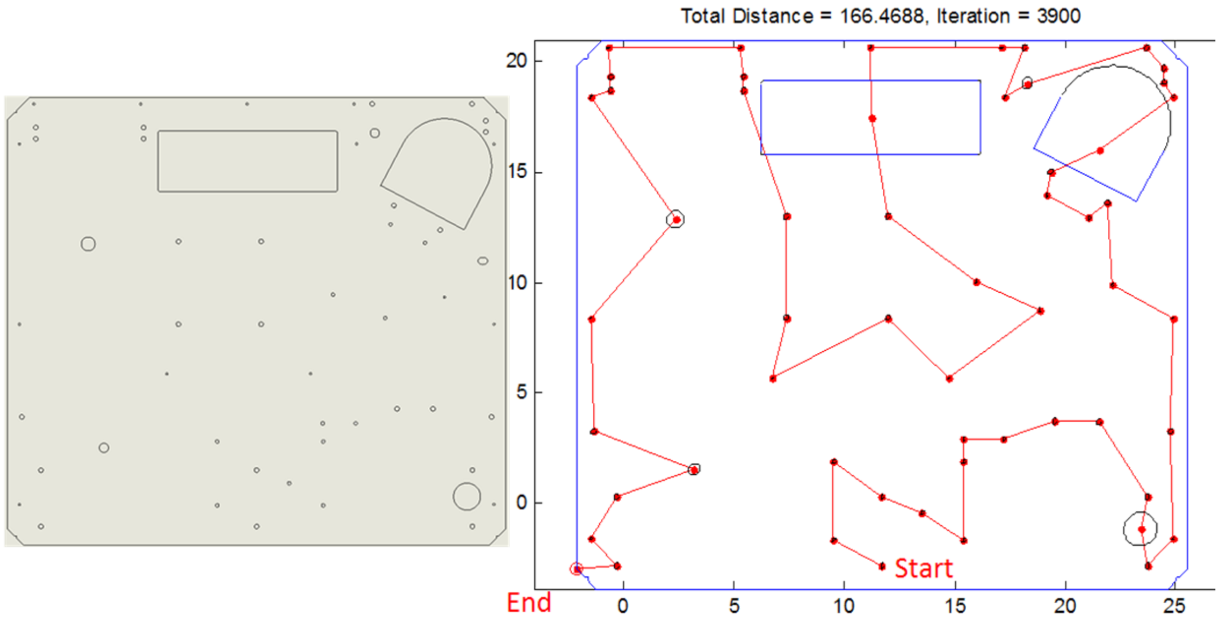
Figure 3.8: Laser Cutting Time Example 1: DXF Generated Drawing (a), Matlab generated drawing with shortest traversal path (b)

$$\begin{aligned} Time_{total} &= Time_{cut} + Time_{trav} \\ Time_{total} &= 1.9942 + 0.1831 \\ Time_{total} &= 2.1773 \text{ Minutes} \end{aligned} \tag{3.5}$$

3.4.2 Laser Estimation Example 2

A drawing of Part 2, shown in Figure 3.9 on the right, has 55 interior features. The shortest laser traversal route calculated by the GA is shown in Figure 3.9b which was determined in 3900 iterations.

The travel time estimate for the estimated by the GA is presented in Equation 3.6a as being 0.0162 minutes. The time estimate calculated as the cut time is seen as 6.4483 minutes, giving a total laser cut time of 6.4645 minutes per part.



a

b

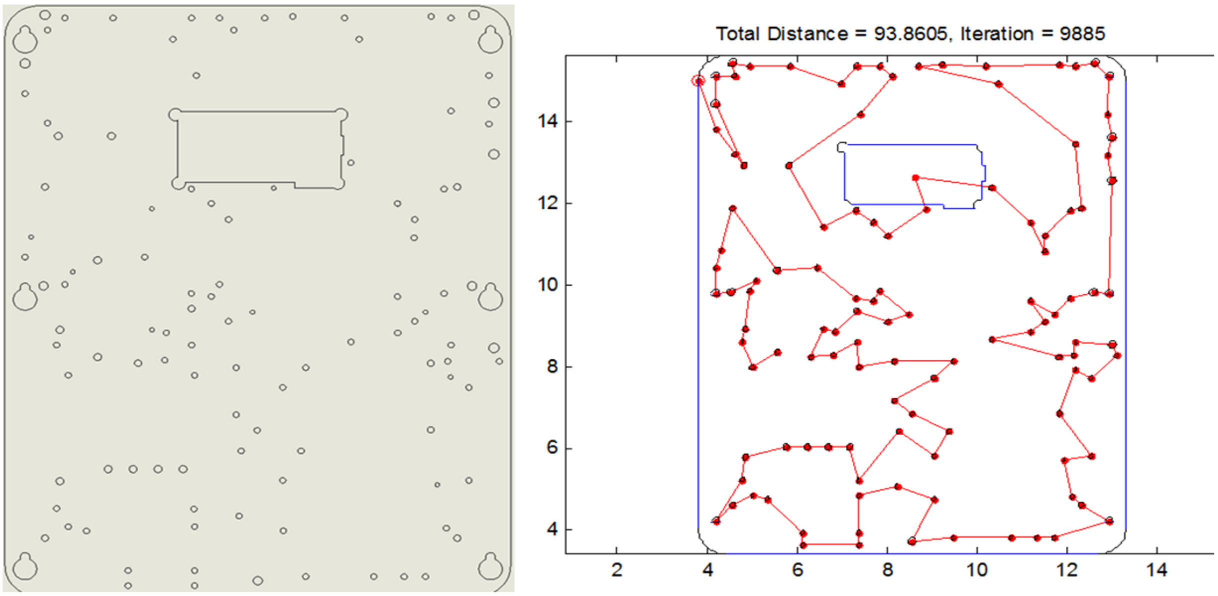
Figure 3.9: Laser Cutting Time Example 2: DXF Generated Drawing (a), Matlab generated drawing with shortest traversal path (b)

$$\begin{aligned} Time_{total} &= Time_{cut} + Time_{trav} \\ Time_{total} &= 6.4483 + 0.0162 \\ Time_{total} &= 6.4645 \text{ Minutes} \end{aligned} \tag{3.6}$$

3.4.3 Laser Estimation Example 3

A drawing of Part 3, shown in Figure 3.10 on the right, has 117 interior features. The shortest laser traversal route calculated by the GA is presented in Figure 3.10a and was determined in 9885 iterations.

The travel time estimate for the estimated by the GA is presented in Equation 3.7 as 0.0077 minutes. The time estimate calculated as the cut time as 4.9762 minutes, giving a total laser cut time of 4.9839 minutes per part.



a

b

Figure 3.10: Laser Cutting Time Example 3: DXF Generated Drawing (a), Matlab generated drawing with shortest traversal path (b)

$$\begin{aligned} Time_{total} &= Time_{cut} + Time_{trav} \\ Time_{total} &= 4.9762 + 0.0077 \\ Time_{total} &= 4.9839 \text{ Minutes} \end{aligned} \tag{3.7}$$

3.4.4 Laser Estimation Example 4

The last and most complex Part 4, has a total of 424 interior features. The shortest laser traversal route calculated by the GA is presented in Figure 3.11a and was determined in 9941 iterations before an optimal solution was found.

The estimated travel time based by the GA is presented in Equation 3.8 and is 0.0198 minutes. The time estimate calculated as the cut time is 6.4483 minutes, giving a total laser cut time of 6.4681 minutes per part.

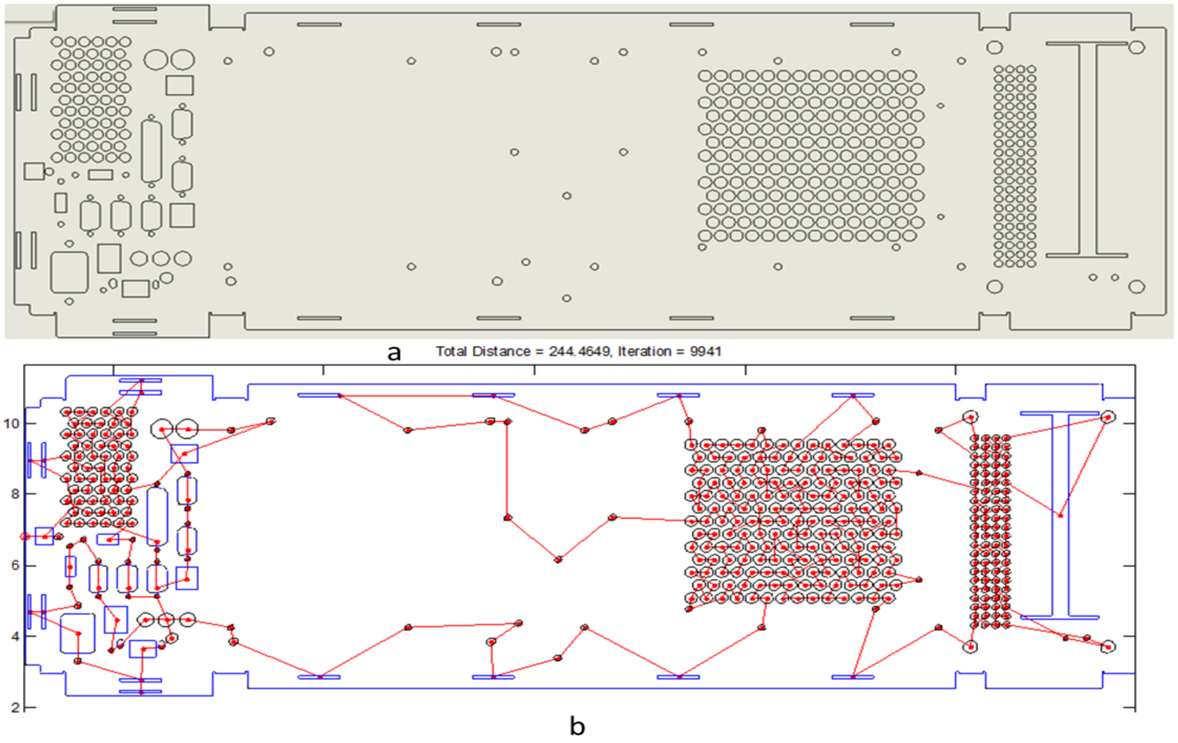


Figure 3.11: Laser Cutting Time Example 4: DXF Generated Drawing (a), Matlab generated drawing with shortest traversal path (b)

$$\begin{aligned} Time_{total} &= Time_{cut} + Time_{trav} \\ Time_{total} &= 6.4483 + 0.0198 \\ Time_{total} &= 6.4681 \text{ Minutes} \end{aligned} \tag{3.8}$$

3.4.5 Laser Cutting Results Summary

As seen in Figures 3.8- 3.11, the genetic algorithm determines the shortest route from a central point within the interior feature. Once an optimum route is established, the traversal path length is calculated from a corner that is nearest to the prior end point of the laser, and not from the central point within the interior features. For interior features that are circles, instead of corners, north, south, east, and west points from the center point are chosen for calculating the traversal path distance. A summary of the results of the four examples are shown in Table 3.4.

Table 3.4: Laser Cutting Time Results (in hours) per Part

Part Number	Calculated Time Using GA for Laser ad Cutting Time	Number of Features	Iterations
Part 1	2.1773	21	1645
Part 2	6.4645	55	3900
Part 3	4.9839	117	9885
Part 4	6.4681	424	9941

In Figure 3.11 the limitations of the Genetic Algorithm are more apparent when looking at the traversal route within features that are patterned and extremely close together. This increases the complexity to the algorithm by adding thousands of possible routes that are suboptimal. In such situations like this, it may be advantageous to create a single boundary feature around pattern interior features or closely positioned features.

A job that consisted of over 2900 parts that was cut using the laser using traditional methods of tool planning and time estimation were recalculated using the new algorithms developed. The parts were run through the MATLAB program developed to calculate the cut time and the laser traversal and the results are shown in Table 3.5.

Table 3.5: Laser Cutting Time Results (in hours) per Job

Actual Time	Calculated Time Using GA	Calculated Time w/o GA	Skilcraft Estimate
71.145	70.59427	63.41252	37.83
Percent Error	0.780%	10.868%	46.827%

Not only are the results more accurate, what used to take hours in estimating laser cutting time now takes a fraction of the time. The program is automated which allows for user to multi-task while obtaining accurate results.

3.5 Weld Time Estimation

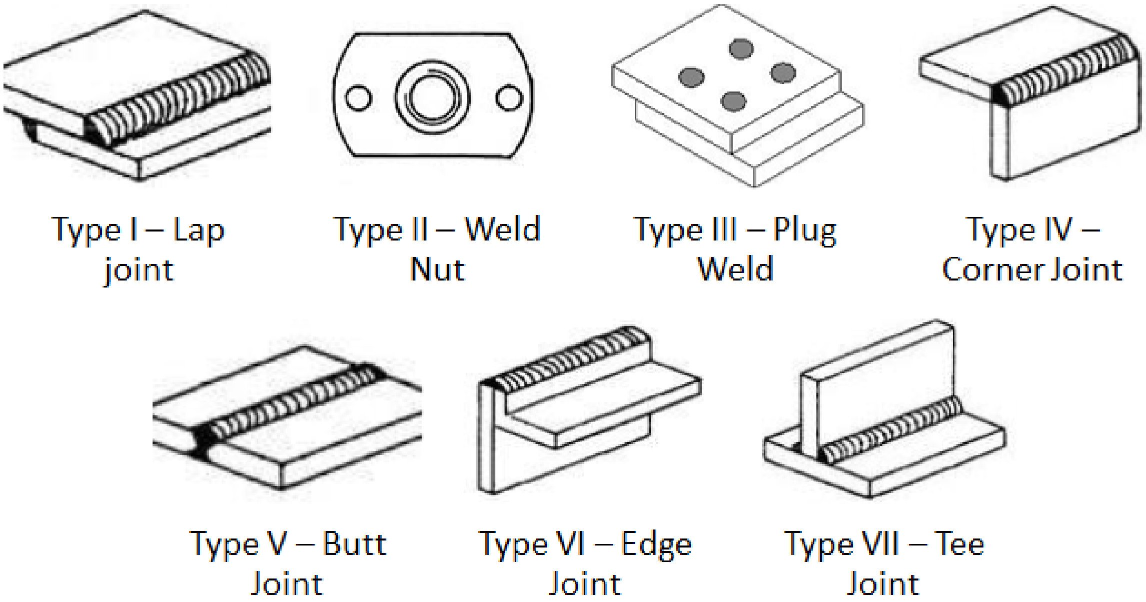
Once the parts have been cut, they go through many additional process before being sent for the final assembly. One of those additional processes is the welding operation. Again it is imperative to know how long the process will take in order to accurately schedule and price the product among other things.

Current cost estimation technique used by Skilcraft is purely based on expert judgment. Sales employees estimate weld times based on jobs they have performed in the past without any consistent company-wide format. This gives rise to inconsistent estimates between other sales employees and at times with the same employee estimating at different times. Over the last 4 years, subjectivity and inconsistency has led to an estimated $98,000 lost at Skilcraft due to labor inefficiency in welding department [15]. There are huge discrepancies between the time estimated to weld the assembly and the actual time it took to weld the assembly, with the average weld time estimate percent error being approximately 102%.

Two methods were utilized in this work in efforts to decrease the error of estimation versus actual time, the qualitative approach of case-based reasoning and the quantitative method of Regression. The first step is to collect historical data that will be used in the cost estimating techniques. Over 6,000 data points from the period 2009-2012 were obtained which included the name of the assembly, the time the assembly took to be welded, and the time that was estimated for the assembly to be welded. From these 6,000 data points, each assembly was analyzed and physical characteristics of each of them were quantified in a tabular form. The following information was obtained from each assembly:

- Number of Parts in the Assembly
- Batch Quantity (Order Quantity)
- Surface Area (in^2)
- Volume (in^3)
- Weight (lb)
- Material Gauge
- Material Type

Along with these physical characteristics, the types of welds that were used in the assembly were also noted and quantified.



Type I – Lap joint

Type II – Weld Nut

Type III – Plug Weld

Type IV – Corner Joint

Type V – Butt Joint

Type VI – Edge Joint

Type VII – Tee Joint

http://www.tpub.com/steelworker1/16.htm

Figure 3.12: Types of Weld Classification

3.5.1 Analysis of Variance (ANOVA)

Knowing all of the physical attributes of each welded assembly, it was necessary to know which of these characteristics were influential and significant in the overall assembly time. The assumption was made that the types of welds used in the assembly were essential, but not all of the physical characteristics were necessary. In order to determine which of the physical characteristic would be used in the cost estimating techniques, a multi-way analysis of variance (ANOVA) was performed. An ANOVA is helpful when trying to determine the influence of independent variables on a results, in this case assembly time. Each factor has a corresponding hypothesis test and looks to prove the null hypothesis:

H0 = The physical characteristic has no effect on the overall time

H1 = The physical characteristic has a significant effect

A p-value results from each hypothesis test and can be translated to the probability of observing a result at least as extreme as the one observed already. When the p-value is low (less than 0.05 or 5%) it is interpreted as being a highly unlikely that an example would be more extreme than the one observed, resulting in the rejection of the null hypothesis. In the case of physical characteristics of weld assemblies, this would assume that the characteristic is significant to the assembly time.

3.5.2 Case-Based Reasoning

Case-based reasoning is simply the idea of solving new problems based on historical problems that have similar features. It is ideal in situations when complex computing power is not available and when there is a plethora of historical data. The nearest neighbor (NN) estimation idea is a rudimentary approach in determining an output for features that may be similar [14]. The graphical example of this is shown in Figure 3.13 where the red star represents the data point in need of estimation. Euclidean distances from this points to the historical data points, shown in blue, are calculated. The output of the data point with the minimum distance (nearest neighbor) represents the output for the point in question.

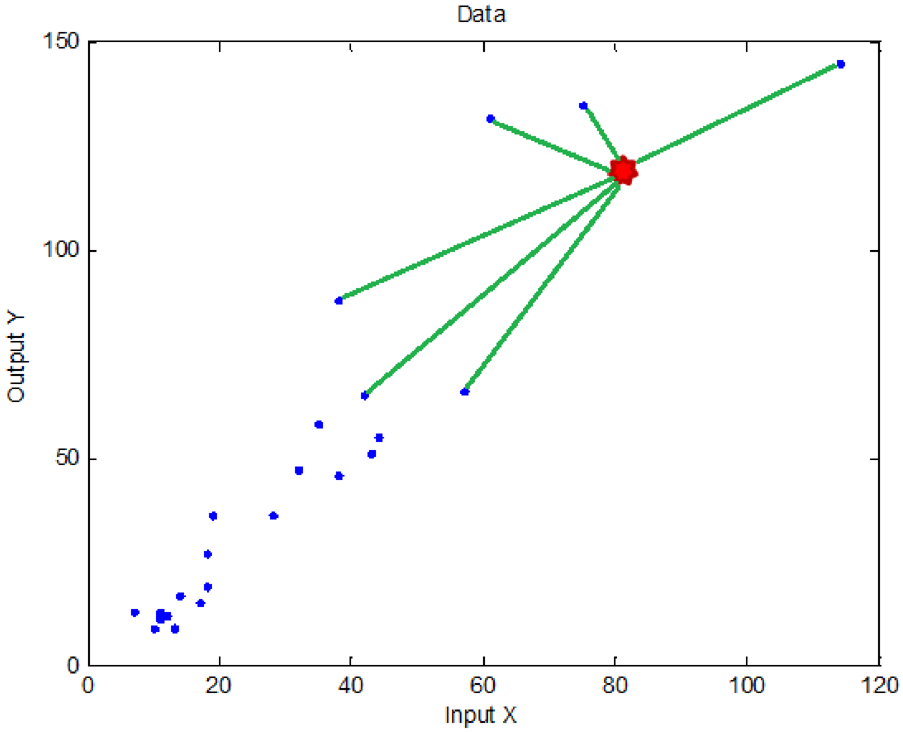


Figure 3.13: Example of Nearest Neighbor Principle

In determining the weld assembly times, the user will input a numerical description of the features of the assembly. The euclidean distance between the user inputs and the historical data are calculated in a effort to find the most similar assembly. The output of the assembly with the smallest deviation from the user input is then classified as the final output.

Modifications are often made to this approach in efforts to increase accuracy. Applying a weighted system to the distances between the user inputs and historical features will vary the relative importance of factors that contribute more to the overall welded assembly time.

The biggest benefit to using case-based reasoning is that it is based on actual historical results. The algorithm will always return results that have been physically attained in the past. It is important to keep in mind the variance between the user input and historical data. The variance equation is shown in Equation 3.9 [14].

$$s = \sum |\Delta(f_1, f_2, \ldots, f_n)| \tag{3.9}$$

This minimization of s results in the assembly with the closest features and therefore should theoretically have a similar welded assembly time.

3.6 Weld Time Estimation Results

To again summarize the methodology in developing the weld time estimation program, a ANOVA was used to determine the significant factors in the overall welded assembly time. The factors that were initially considered are as follows:

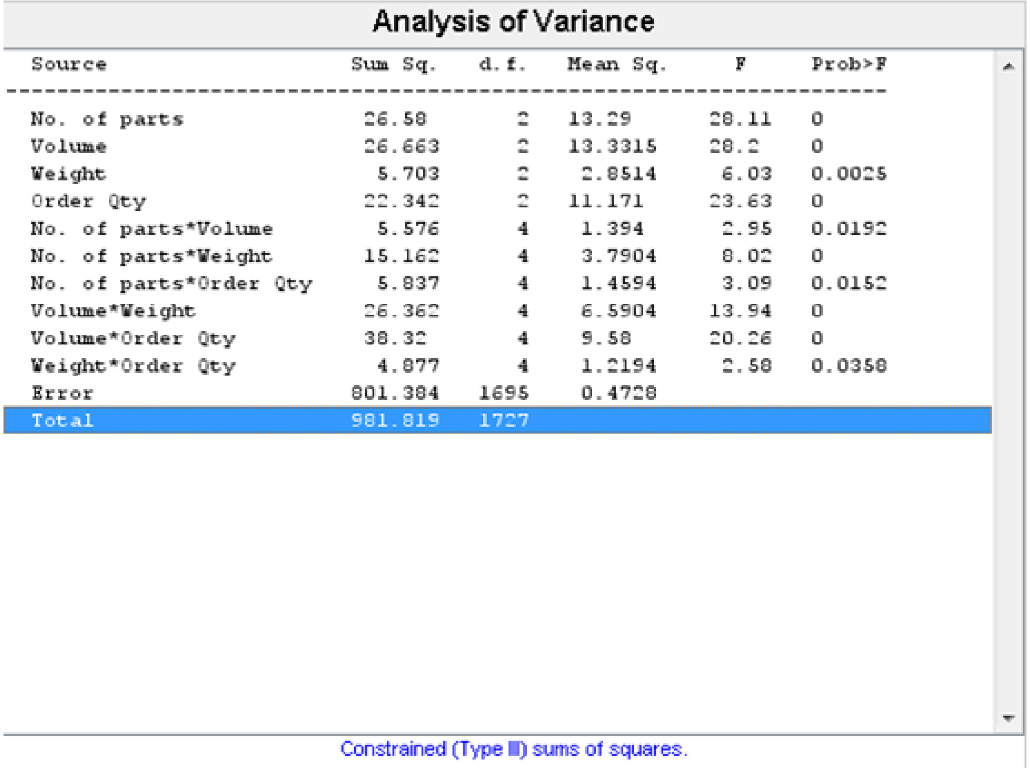
- Number of Parts in the Assembly
- Batch Quantity (Order Quantity)
- Surface Area (in^2)
- Volume (in^3)
- Weight (lb)
- Material Gauge
- Material Type

Since weight is the product of surface area, material gauge, and material type (density), these seven original characteristics were then narrowed down to the following:

- Number of Parts in the Assembly

- Batch Quantity (Order Quantity)
- Volume (in^3)
- Weight (lb)

An ANOVA was then performed to determine which of these physical characteristics contribute significantly to the overall assembly time. The results are shown in Figure 3.14.



Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
No. of parts	26.58	2	13.29	28.11	0
Volume	26.663	2	13.3315	28.2	0
Weight	5.703	2	2.8514	6.03	0.0025
Order Qty	22.342	2	11.171	23.63	0
No. of parts*Volume	5.576	4	1.394	2.95	0.0192
No. of parts*Weight	15.162	4	3.7904	8.02	0
No. of parts*Order Qty	5.837	4	1.4594	3.09	0.0152
Volume*Weight	26.362	4	6.5904	13.94	0
Volume*Order Qty	38.32	4	9.58	20.26	0
Weight*Order Qty	4.877	4	1.2194	2.58	0.0358
Error	801.384	1695	0.4728		
Total	981.819	1727			

Constrained (Type III) sums of squares.

Figure 3.14: Results of ANOVA and the Weld Assembly Physical Characteristics

From this table we can determine that all four of the physical characteristics are significant to the overall assembly time. These four physical characteristics along with the seven weld types were considered to be contributing to the overall assembly time and are therefore used in the cased based reasoning nearest neighbor calculations.

A plot of the main effects is seen in Figure 3.15 showing the mean at each treatment level (-1,0,1). There are some conclusions that can be draw from this plot. As the number of parts increase the overall assembly time will also increase. As the volume increases the average time decrease and then increases. This could be because extremely small and extremely large assemblies are often hard to work with and will require more time than an average sized assembly.

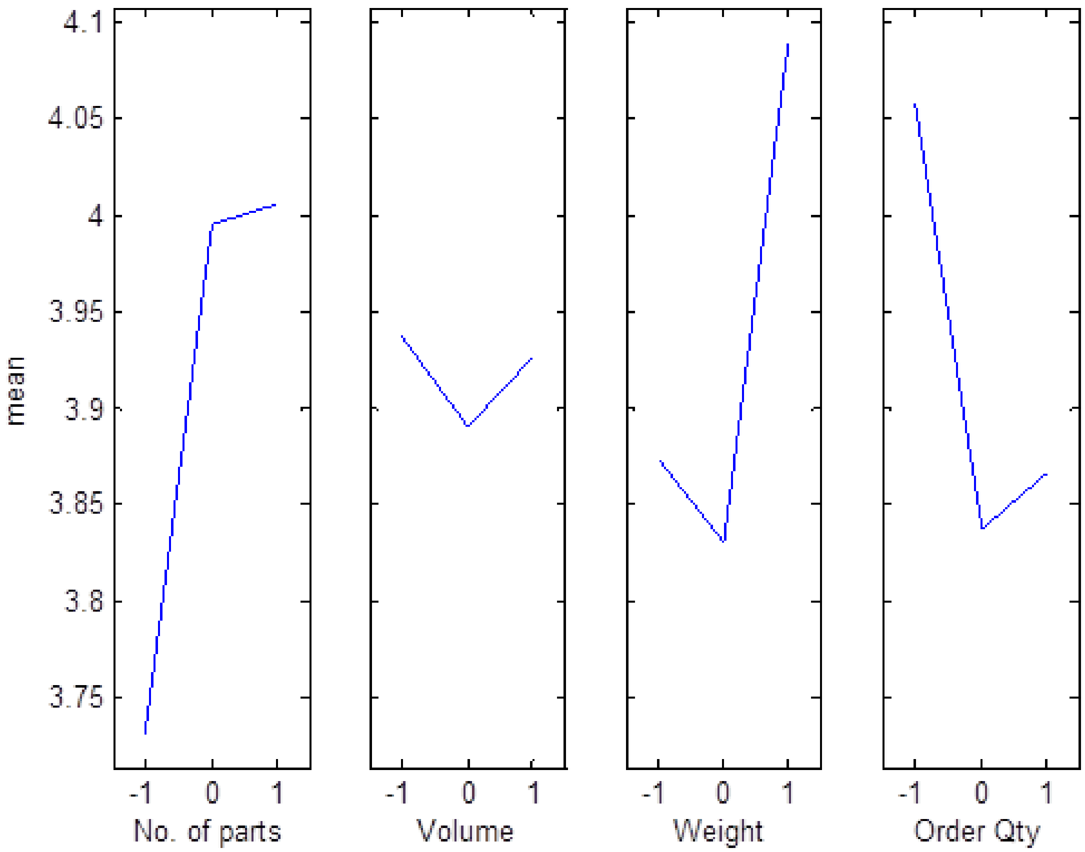


Figure 3.15: Main effects plot of the Physical Characterstics used in the ANOVA

3.6.1 Weld Estimation Program

A user friendly software program was developed in order to aid the engineering and sales team at Skilcraft more accurately to price weld assemblies. The excel based cost estimating program that was developed and has an easy to use interface as seen in Figure 3.18. There are two tabs to choose from to get a time estimates, Case-Based Reasoning or Existing Part. "Case-Based Reasoning" uses the nearest neighbor approach to estimate the time the assembly will take to be welded and "Existing Part" give the user the opportunity to look up data on a part that has already been welded. It was assumed that all of the 7 weld types were significant and in determining a closest nearest neighbor, the weld types would be matched first followed by the physical characteristics and then the order quantity. This hierarchical nearest neighbor algorithm is depicted in Figure 3.16. The nearest neighbor algorithm puts more emphasis on the weld types, and less emphasis on the order quantity.

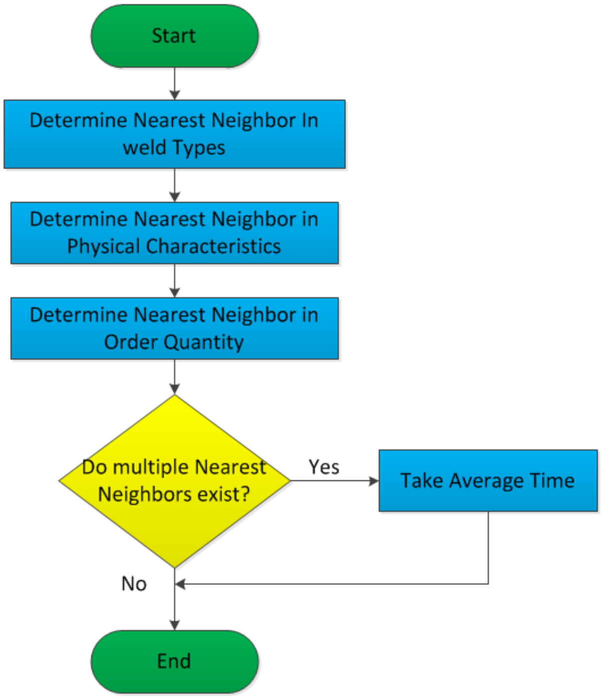
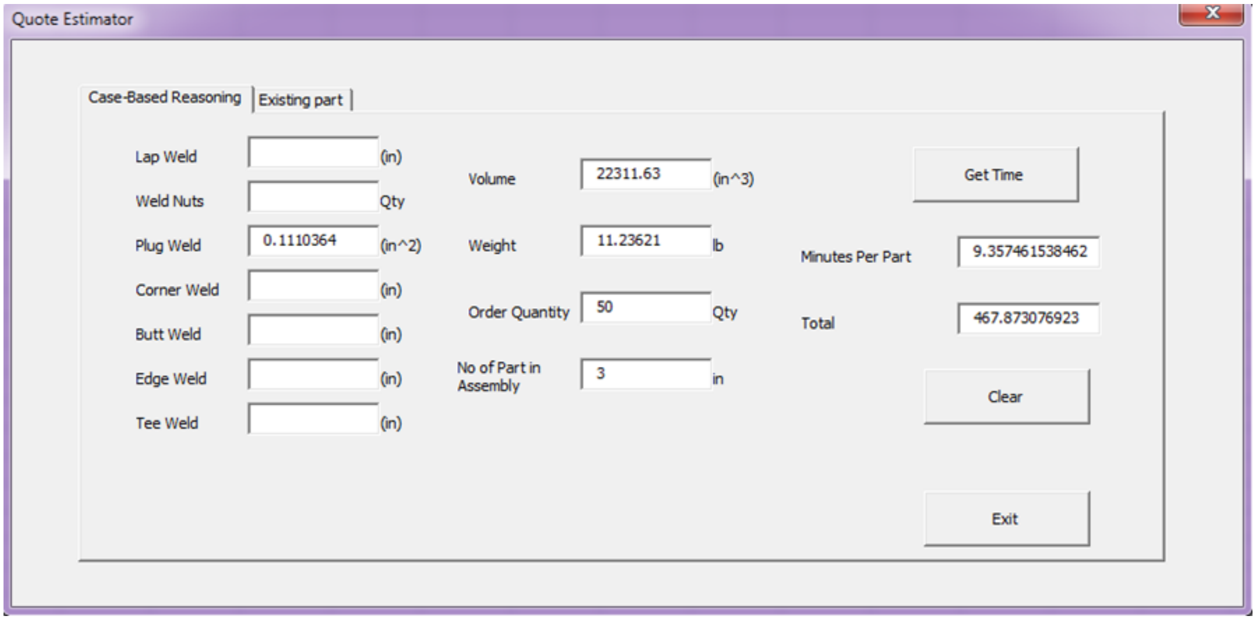


Figure 3.16: Nearest Neighbor Algorithm with Hierarchical Classification

3.6.2 Weld Estimation Example 1

Example 1 is an assembly of a box with three parts, a box, hinge, and lid. The physical characteristics and weld information was entered into the weld estimation program, shown in Figure 3.18, and the results are shown in Table 3.6.

Figure 3.17: Model of Example 1



Lap Weld (in)
Weld Nuts Qty
Plug Weld 0.1110364 (in^2)
Corner Weld (in)
Butt Weld (in)
Edge Weld (in)
Tee Weld (in)
Volume 22311.63 (in^3)
Weight 11.23621 lb
Order Quantity 50 Qty
No of Part in Assembly 3 in
Minutes Per Part 9.357461538462
Total 467.873076923

Figure 3.18: User Inputs for Weld Assembly Time Estimation for Example 1

Table 3.6: Weld Assembly Time Results (in min) Example 1

Average (min/part)	CBR (min/part)	Percent Error
8.056	9.357	16.15%

3.6.3 Weld Estimation Example 2

The assemble shown in Figure 3.19 of Example 2 is storage shelving unit consisting of just one piece. Corner welds and Butt welds are used to fuse the edges of the piece. Again, the physical characteristics and weld information was entered into the weld estimation program shown in Figure 3.20 and the results are shown in Table 3.7.

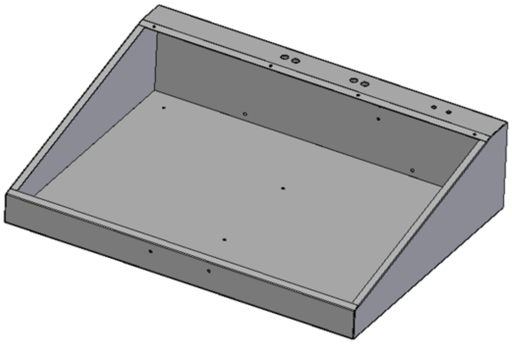
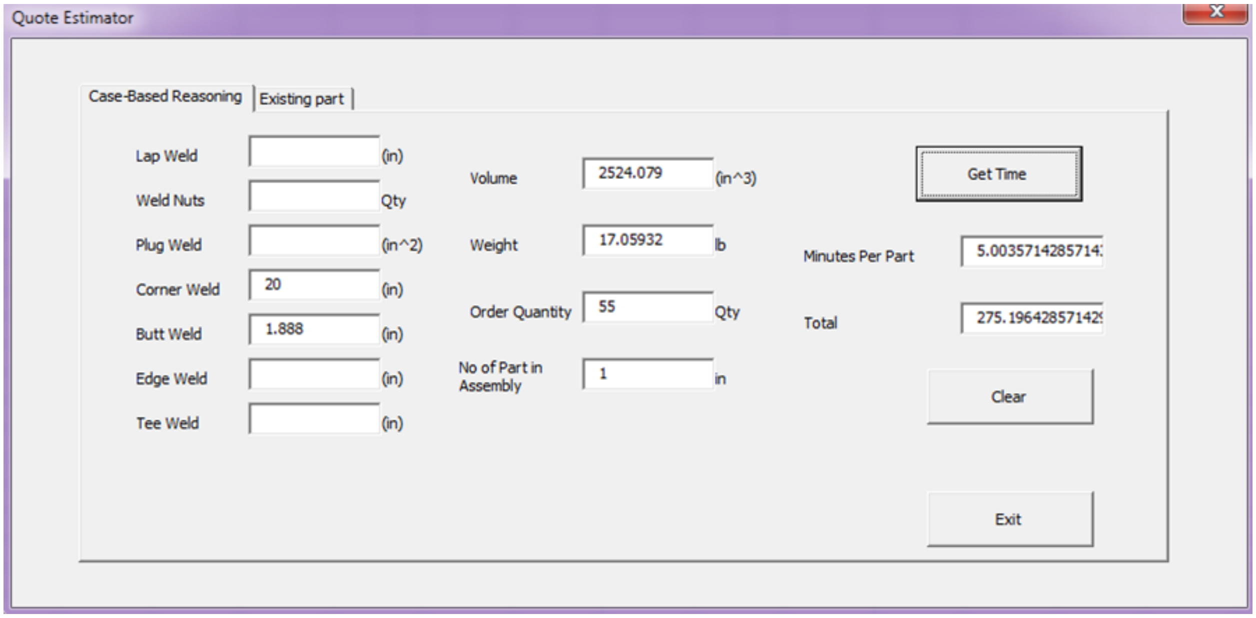


Figure 3.19: Model of Example 2



Quote Estimator

Lap Weld (in)
Weld Nuts Qty
Plug Weld (in^2)
Corner Weld 20 (in)
Butt Weld 1.888 (in)
Edge Weld (in)
Tee Weld (in)

Volume 2524.079 (in^3)
Weight 17.05932 lb
Order Quantity 55 Qty
No of Part in Assembly 1 in

Minutes Per Part 5.0035714285714
Total 275.19642857142

Figure 3.20: User Inputs for Weld Assembly Time Estimation Example 2

Table 3.7: Weld Assembly Time Results (in min) Example 2

Average (min/part)	CBR (min/part)	Percent Error
5.738	5.004	12.80%

The weld estimation program is fairly accurate, however will only increase in accuracy with more historical data.

The last tab gives users the capability of looking at parts that have been previously made seen in Figure 3.21. It will then return the average minutes per part based on the average historical data. The total is the product of the quantity and the minutes per part, which represents the total time it will take to weld the entire customer order.

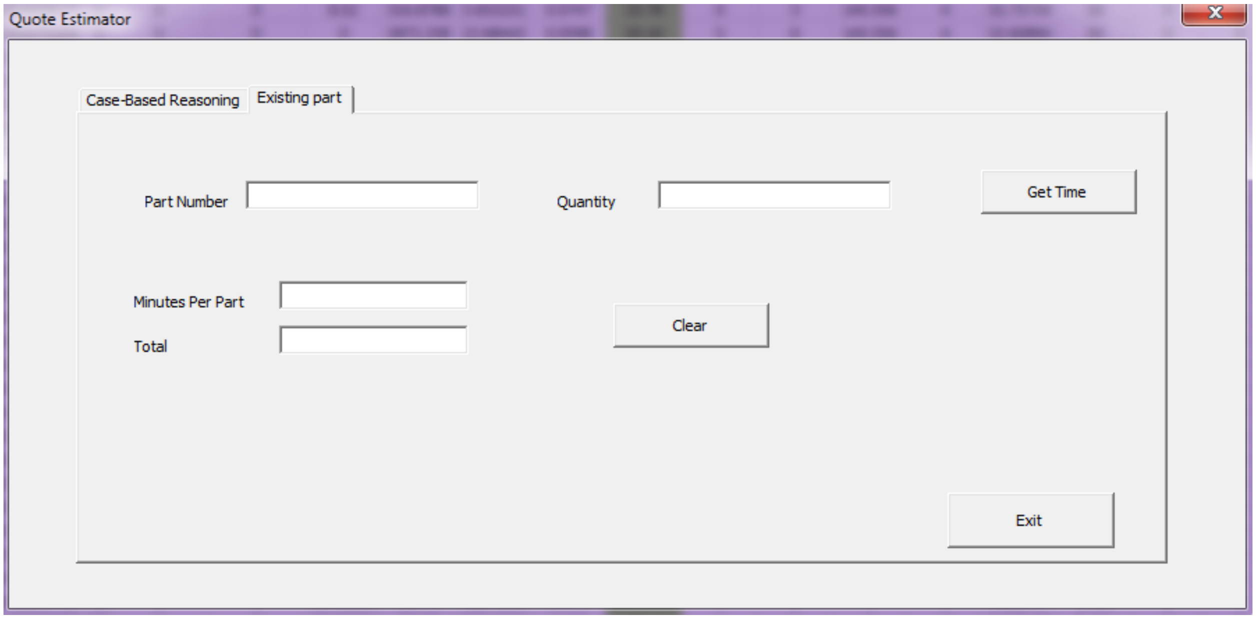
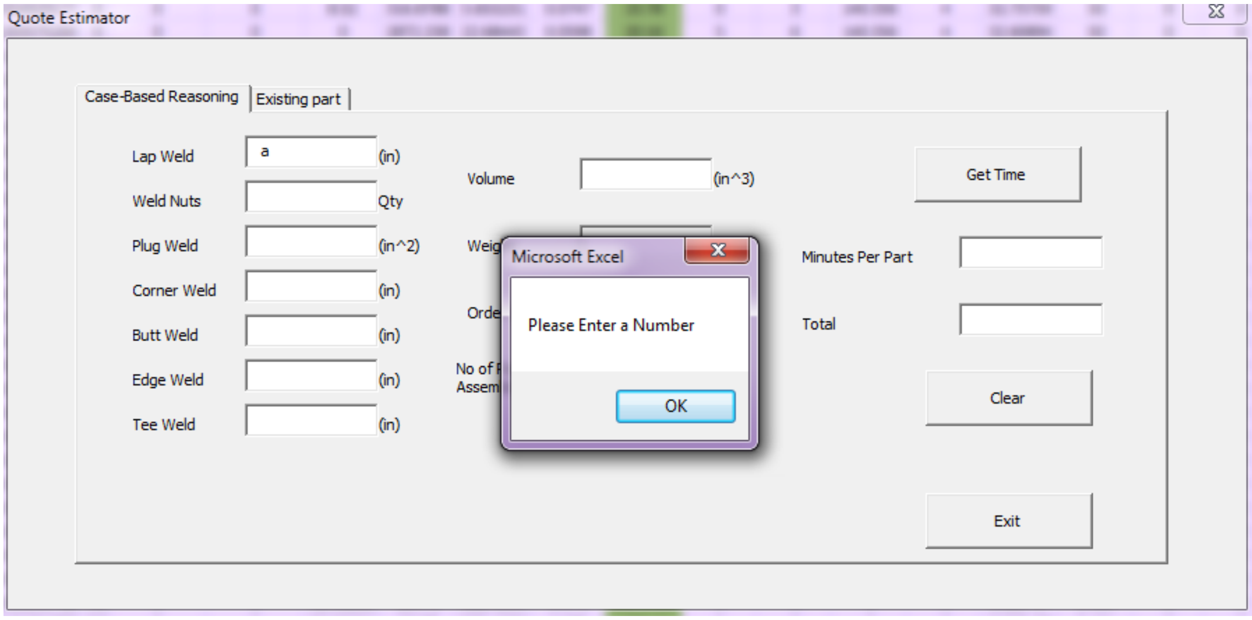


Figure 3.21: User Interface for Time of Existing Part

The time estimator was also programmed to catch user input errors shown in Figure 3.22, helping to guide the user on how to use the estimator program.



Lap Weld: a (in)

Please Enter a Number

Figure 3.22: Error Corrector

Creating a robust program allows for user of any skill level to accurately estimate weld assembly times.

Chapter 4

Conclusion

4.1 Summary of Results

This thesis focused on three aspects of improving productivity and reducing cost in a fabrication facility focused on assembly of welded parts. They are (a) a methodoogy and associated software by assessing weld assembly time, (b) an optimization approach to minimize laser travel time during the cutting process and (c) a design for manufacturability analysis to minimize quality problems and scrap rate.

To accurately predict process times, a weld assembly time estimating approach and associated software was developed. This proved to be 67% more accurate in predicting the time when compared to the current estimation approach being used at Skilcraft.

In addition to the weld assembly time estimating program, a laser cutting time estimation methodology was developed accurately estimated laser cutting times.

The Design for Manufacturability rules developed and implemented in this thesis for customer provided sheet metal parts will help increase productivity on the shop floor allowing for quicker turn around times at Skilcraft, it will also increase tool life and save money in machining services, and it will show customers what sets Skilcraft apart from the competition.

These tools were created specifically to aid in optimizing and estimating the laser cutting times and welded assembly times for Skilcraft. They are designed to be used in conjunction with knowledge and experience of the machine operators and the engineers to optimize the process and create more accurate time standards quicker and easier.

4.2 Future Work

There are several improvements that could be made to increase the accuracy of the estimation tools presented. With more historical data, the nearest neighbor analysis presented here will become progressively more accurate in estimating the weld assembly time. Also, with large amounts of historical data, an intelligent system, such as a neural network or support vector machine, maybe another option for increasing accuracy.

The laser travel optimization developed here can be improved by tightly integrating with the sheet metal nesting algorithm. In addition, actual laser start and end points for each 2D shape can be taken into consideration for the optimization.

Improvements to the manufacturability analysis presented in this thesis would be very helpful in finding ways to aid the manufacturing process. Additional Design for Manufacturing sheet metal rules can be added to the software relatively easily. The current software also has capabilities of analyzing bend lines and whether interior features comply with bend line clearances. This is a common reason for re-design that is not captured in the 3-D model.

These three tools along with Skilcraft's continuous improvement initiatives will ensure that the company stays on the cutting edge of the sheet metal business. Internal improvements to the company will be directly seen by the customer in terms of quicker turn around times with the same high quality that Skilcraft has built its reputation on. Skilcraft's motivation to improve on areas of weakness shows a company that is willing to grow to meet the needs and satisfaction of their customers.

References

[1] Kenneth Baker. *Optimization Modeling with Spreadsheets*. Wiley, Hoboken, NJ, 2011. Print.

[2] Roger William Bolz. Production process: their influence on design. Penton Pub. Co, Cleveland.

[3] James G. Bralla. *Design for Manufacturability Handbook*. McGraw-Hill, 2 edition, 1999.

[4] Kenneth Castelino and Paul K. Wright. Tool-path optimization for minimizing airtime during machining. *Journal of Computing and Information Science in Engineering*, pages 235–241, 2004.

[5] Frank Barkley Copley. *Frederick W. Taylor: father of scientific management*. Harper and brothers, 1923.

[6] Nicholas D. Ernest. *UAV Swarm Cooperative Control Based on a Genetic-Fuzzy Approach*. PhD thesis, University of Cincinnati, 2012.

[7] Shashikiran Hegde. A sheet metal design advisor: Design rules and inter-feature design rule checking. Master's thesis, University of Cincinnati, 2002.

[8] Joseph Kirk. Traveling salesman problem - genetic algorithm. MATLAB Central File Exchange, 2007.

[9] Rajender Singh G.S. Sekhon Kumar, Shailendra. Cckbs: A component check knowledge-based system for assessing manufacturability of sheet metal parts. *Journal of Material Processing Technology*, 172:64–69, 2006.

[10] Charles S. Maier. Between taylorism and technocracy: European ideologies and the vision of industrial productivity in the 1920s. *Journal of Contemporary History*, 5(2), 1970.

[11] Harold Bright Maynard and Kjell B Zandin. *Maynard's Industrial Engineering Handbook*. McGraw-Hill, 2001.

[12] H. S. Mianaei and S. H. Iranmanesh. Case-based reasoning method in cost estimation of drilling wells. *Research Journal of Applied Sciences, Engineering and Technology*, 5(4):1086–1112, 1999.

[13] Jongsoo Kim Gyunghyun Choi Yoonho Seo Moon, Chiung. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research*, 140:606–617, 2002.

[14] Adnan Niazi, Jian S. Dai, Stavroula Balabani, and Lakmal Seneviratne. Product cost estimation: Technique classification and methodology review.

[15] Bureau of Labor Statistics. Occupational outlook handbook. Technical report, United States Department of Labor, 2010.

[16] Quality Tool Inc. *Sheet Metal Design Handbook*.

[17] Raj Radhakrishnan, Araya Amsalu, Mehran Kamran, and B.O. Nnaji. Design rule checker for sheet metal components using medial axis transforms and geometric reasoning. *Journal of Manufacturing Systems*, 15:179–189, 1996.

[18] Christopher Rush and Rajkumar Roy. Expert judgement in costing estimating: Modelling the reasoning process. *Concurrent Engineering: Research and Applications*, 9(4):271–284, 2001.

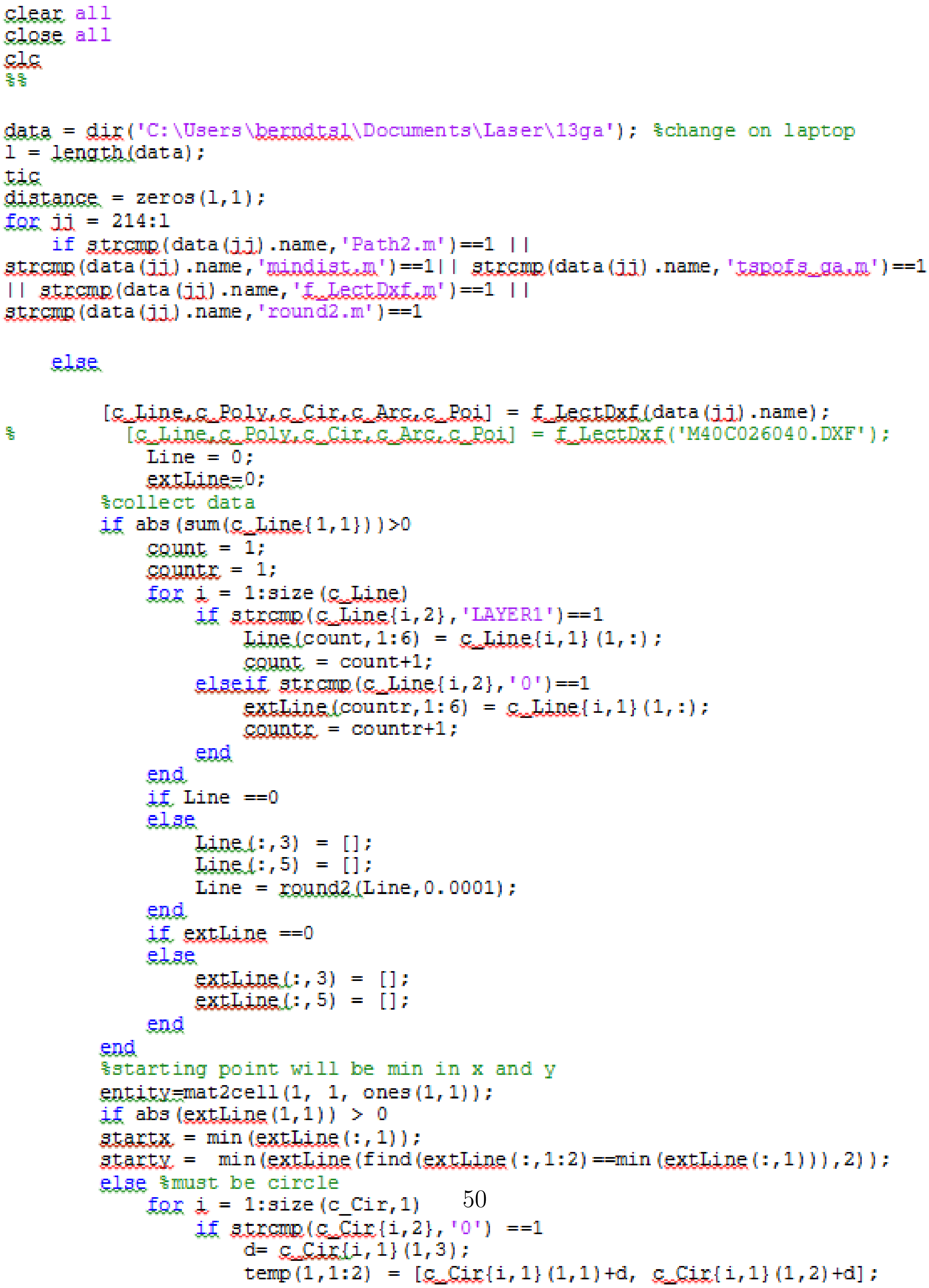
[19] Sebastian. Read dxf file data. MATLAB Central File Exchange, 2009.

[20] Michael C. Wood and John C. Wood, editors. *Frank and Lillian Gilbreth: Critical Evaluations in Business and Management*, volume 1. Routledge, 2003.

[21] Y.F. Zhang and J.Y.H. Fuh. A neural network approach for early cost estimation of packaging products. *Computers and Industrial Engineering*, 34(2):433–450, 1998.

Appendix

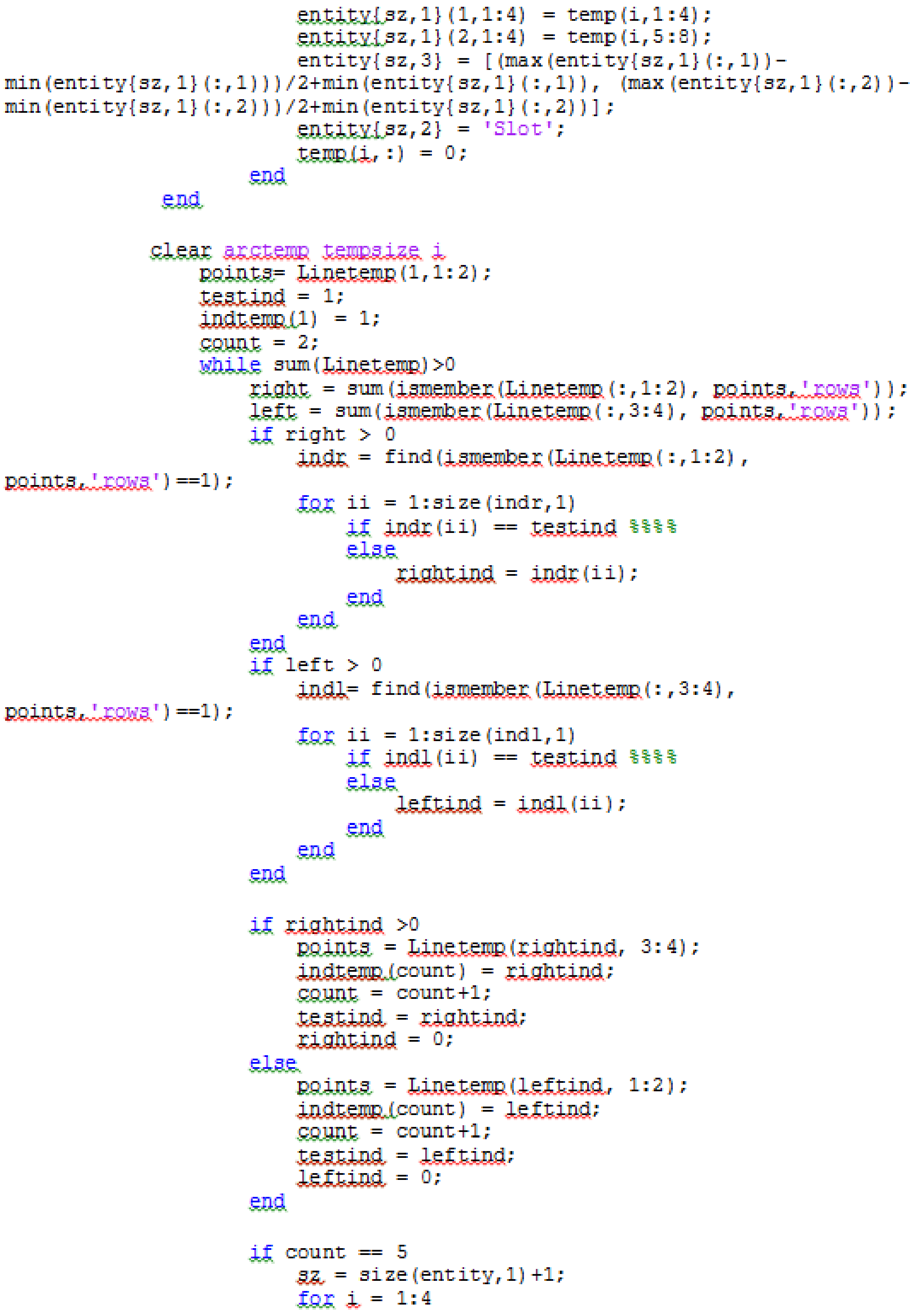
Matlab Code



```
clear all
close all
clc
%%

data = dir('C:\Users\berndtsl\Documents\Laser\13ga'); %change on laptop
l = length(data);
tic
distance = zeros(1,1);
for jj = 214:1
    if strcmp(data(jj).name,'Path2.m')==1 ||
strcmp(data(jj).name,'mindist.m')==1|| strcmp(data(jj).name,'tspofs_ga.m')==1
|| strcmp(data(jj).name,'f_LectDxf.m')==1 ||
strcmp(data(jj).name,'round2.m')==1

    else

        [c_Line,c_Poly,c_Cir,c_Arc,c_Poi] = f_LectDxf(data(jj).name);
%         [c_Line,c_Poly,c_Cir,c_Arc,c_Poi] = f_LectDxf('M40C026040.DXF');
            Line = 0;
            extLine=0;
        %collect data
        if abs(sum(c_Line{1,1}))>0
            count = 1;
            countr = 1;
            for i = 1:size(c_Line)
                if strcmp(c_Line{i,2},'LAYER1')==1
                    Line(count,1:6) = c_Line{i,1}(1,:);
                    count = count+1;
                elseif strcmp(c_Line{i,2},'0')==1
                    extLine(countr,1:6) = c_Line{i,1}(1,:);
                    countr = countr+1;
                end
            end
            if Line ==0
            else
                Line(:,3) = [];
                Line(:,5) = [];
                Line = round2(Line,0.0001);
            end
            if extLine ==0
            else
                extLine(:,3) = [];
                extLine(:,5) = [];
            end
        end
        %starting point will be min in x and y
        entity=mat2cell(1, 1, ones(1,1));
        if abs(extLine(1,1)) > 0
        startx = min(extLine(:,1));
        starty =  min(extLine(find(extLine(:,1:2)==min(extLine(:,1))),2));
        else %must be circle
            for i = 1:size(c_Cir,1)
                if strcmp(c_Cir{i,2},'0') ==1
                    d= c_Cir{i,1}(1,3);
                    temp(1,1:2) = [c_Cir{i,1}(1,1)+d, c_Cir{i,1}(1,2)+d];
```

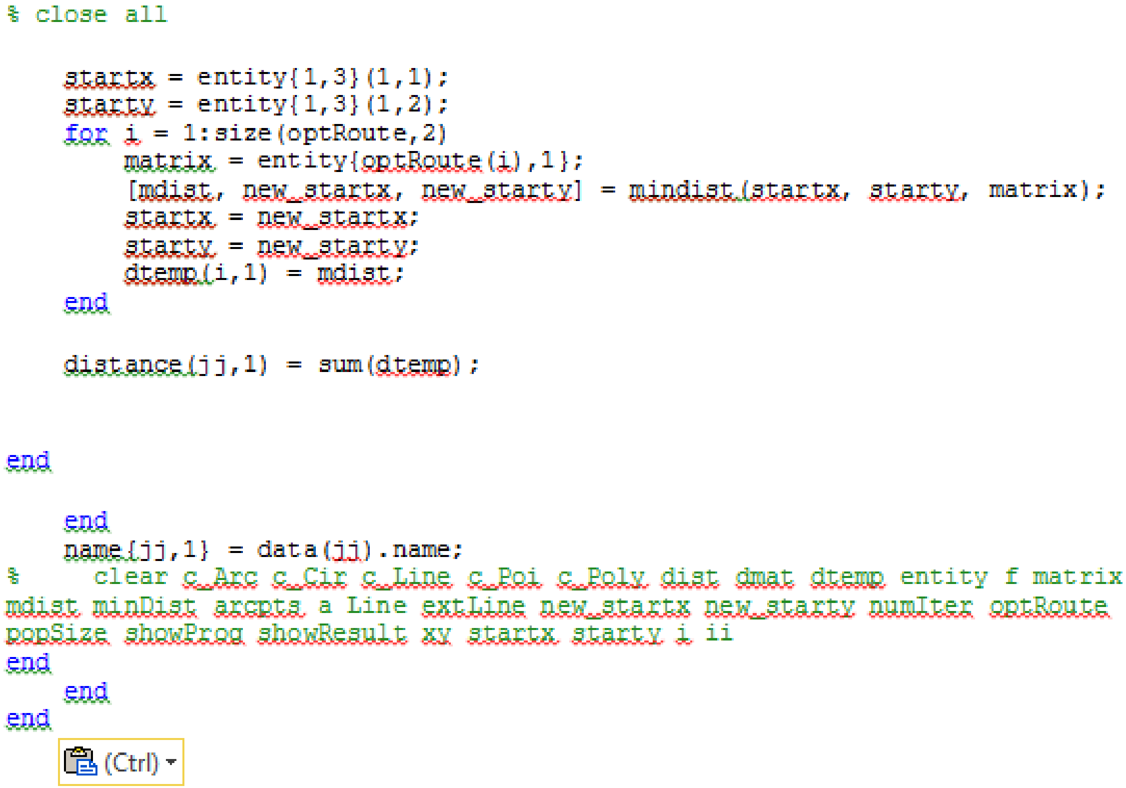
```
                    temp(2,1:2) = [c_Cir{i,1}(1,1)-d, c_Cir{i,1}(1,2)-d];
                    temp(3,1:2) = [c_Cir{i,1}(1,1)+d, c_Cir{i,1}(1,2)-d];
                    temp(4,1:2) = [c_Cir{i,1}(1,1)-d, c_Cir{i,1}(1,2)+d];
                end
            end
            tt = min(temp);
            startx = tt(1);
            starty = tt(2);
            clear temp tt
        end
        entity{1,1} = [startx, starty];
        entity{1,2} = 'Start';
        entity{1,3} = [startx, starty];
%         clear startx starty extLine
        if sum(c_Cir{1,1})>0 %circles are automatic entities
            count = 2;
            for i = 1:size(c_Cir,1)
                d = c_Cir{i,1}(1,3);
                entity{count,1}(1,1:2) = [c_Cir{i,1}(1,1)+d,
c_Cir{i,1}(1,2)+d];
                entity{count,1}(1,3:4) = [c_Cir{i,1}(1,1)-d, c_Cir{i,1}(1,2)-
d];
                entity{count,1}(2,1:2) = [c_Cir{i,1}(1,1)+d, c_Cir{i,1}(1,2)-
d];
                entity{count,1}(2,3:4) = [c_Cir{i,1}(1,1)-d,
c_Cir{i,1}(1,2)+d];
                entity{count,2} = 'Circle';
                entity{count,3} = [c_Cir{i,1}(1,1),c_Cir{i,1}(1,2)];
                count = count+1;
            end
        end
        if sum(c_Arc{1,1})>0 && Line(1,1) >0 && size(Line,1)>1
            go = 'no';
            for i = 1:size(c_Arc,1)
                if strcmp(c_Arc{1,2},'LAYER1')==1
                    go = 'yes';
                end
            end
            if strcmp(go, 'yes')
            count = 1;
            for ii = 1:size(c_Arc,1)
                if strcmp(c_Arc{ii,2},'LAYER1')==1
                    startpt = [cosd(c_Arc{ii,1}(1,4))*c_Arc{ii,1}(1,3),
sind(c_Arc{ii,1}(1,4))*c_Arc{ii,1}(1,3)];
                    endpt = [cosd(c_Arc{ii,1}(1,5))*c_Arc{ii,1}(1,3),
sind(c_Arc{ii,1}(1,5))*c_Arc{ii,1}(1,3)];
                    arcpts(count,1:2) = c_Arc{ii,1}(1,1:2)+startpt;  %start
x,y
                    arcpts(count,3:4) = c_Arc{ii,1}(1,1:2)+endpt;  %end x,y
                    count = count+1;
                end
            end
            arcpts = round2(arcpts,0.0001);
            count = 1;
            %need to determine arcs and lines connectivity
            for i = 1:size(arcpts,1)
```

```
                a = ismember(Line,arcpts(i,:)); %finds which two lines
connected to arc(i)
                if size(find(sum(a(:,1:2),2)==2),1)==1 &&
size(find(sum(a(:,3:4),2)==2),1)==1
                    tempind(1,1) = find(sum(a(:,1:2),2)==2);
                    tempind(2,1) = find(sum(a(:,3:4),2)==2);
                    storeind(count,:) = tempind;
                    count = count+1;
                elseif size(find(sum(a(:,1:2),2)==2),1)==2
                    en = find(sum(a(:,1:2),2)==2);
                    tempind(1,1) = en(1);
                    tempind(2,1) = en(2);
                    storeind(count,:) = tempind;
                    count = count+1;
                elseif size(find(sum(a(:,3:4),2)==2),1)==2
                    en = find(sum(a(:,3:4),2)==2);
                    tempind(1,1) = en(1);
                    tempind(2,1) = en(2);
                    storeind(count,:) = tempind;
                    count = count+1;
                end
                if size(tempind,1) == 2
                    temp(i,1:4) = Line(tempind(1),:);%which lines connect to
which arcs

                    temp(i,5:8) = Line(tempind(2),:);
                else
                    error('Arc Connectivity')
                end
            end
            arctemp = temp;
            tempsize = size(arctemp,1);
            temp = unique(temp,'rows');
            Linetemp = Line;
            for i = 1:size(storeind,1)
                Linetemp(storeind(i,1),:) = 0;
                Linetemp(storeind(i,2),:) = 0;
            end
            Linetemp( ~any(Linetemp,2), : ) = [];  %rows

            for i = 1:size(temp)
                    if sum(temp(i,:),2)>0 &&
sum(find(sum(ismember(temp(:,5:8),temp(i,1:4)),2)==4))>0 %slot criteria
                        del =
find(sum(ismember(temp(:,5:8),temp(i,1:4)),2)==4);
                        if temp(del,1:4) == temp(i,5:8)
                            sz = size(entity,1)+1;
                            entity{sz,1}(1,1:4) = temp(i,1:4);
                            entity{sz,1}(2,1:4) = temp(i,5:8);
                            entity{sz,3} = [(max(entity{sz,1}(:,1))-
min(entity{sz,1}(:,1)))/2+min(entity{sz,1}(:,1)), (max(entity{sz,1}(:,2))-
min(entity{sz,1}(:,2)))/2+min(entity{sz,1}(:,2))];
                            entity{sz,2} = 'Slot';
                            temp(i,:) = 0;
                            temp(del,:)=0;
                        end
                    elseif sum(ismember(arctemp,temp(i,:),'rows'))==2
                        sz = size(entity,1)+1;
```



```
                        entity{sz,1}(1,1:4) = temp(i,1:4);
                        entity{sz,1}(2,1:4) = temp(i,5:8);
                        entity{sz,3} = [(max(entity{sz,1}(:,1))-
min(entity{sz,1}(:,1)))/2+min(entity{sz,1}(:,1)), (max(entity{sz,1}(:,2))-
min(entity{sz,1}(:,2)))/2+min(entity{sz,1}(:,2))];
                        entity{sz,2} = 'Slot';
                        temp(i,:) = 0;
                    end
             end

            clear arctemp tempsize i
                points= Linetemp(1,1:2);
                testind = 1;
                indtemp(1) = 1;
                count = 2;
                while sum(Linetemp)>0
                    right = sum(ismember(Linetemp(:,1:2), points,'rows'));
                    left = sum(ismember(Linetemp(:,3:4), points,'rows'));
                    if right > 0
                        indr = find(ismember(Linetemp(:,1:2),
points,'rows')==1);
                        for ii = 1:size(indr,1)
                            if indr(ii) == testind %%%%
                            else
                                rightind = indr(ii);
                            end
                        end
                    end
                    if left > 0
                        indl= find(ismember(Linetemp(:,3:4),
points,'rows')==1);
                        for ii = 1:size(indl,1)
                            if indl(ii) == testind %%%%
                            else
                                leftind = indl(ii);
                            end
                        end
                    end

                    if rightind >0
                        points = Linetemp(rightind, 3:4);
                        indtemp(count) = rightind;
                        count = count+1;
                        testind = rightind;
                        rightind = 0;
                    else
                        points = Linetemp(leftind, 1:2);
                        indtemp(count) = leftind;
                        count = count+1;
                        testind = leftind;
                        leftind = 0;
                    end

                    if count == 5
                        sz = size(entity,1)+1;
                        for i = 1:4
```

```
                            entity{sz,1}(i,1:4) = Linetemp(indtemp(i),:);
                            entity{sz,2} = 'Rect';
                            Linetemp(indtemp(i),:)=0;
                        end
                        entity{sz,3}=[(max(entity{sz,1}(:,1))-
min(entity{sz,1}(:,1)))/2+min(entity{sz,1}(:,1)), (max(entity{sz,1}(:,2))-
min(entity{sz,1}(:,2)))/2+min(entity{sz,1}(:,2))];
                        Linetemp( ~any(Linetemp,2), : ) = [];
                        countr = countr +1;
                        indtemp =1;
                        test = 1;
                        count = 2;
                        testind = 1;
                        if sum(Linetemp)>0
                            points = Linetemp(1,1:2);
                        end
                    end

                end


        clear i ii d count countr tempind tempend tempsize ttemp temp endpt
startpt points count testind test Linetemp leftind rightind left right

        %create distance matrix
         if abs(sum(entity{1,1}))>0 && sum(size(entity,1)) > 1
             count = 1;
             dist = zeros(size(entity,1),size(entity,1));
             for i = 2:size(entity)
                 for ii = 1:count
                     dist(i,ii) = sqrt((entity{i,3}(1,1)-entity{ii,3}(1,1))^2
+(entity{i,3}(1,2)-entity{ii,3}(1,2))^2);
                 end
                 if count ~=size(entity,1)-1
                     count = count+1;
                 end
             end
             dist = dist + dist';
             f = dist(~eye(size(dist')));

         end

         %run GA
         if size(entity,1)==1
             d(jj) = 0;
         else
for i = 1:size(entity,1);
    xy(i,1:2) = entity{i,3};
end
popSize = 60;
numIter = 1e4;
showProg = 1;
showResult = 1;
dmat = dist;
[optRoute,minDist] = tspofs_ga(xy,dmat,popSize,numIter,showProg,showResult);
```



```
% close all

    startx = entity{1,3}(1,1);
    starty = entity{1,3}(1,2);
    for i = 1:size(optRoute,2)
        matrix = entity{optRoute(i),1};
        [mdist, new_startx, new_starty] = mindist(startx, starty, matrix);
        startx = new_startx;
        starty = new_starty;
        dtemp(i,1) = mdist;
    end

    distance(jj,1) = sum(dtemp);


end

    end
    name{jj,1} = data(jj).name;
%     clear c_Arc c_Cir c_Line c_Poi c_Poly dist dmat dtemp entity f matrix
mdist minDist arcpts a Line extLine new_startx new_starty numIter optRoute
popSize showProg showResult xy startx starty i ii
end
    end
end
```