DESIGN FOR ASSEMBLY AND
SCHEDULE OPTIMIZATION
FOR LARGE STRUCTURE
CONSTRUCTION

A Thesis Submitted to the College of
Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in the Department of Mechanical Engineering
University of Saskatchewan
Saskatoon

By
Harold Paul Sauder
December 1998

© Copyright Harold Paul Sauder, 1998. All rights reserved.
PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of Department or the Dean in the College in which my thesis work was done. It is understood that any copying or publication or of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Request for permission to copy or make other use of material in this thesis in whole or in part should be addressed to:

Head of the Department of Mechanical Engineering
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5A9
ABSTRACT

The efficient construction or assembly of a large structure is a critical factor in the success of a project because of the cost and time required. The construction of a large structure is long and expensive due to the number of processes and resources required, the complexity of the processes and the dependencies between tasks. Two methods have been identified for increasing construction efficiency: Design for Assembly (DFA) and schedule optimization.

To improve assembly efficiency of smaller products, DFA methodologies have been widely developed and applied. The basis of these methods is to reduce the number of parts in the assembly and to provide a quantitative measure of the assembly efficiency.

To achieve similar advantages to those DFA has provided for smaller assemblies, design for large assembly guidelines have been developed. The guidelines are the result of identifying relevant paradigms and the considerations which affect assembly efficiency. The paradigms which have been identified are concurrent engineering, simplicity and modularity. The considerations which must be addressed to improve assembly are joining, handling, positioning and assembly planning. To aid designers, the paradigms and considerations have been formulated as guidelines and organized to correspond with the design process.

Resource constrained project scheduling is an active area of research focused on reducing project duration and cost. Several optimization techniques have been investigated, including heuristics, branch and bound, genetic algorithms and constraint satisfaction.

A new algorithm for resource constrained schedule project optimization has been developed which combines genetic algorithms and constraint satisfaction. The algorithm uses a genetic algorithm to determine the priority of the tasks for resource allocation. Constraint satisfaction techniques are used to generate the schedules. The algorithm has been implemented in a computer program and demonstrated by successfully solving two problems taken from literature.
ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Peihua Gu, for the guidance, support and encouragement which he has offered during my Master’s program. Particularly, I would like to thank Dr. Gu for his vision for the value and importance of this work and ensuring that it was completed.

I would also like to thank my fellow students in the Advanced Engineering Design Laboratory for making it an enjoyable place to study and for their comments and suggestions. Thanks to Neil Schemenauer for his comments on the scheduling algorithm and help in the implementation. Thanks also to Yaw Asiedu for assistance in arranging my thesis defense.

I would like thank the University of Saskatchewan for their financial support through the graduate scholarship program. Thank you also to NSERC/AECL for their support of the design lab at the University of Saskatchewan.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PERMISSION TO USE</th>
<th>ABSTRACT</th>
<th>ACKNOWLEDGEMENTS</th>
<th>TABLE OF CONTENTS</th>
<th>LIST OF TABLES</th>
<th>LIST OF FIGURES</th>
<th>LIST OF ABBREVIATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>iv</td>
<td>vi</td>
<td>vii</td>
<td>ix</td>
</tr>
</tbody>
</table>

1. Introduction .................................................................................. 1
   1.1 Large Structure Assembly and Construction .................................. 1
   1.2 Problem Statement ......................................................................... 2
   1.3 Research Approach ........................................................................ 3
   1.4 Organization of Thesis ................................................................... 3

2. Literature Review .............................................................................. 5
   2.1 Introduction .................................................................................. 5
   2.2 Design for Assembly Review .......................................................... 5
     2.2.1 Introduction ............................................................................ 5
     2.2.2 The Boothroyd-Dewhurst Approach ....................................... 7
     2.2.3 Assembly Evaluation Method .................................................. 13
     2.2.4 Design for Simplicity ............................................................. 14
     2.2.5 Axiomatic Design Theory and Application to DFA ................... 20
     2.2.6 Axiom 1 (Independence Axiom) .............................................. 21
     2.2.7 Axiom 2 (Information Axiom) ................................................. 22
     2.2.8 Summary of DFA Review ......................................................... 24
   2.3 Review of Project Scheduling ......................................................... 25
     2.3.1 Schedule Representations ....................................................... 26
     2.3.2 Schedule Optimization Problems ............................................ 28
     2.3.3 Schedule Optimization Methods ............................................. 29
     2.3.4 Summary of Project Scheduling ............................................... 32

3. Design for Large Assembly ............................................................... 33
   3.1 Introduction .................................................................................. 33
   3.2 Design Paradigms .......................................................................... 34
     3.2.1 Concurrent Engineering ........................................................... 34
     3.2.2 Simplicity ............................................................................... 36
     3.2.3 Modularity ............................................................................. 38
   3.3 Large Structure Assembly Considerations ...................................... 42
     3.3.1 Joining .................................................................................. 42
     3.3.2 Handling ................................................................................. 52
     3.3.3 Positioning .............................................................................. 53
LIST OF TABLES

Table 2.1 Assembly process characteristics. ................................................................. 8
Table 2.2 Assembly operation elements and penalty points for DFS the method .......... 15
Table 2.3 Composition of conventional DFA methods .................................................. 24
Table 2.4 Examples of scheduling heuristics ................................................................. 29
Table 4.1 Tasks for demonstration example ................................................................. 72
Table 4.2 Task specifications for the bridge problem ...................................................... 82
Table 4.3 Number of generations required to find bridge problem optimal solution ....... 83
Table 4.4 Average number of modifications per chromosome ....................................... 86
LIST OF FIGURES

Figure 2.1 Relation between conceptual and existing DFA principles. ................................ 12
Figure 2.2 A schematic of the DFS process. ........................................................................ 15
Figure 2.3 Relation between cost and the number of parts for a product with 500 parts. .......................................................................................................................... 18
Figure 2.4 Effect of DFS on the number of parts. ................................................................. 19
Figure 2.5 Effect of DFS on assembly time........................................................................ 19
Figure 2.6 The axiomatic design domains ........................................................................ 21
Figure 2.7 Zig-zag mapping between function and physical domains. ............................... 22
Figure 2.8 Modified axiomatic design domains ................................................................ 23
Figure 2.9 The assembly space in axiomatic design ............................................................ 24
Figure 2.10 Activity on arrow method activity relation ...................................................... 27
Figure 2.11 The PDM logical relations ................................................................................ 28
Figure 3.1 The design paradigms and assembly considerations of design for large assembly ................................................................................................................................. 34
Figure 3.2 Effect of modularity on project schedule of refinery .......................................... 40
Figure 3.3 Groove welding positions .................................................................................. 49
Figure 3.4 Schematic showing joint design changes made to improve accessibility.............. 50
Figure 3.5 Single bevel and double bevel fillet welds ........................................................ 51
Figure 3.6 Space frame ........................................................................................................ 52
Figure 3.7 Tubing intersection subassembly for intersection of two skewed side diagonals with lower chord ................................................................................................. 52
Figure 3.8 Headstock-tailstock positioner ......................................................................... 54
Figure 3.9 Tilting turntable positioner ................................................................................ 55
Figure 3.10 Typical bolted beam connections ................................................................... 56
Figure 3.11 Fixture design for fixture reuse ......................................................................... 58
Figure 4.1 Schematic of a simple crossover ....................................................................... 69
Figure 4.2 Concept for combining genetic algorithm and constraint satisfaction techniques .......................................................................................................................... 71
Figure 4.3 Sample schedules for demonstration example .................................................. 72
Figure 4.4 Illustration of chromosome modification .......................................................... 75
Figure 4.5 Developed crossover method to combine the gene ordering information in the parent chromosomes. ................................................................. 77
Figure 4.6 Flowchart of the implemented algorithm. ............................................... 78
Figure 4.7 Simple five segment bridge. .................................................................. 81
Figure 4.8 Specifications for the randomly generated problem. .............................. 84
Figure 4.9 Average results for the randomly generated problem for 5 trials using a population of 200. ........................................................................ 85
LIST OF ABBREVIATIONS

AEM Assembly Evaluation Method
CPM Critical Path Method
DFA Design For Assembly
DP Design Parameter
DSP Decision Support Problem
FR Functional Requirement
GMAW Gas Metal Arc Welding
GTAW Gas Tungsten Arc Welding
OFW Oxyfuel Gas Welding
PDM Precedence Diagramming Method
PERT Program Evaluation and Review Technique
PR Process Requirements
PV Process Variable
SAW Submerged Arc Welding
SMAW Shielded Metal Arc Welding
TMNP Theoretical Minimum Number of Parts
1. Introduction

1.1 Large Structure Assembly and Construction

The assembly or construction of large structures is an expensive, long and complex process. The high cost of assembly is due to the amount of labour and the equipment required. The time required for large structure assembly is due to the number and complexity of the required operations and the tight constraints on the order in which operations can be performed. The complexity of assembly is caused by many factors, including: the amount of equipment and materials, designs which involve several engineering disciplines and the number of parts and processes involved. Since the assembly of large structures is expensive, long and complex, the efficiency of the assembly process is a critical factor in determining the success of a project.

An example of a large product where assembly is recognized as a significant factor is a nuclear power plant. Nuclear power plant construction is a particularly difficult assembly problem. The difficulty arises from the amount of labour and equipment required, the variety and complexity of processes required and the coordination of activities between many sites. The amount of activity required on site has led to the use of computer simulation analysis [O'Connor 88]. CAD programs are also used to analyze complicated and heavy lifts. Due to the cost and the technology required in fabricating nuclear power station components, components or assemblies are often produced and shipped overseas.

Nuclear power plant construction also clearly demonstrates that assembly is an important factor from the construction improvements which have been made in response to competition. Atomic Energy of Canada Limited approximates the savings of reducing
the schedule for a CANDU 9 reactor by one month at $10 million and has decreased the construction schedule compared to the CANDU 6 [AECL 96]. General Electric has developed the Advanced Boiling Water Reactor which has a seven month shorter construction time compared to standard Boiling Water Reactor construction [Valenti 95]. Lastly, the capital construction cost of the Westinghouse AP600 was reduced by 30% compared to conventional reactors through use of modularity and design simplification [Valenti 95].

1.2 Problem Statement

The goal of this research is to improve the assembly of large projects. Large structure assembly is specifically addressed since considerable work has been done in developing methods to improve smaller product assembly. The products considered in this research are those large enough to require mechanical assistance to handle components and low production volume such that either project or job shop assembly are used. Examples of products which are included are large turbines, ships, oil platforms and chemical plants.

Since the products which are included are varied, the type of assembly which is addressed is more easily identified by the following characteristics:

- Assembly requires several different processes,
- The repetition in assembly processes is relatively small,
- Handling of components requires mechanical assistance,
- The sequence of processes is complex and constrained by dependencies between processes, and
- The design of the product is complex, requiring several designers.
1.3 Research Approach

To improve the construction of large structures, two specific approaches have been selected:

- To improve assembly by creating guidelines for aiding designers in considering assembly during design, and
- To decrease construction time by developing an algorithm for schedule optimization.

To develop guidelines for large structure assembly, existing Design For Assembly (DFA) methodologies are reviewed to identify principles which can be applied to large structure assembly. The guidelines are then developed from design paradigms and the significant assembly concerns identified from the DFA review and existing methods used to improve assembly. The guidelines are then organized using the design process as an outline.

To optimize project schedules an algorithm has been developed which uses a genetic algorithm and constraint satisfaction techniques. The genetic algorithm is used to optimize the project schedules by changing the sequence in which tasks are scheduled. Constraint satisfaction techniques are used to create schedules which satisfy the precedence constraints between tasks and resource constraints. To demonstrate the algorithm, it was implemented in C++ and two test cases taken from the schedule optimization literature were used.

1.4 Organization of Thesis

This thesis is organized into the following five chapters:

1. Introduction.
2. Literature Review. Design For Assembly methods are reviewed and discussed. The background of project scheduling and developed approaches for schedule optimization are summarized.
3. Design for Large Assembly. The design for assembly guidelines and the development from design paradigms and specific assembly considerations is presented.
4. Project Schedule Optimization. The developed schedule optimization algorithm is presented and the results of the test cases discussed.

5. Conclusion.
2. Literature Review

2.1 Introduction

The purpose of this chapter is to present a review of Design For Assembly (DFA) methodologies and project scheduling algorithms to serve as a basis for the presentation of the research in the following chapters. The specific areas selected for research in this thesis are the development of design for large assembly guidelines and resource-constrained project schedule optimization. DFA methodologies have been extensively developed for small product, large lot assembly. Methodologies for addressing large assemblies is an area where significant work remains. Schedule optimization has been an area of research in management and computer science for 50 years and many scheduling methodologies have been produced. However, schedule optimization remains an active area of research, developing better algorithms which more accurately address practical scheduling problems.

2.2 Design for Assembly Review

2.2.1 Introduction

The purpose of this section is to present a review of the available DFA literature and to evaluate how DFA can be applied to large assemblies. Therefore, principles and methods will be investigated, rather than the detailed DFA data and application specific knowledge that is available.

Although there are many DFA methodologies, four DFA methods are reviewed here, which are representative and the most commonly discussed of existing methods. The first three methods reviewed are the Boothroyd-Dewhurst method, the Hitachi
Assembly Evaluation Method and the AT&T Design For Simplicity method. These methods are all conventional DFA methods which focus on the evaluation of designs by evaluating the assembly operations required for each part. Although it may seem redundant to present three similar methods, it demonstrates the importance and success of DFA methods.

The fourth method is an adaptation of Axiomatic Design Theory for evaluating assembly quality and efficiency. Rather than evaluating the ease of assembly, this method attempts to establish a direct relationship between the assembly requirements and the assembly processes. Three steps in the evolution of Axiomatic Design Theory into a DFA analysis tool are presented.

**Motivation**

The motive for reviewing these methods is to gather DFA knowledge which will assist in the development of new DFA principles and application of existing principles to large assembly applications such as power plants and oil refineries. It is anticipated that applying DFA to new, larger assembly applications will produce similar successes as existing applications. An example of advantages achieved through DFA is Ford Motor Co. which saved $1.2 billion in 1987 by applying DFA [Constance 92]. AT&T has also gathered cost reduction data which is presented in this chapter along with the Design For Simplicity method.

**Design for Assembly**

DFA is the consideration of assembly issues within the design phase. DFA is included in the general term Design For X, where X is any downstream product concern. Considering the assembly of products during the design phase is important because it is in the design phase that most of the product cost is determined. Cost is also reduced by decreasing the amount of redesign required when a product enters production. DFA methods aim to achieve cost reductions in two main ways. First unnecessary parts are identified and eliminated. The assembly operations are then examined for how they may be simplified by altering the assembly process or the parts.
2.2.2 The Boothroyd-Dewhurst Approach

The Boothroyd-Dewhurst DFA methodology is the most widely recognized and used DFA method. The method was initially developed in the late 1970's and the success of the method has inspired other life-cycle motivated design methods. The method is directed towards small, mass produced products assembled manually or automatically.

The method is general at the conceptual design level. The strength of the method is the procedure for evaluating the effects of design changes at the embodiment and detail design level. The effects are predicted from the charts using the characteristics of the parts and assembly processes.

DFA in Conceptual Design

There are two considerations included in the conceptual design phase of the Boothroyd-Dewhurst approach. The first is a set of general guidelines to assist design engineers in DFA during conceptual design. The other is a method to predict the most appropriate assembly method for the product.

To aid designers in addressing assembly during the design process, the following guidelines are given [Boothroyd 91]:

- provide assembly information to the design engineers so they are not dependent on manufacturing engineers
- encourage simple designs, since simple designs are generally lower cost due to the decreased price of parts and amount of assembly time required
- provide information which has been gathered by experienced designers to inexperienced designers in a convenient way
- establish a data base of assembly times and costs for various assembly processes.

The most important assembly decision during conceptual design is the selection of the assembly method [Boothroyd 83]. The selection of the assembly method is critical since the design requirements are dependent on the assembly processes. For the assembly method selection, Boothroyd and Dewhurst identify three basic methods: manual assembly, special-purpose machine assembly and programmable-machine assembly. The important characteristics of the assembly processes to consider are contained in Table 2.1.
Table 2.1 Assembly process characteristics [Boothroyd 83].

<table>
<thead>
<tr>
<th>Assembly Process</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual Assembly</td>
<td>tooling simple and relatively inexpensive</td>
</tr>
<tr>
<td></td>
<td>negligible downtime due to defective parts</td>
</tr>
<tr>
<td></td>
<td>cost is relatively constant and independent of product volume</td>
</tr>
<tr>
<td></td>
<td>process is flexible and adaptable</td>
</tr>
<tr>
<td>Special-Purpose Assembly</td>
<td>developed for a specific product</td>
</tr>
<tr>
<td></td>
<td>single-purpose workheads</td>
</tr>
<tr>
<td></td>
<td>parts feeders may cause downtime due to defective parts</td>
</tr>
<tr>
<td></td>
<td>expensive</td>
</tr>
<tr>
<td></td>
<td>considerable engineering development required</td>
</tr>
<tr>
<td></td>
<td>fixed rate of production</td>
</tr>
<tr>
<td>Programmable Assembly</td>
<td>general-purpose programmable workheads</td>
</tr>
<tr>
<td></td>
<td>flexibility in production volume</td>
</tr>
<tr>
<td></td>
<td>adaptable for design changes and greater product styles</td>
</tr>
<tr>
<td></td>
<td>manually loaded parts magazines</td>
</tr>
<tr>
<td></td>
<td>parts feeders are expensive</td>
</tr>
</tbody>
</table>

Although at the conceptual design stage design information is incomplete, Boothroyd and Dewhurst [Boothroyd 83] have devised a method to select the most economical assembly method based on the following six factors:

- Product volume per shift,
- Number of parts in an assembly,
- Single product vs. variety of products,
- Number of parts required for different styles of the product,
- Number of major design changes expected during product life, and
- Company policy on investment in labor saving machinery.

Using these factors the correct assembly method can be selected from a chart developed by Boothroyd and Dewhurst, which is based on mathematical models [Boothroyd 83].

**Embodiment and Detailed Design General Guidelines**

Boothroyd and Dewhurst have developed general guidelines for manual assembly. The guidelines are divided into those for part handling and those for insertion and fastening [Boothroyd 91]:

8
A. Guidelines for Part Handling

In general, for ease of part handling, a designer should attempt to:

1. Design parts that have end-to-end symmetry and rotational symmetry about the axis of insertion. If this cannot be achieved, try to design parts having the maximum possible symmetry...

2. Design parts that, in those instances in which the part cannot be made symmetric, are obviously asymmetric...

3. Provide features that will prevent jamming of parts that tend to nest or stack when stored in bulk...

4. Avoid features that will allow tangling of parts when stored in bulk...

5. Avoid parts that stick together or are slippery, delicate, flexible, very small or very large, or that are hazardous to the handler...

B. Design Guidelines for Insertion and Fastening

For ease of insertion a designer should:

1. Design so there is little or no resistance to insertion and provide chamfers to guide insertion of two mating parts. Generous clearance should be provided, but care must be taken to avoid clearances that will result in a tendency for parts to jam or hang up during insertion.

2. Standardize by using common parts, processes, and methods across all models and even across product lines to permit the use of higher volume processes that normally result in lower product costs.

3. Use pyramid assembly—provide for progressive assembly about one axis of reference. In general, it is best to assemble from above.

4. Avoid, where possible, the necessity for holding parts down to maintain their orientation during manipulation of the subassembly or during the placement of another part. If holding down is required, then try to design so that the part is secured as soon as possible after it has been in inserted.

5. Design so that a part is located before it is released. A potential source of problems in the placing of a part occurs when, because of design constraints, a part has to be released before it is positively located in the assembly. Under these circumstances, reliance is placed on the trajectory of the part being sufficiently repeatable to consistently locate it.

6. Consider when mechanical fasteners are used, the following sequence, which indicates the relative cost of different fastening processes, listed in order of increasing manual assembly cost:
   - Snap fitting
   - Plastic bending
   - Riveting
   - Screwing

7. Avoid the need to reposition the assembly in the fixture.
The guidelines provided are valuable but lack two important characteristics. First, the guidelines do not provide a quantitative evaluation of the ease of assembly of the design. Secondly, the guidelines are not ranked so that the guidelines which offer the greatest benefit are evident. These issues are addressed by the Systematic DFA Methodology.

The Systematic DFA Methodology

The goal when applying the Boothroyd and Dewhurst Systematic DFA Methodology is to increase the assembly efficiency. The assembly efficiency, \( E_{ma} \) is defined as

\[
E_{ma} = \frac{N_{\text{min}}}{t_{ma}},
\]

where \( N_{\text{min}} \) is the theoretical minimum number of parts, \( t_a \) the basic assembly time for one part, and \( t_{ma} \) is the estimated time for the complete assembly [Boothroyd 91]. The theoretical minimum number of parts is found by combining parts together which do not meet the following criteria [Boothroyd 91]:

1. The part moves relative to all other parts already assembled during the normal operating mode of the final product. (Small motions that can be accommodated by elastic hinges do not qualify.)
2. The part must be made of a different material than all other parts assembled (for insulation, isolation, vibration damping etc.).
3. The part must be separate from all other assembled parts; otherwise the assembly of parts meeting one of the above criteria would be prevented.

The manual assembly time of the product design is estimated using data tabulated by Boothroyd and Dewhurst. The data measures the effects of part and process characteristics on the assembly time for the part. Some of the part features which affect manual handling time are:

- Symmetry
- Size
- Weight
- Necessity for mechanical assistance
- Tangling
- Fragility
- Flexibility
Some of the design features that significantly affect manual insertion and fastening times are:

- Accessibility of assembly location,
- Ease of operation of assembly tool, and
- Ease of alignment and positioning during assembly.

By assessing each part by the significant features and referencing data contained in tables and graphs the estimated assembly time for each part is found. The data is based on experimental and theoretical analysis. Summing the assembly times for all parts gives the total assembly time. The assembly efficiency can then be calculated. The results indicate where design improvements can be made and compares the relative ease of assembly of competing designs.

**Comments on the Boothroyd-Dewhurst Method**

The strength of the method is the guidelines for evaluating design changes at the detailed design stage. The method is directed towards small, simple, mass produced products and does not address large products. The method could be expanded at the conceptual and embodiment design stages. The embodiment design phase requires guidelines and evaluation techniques for the system level rather than the parts level where the Systematic DFA Method are applied.

**Applying the Boothroyd-Dewhurst Principles to Conceptual Design**

The DFA principles established by Boothroyd and Dewhurst are too specific for use in conceptual design. Simpson et al [Simpson 95] have therefore suggested abstracted principles for application in conceptual design. The relation between the Boothroyd-Dewhurst and the abstracted principles is shown in Figure 2.2.
The abstracted DFA principles are explained by Simpson et al. [Simpson 95] as:

- **Relax Constraints** - A solution concept which has fewer constraints and tight fittings may be easier to assemble.
- **Proper Material Selection** - Some materials are easier to use in certain applications and some materials are better suited for manufacturing and precision equipment. Choosing the right material will facilitate assembly.
- **Minimize solution complexity** - A more complex solution concept typically requires more parts and components and thus is more difficult and costly to assemble.
- **Minimize Relative Motion of Parts** - If there is no need for relative motion between two parts, then there is no functional need to make two parts, unless they must be made of different materials. By minimizing relative motion and number of different materials, we can decrease the number of parts and hence decrease the complexity of the solution and thereby facilitate assembly.
- **Maximize Solution Symmetry** - A more symmetric concept will lend itself to having more components/parts which are the same, thus minimizing the number of parts which assembled incorrectly and reorientations during assembly.

In the paper by Simpson et al. the abstracted DFA principles are applied to an example design using Decision Support Problems (DSPs). The DSP process begins by identifying solution alternatives and the decision criteria. In this case the decision criteria are the abstracted DFA principles. The alternatives are then compared to a
datum solution using the decision criteria. The results are then analyzed for the best alternative.

Comments on the Abstracted DFA Principles for Conceptual Design

The abstracted DFA principles are an improvement over Boothroyd and Dewhurst's design guidelines for use in the conceptual and embodiment design stage. The real advantage of the abstracted principles is that they are applicable at a more abstract level where decisions about solution principles and design layout have not been made. The abstracted principles also are more specific than simple organizational guidelines and provide a basis for decision making.

2.2.3 Assembly Evaluation Method

The Assembly Evaluation Method (AEM) is among the oldest of all DFA methods. It began at Hitachi in 1976 and has been combined with Hitachi's Machining Productivity Evaluation Method to form the Producibilty Evaluation Method [Whitney 95]. Miyakawa and Onashi are the researchers associated with this method.

The method is to examine each part and, beginning with 100 points, deduct points for the undesirable characteristics of the part and the assembly operations related to the part. The parts are then redesigned until the parts cannot be improved further or the average for the parts in the assembly is above 80 [Whitney 95]. The method is based on having only "one motion for one part" during assembly [Boothroyd 94]. The method uses two design measures, the assembly-evaluation score and the assembly-cost ratio, which quantify design quality and assembly cost reduction [Miyakawa 87]. The assembly-evaluation score quantifies the assembly difficulty. The assembly-cost ratio relates the projected assembly cost to the present assembly cost [Boothroyd 94].

Hitachi claims that the AEM program is superior to the Boothroyd-Dewhurst DFA method. An important advantage is that the AEM cost reduction predictions have been verified in an industrial setting and are accurate within 15 percent [Whitney 95]. The AEM method is sold by Hitachi and the details of the program are proprietary.

Comments on the Assembly Evaluation Method

Although there is little information available about the AEM, it is evident that the strengths and faults are similar to the Boothroyd-Dewhurst method. The focus of the
program is the detail design stage and is most valuable in product redesign evaluation. The AEM does not investigate improving assembly by design at the conceptual and embodiment design stages. As mentioned before, a difference compared to the Boothroyd-Dewhurst method is that the AEM uses actual cost reduction data gathered by Hitachi. The need to withhold the cost reduction information has led to the Hitachi method being less widely known and discussed compared to the Boothroyd-Dewhurst method.

2.2.4 Design for Simplicity

The Design For Simplicity (DFS) method is a DFA method developed by AT&T Bell Laboratories. The method began from the Hitachi Assembly Evaluation method and was then refined by AT&T and General Electric [Kim 95]. The method is similar to the Boothroyd-Dewhurst method in the design evaluation stage although the scoring process is simpler. A difference from the methods previously discussed is that the managerial aspects of the program have greater development.

The basic purpose of the DFS method is to “eliminate unnecessary parts and simplify the basic assembly process” [Watson 90]. The method stresses that choosing the correct members for the DFS team is critical. The DFS team must have an understanding of the required product function, the ability to design a product which meets the requirements, and a knowledge of assembly and manufacturing processes.

The DFS Method

The DFS method provides the basis for the evaluation and improvement of the ease of assembly of the design. The method is divided into five steps. The process is shown schematically in Figure 2.2.
Measure the Design

The first step in the processes is to determine the motions and processes needed to assemble the design. This can be done with the aid of drawings, computer models and a prototype if available. Once the motions and processes have been determined, penalty points are charged for each motion and process. The only movement that is not assigned penalty points is the straight-down movement of one part onto a connecting part. Some examples of penalty points for a few assembly operations are given Table 2.2. The penalty point system used was created by General Electric.

Table 2.2 Assembly operation elements and penalty points for DFS the method [Watson 90].

<table>
<thead>
<tr>
<th>Operation</th>
<th>Penalty Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move part downward in straight line</td>
<td>0</td>
</tr>
<tr>
<td>Move part horizontally in straight line</td>
<td>20</td>
</tr>
<tr>
<td>Move part diagonally downward in straight line</td>
<td>30</td>
</tr>
<tr>
<td>Move part upward in straight line</td>
<td>30</td>
</tr>
<tr>
<td>Move part in a circular path in vertical or horizontal plane (includes turning a screw)</td>
<td>30</td>
</tr>
<tr>
<td>Maintain relative position of attaching part or part on which attachment is made; use for only one assembly operation</td>
<td>30</td>
</tr>
<tr>
<td>Join parts by staking, clinching, riveting, or mechanical press fitting</td>
<td>20</td>
</tr>
<tr>
<td>Join parts by welding</td>
<td>20</td>
</tr>
<tr>
<td>Join parts by soldering or brazing</td>
<td>30</td>
</tr>
</tbody>
</table>
The total score for the product design is then found by summing the penalty points for all the required operations. The important design measures found from this analysis are the parts count, assembly time and assembly efficiency. How the design can be improved and how competing designs compare is determined from the design measures.

**Challenge the Need for Each Part**

Once the design measures have been calculated, the design team examines the product design for parts which can be eliminated or combined with other parts. By eliminating or combining parts the product design becomes simpler, leading towards an easier to assembly design. General Electric formulated three questions to identify parts which could be combined with others. The questions apply the same principles that Boothroyd and Dewhurst use for achieving the same goal. The questions are [Watson 90]:

1. Does the part move relative to its mating part?
2. Must the part be different in material or isolated from its mating part?
3. Must the part be separate because the disassembly and reassembly for service would otherwise be impossible?

After applying the three questions to each part, a fourth design measure is determined, the theoretical minimum number of parts (TMNP). Although it is difficult to approach the TMNP, this design measure can be used to give an indication of progress during design. To put the achievable number of parts in perspective, an AT&T design involving with more then 200 parts is rarely produced with less then 150 percent of the TMNP [Watson 90].

**Iterate the Design**

Once the first two steps of the DFS method are completed, the information gathered can be used to improve the design. The major goals of the redesign are the reduction in the number of parts and the reduction of the assembly time.
In order to reduce assembly time it may be feasible to design subassemblies, which would be assembled by vendors. The important factors in deciding to purchase a subassembly from a vendor are [Watson 90]:

- Can the vendor do the subassembly faster, less expensively, or with higher quality because of its tools, technology, experience, or labor rates?
- Does the subassembly allow the vendor to perform a functional test to verify that it meets the requirements?
- Is the vendor making mating parts?

**Measure the Redesign**

After the design is modified the first two steps of the method are repeated. The four design measures, the number of parts, assembly time, assembly efficiency and TMNP are recalculated.

**Compare and Iterate with Redesigns**

To check if progress has been made it is easiest to compare one design with the previous version. The design measures provide the basis for the comparison. The product is redesigned until no cost-effective improvements can be made. The point at which design iterations are stopped is determined by the product development budget, projected volume and the optimum product design zone.

**The Optimum Design Zone**

Since design is an iterative process it may be difficult to determine when to stop. The DFS program claims there is an optimum design zone. The optimum design zone is based on features, cost, quality and schedule. Figure 2.3 shows the relationship between cost and complexity for a typical AT&T product of 500 parts. Determining the position of the optimum design zone is difficult and requires an experienced design engineer. The optimum design zone is often pushed to the right of the minimum cost point because of time to market concerns. There are four ways identified to reduce the cost increases
due to complex parts [Watson 90]:

- Reuse existing technology
- Allow for parallel design and development effort
- Freeze all designs early and do not allow changes
- Work with vendors as early as possible and let them develop as much of the complex technology as prudent.

![Diagram](image)

Figure 2.3 Relation between cost and the number of parts for a product with 500 parts [Watson 90].

**DFS Success at AT&T**

The DFS program at AT&T has been very successful. An example of the success was the 5ESS Switch Power Distribution which was presented by Kim et al [Kim 95] at the 1995 ASME Design Theory and Methodology Conference. The product is a digital
electronic system for regional telephone systems. The product was redesigned applying the DFS method resulting in:

- 68 percent fewer parts,
- 87 percent less assembly time, and
- 40 percent price reduction.

Results of applying DFS to AT&T products and those of other companies have been gathered by AT&T. The data gathered is shown in Figure 2.4 and 2.5. From the data it can be seen that a typical reduction in the number of parts is 25 to 45 percent and the reduction in assembly time for a complex product is 40 percent.

---

**Figure 2.4** Effect of DFS on the number of parts [Watson 90].

---

**Figure 2.5** Effect of DFS on assembly time [Watson 90].
Comments on Design for Simplicity

DFS is a conventional DFA method which is similar to the Boothroyd-Dewhurst and AEM in strengths and weaknesses. The interesting parts of the DFS is the effort made to determine when redesign should be stopped and the consideration given to having subassemblies supplied by vendors. Although the optimum design zone determined for AT&T products will be different than that for other companies, it is an important concept which can be applied to any product.

2.2.5 Axiomatic Design Theory and Application to DFA

Axiomatic Design Theory

Axiomatic Design Theory is based upon using design axioms to determine the acceptability of design solutions. Axiomatic Design Theory was developed by Nam Suh and is an evolved version of previous design axioms [Dimarogonas 93]. Albano and Suh [Albano 94] identify the following as the key concepts for Axiomatic Design Theory:

- Design involves the continuous processing of information between and within four distinct domains.
- Solution alternatives are created by mapping the requirements specified in one domain to a set of characteristic parameters in an adjacent domain.
- The output of each domain evolves from abstract concepts to detailed information in a top-down or hierarchical manner.
- Hierarchical decomposition in one domain cannot be performed independently of the evolving hierarchies in the other domains (i.e. decomposition follows zig-zag mapping between adjacent domains).
- Two design axioms provide a rational basis for the evaluation of proposed solution alternatives and the subsequent selection of the best alternative.

The four design domains are the consumer, functional, physical and process domain. The domains are shown in Figure 2.6. The user needs are identified in the consumer domain. The user needs are quantified into functional requirements (FRs), which exist in the functional domain. The bulk of the design work is then done in moving to the physical domain by translating the FRs into design parameters (DPs) while following the design axioms. The design process is then completed by determining the process variables (PVs) in the process domain.
2.2.6 Axiom 1 (Independence Axiom)

The first design axiom is the independence axiom which states: maintain the independence of the FRs. The effect of applying axiom 1 is that each DP has one and only one corresponding FR (i.e. no DP corresponds to two or more FRs). The decomposition of the FRs and the DPs as the hierarchies are developed is illustrated by 'zig-zag mapping' as shown in Figure 2.7. The relation expressed mathematically using matrix algebra is:

\[
\text{FR} = \text{A DP}
\]

where \( \text{FR} \) is a vector of the FRs, \( \text{DP} \) is a vector of the DPs and \( \text{A} \) is the design matrix containing the relationship between the FRs and DPs. If the design matrix is a diagonal matrix, then axiom 1 is satisfied. If the design matrix is a triangular matrix, axiom 1 may or may not be satisfied depending on the decomposition of the FRs and DPs.
2.2.7 Axiom 2 (Information Axiom)

The second design axiom is the information axiom which states: minimize the information content of the design. The information content, $I$, of a design is defined as:

$$I = \log_2(1/p)$$

where $p$ is the probability of the design meeting the FRs. The information axiom can be expressed in terms of the system capabilities and design factors.

Axiomatic Design and Concurrent Engineering

The Axiomatic Design Theory provides a basis for developing an acceptable design [Albano 94]. By applying the independence axiom to the DPs and PVs a second design matrix $B$ is found:

$$DP = B PV$$

where $PV$ is a vector of the PVs. The information axiom can also be applied using the probability that the selected PVs will satisfy the DPs.

The zig-zag mapping and decomposition techniques offer a method of controlling the concurrent engineering process. The design process then considers the highest level FR, then DP, then PV and then moves down a level. In this way the design process cannot address an FR until the previous DP and PV have been solved.

Modification of the Axiomatic Theory Design

A modification to the domains used in the axiomatic design theory has been suggested by Sohlenius [Vallhagen 94]. It is proposed that design process relates the customer, designer and manufacturing worlds. Inside the manufacturing world another
design domain is added; the process requirements (PRs). The zig-zag mapping between the designer and manufacturing is replaced by a simple transfer of the DPs. The modified axiomatic design domains is illustrated in Figure 2.8.

![Figure 2.8 Modified axiomatic design domains [Vallhagen 94].](image)

A further modification to the design domains was suggested by Vallhagen [Vallhagen 94]. Vallhagen proposes that the following spaces be introduced into the manufacturing world:

- Parts manufacturing space,
- Assembly space,
- Material handling space,
- Control and integration space, and
- Human factor space.

These spaces are then connected inside the manufacturing world and each consist of a PR and PV domains. The assembly space is shown in Figure 2.9. The design axioms can now be applied to the assembly space to evaluate possible assembly methods and processes.
Figure 2.9 The assembly space in axiomatic design [Vallhagen 94].

Comments on Axiomatic Design Theory and DFA

The most important contribution of axiomatic design in this area is to demonstrate that there are more DFA methods to be discovered. The strength of the Axiomatic Design Theory is that it provides a mathematical basis for developing a design. The problem with the strictly mathematical approach is that it will be difficult to apply to complex designs and, more specifically, to complex DFA problems. It is interesting to note that when the axiomatic design theory is evolved to develop the assembly space, it mathematically expresses the ‘one motion for one part’ principle used at Hitachi.

2.2.8 Summary of DFA Review

In conclusion, although conventional DFA methods are very effective in product redesign and are commonly used, there are opportunities for further work. The basic composition of a conventional method is shown in Table 2.3.

Table 2.3 Composition of conventional DFA methods.

<table>
<thead>
<tr>
<th>Design Stage</th>
<th>DFA Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual</td>
<td>Organizational Guidelines</td>
</tr>
<tr>
<td>Embodiment</td>
<td>General Design Guidelines</td>
</tr>
<tr>
<td>Detailed</td>
<td>Evaluation Techniques and Improvement Suggestions</td>
</tr>
</tbody>
</table>
The organizational guidelines at the conceptual design stage deal with arranging design teams which have assembly and other special knowledge and arranging assembly information for design engineers.

The guidelines at the embodiment design stage are statements of general design goals which are valid for any product. The two most common and important are designing to reduce complexity and maximize symmetry.

The strength of the conventional DFA methods is in the detailed design stage. The ease of assembly of the design is evaluated using factors tabulated on the basis of part and assembly process characteristics. The design is then iterated using the evaluation method to assess the impact of design changes.

Axiomatic Design Theory and DFA are not readily compatible concepts. Axiomatic design theory has not been formulated into a practical DFA methodology. A practical method is not likely to be produced since the theory is mathematically based and DFA is not a structured problem. The benefit of applying Axiomatic Design Theory to DFA is that it provides a different perspective.

Opportunities for Further Research

Conventional DFA methods are very developed at the detailed design stage. Therefore, the areas of conceptual and embodiment design are where significant improvement could be made. The applications for which DFA has been developed are small, mass produced products. The value of the methods applied to these smaller products is significant and it is expected that DFA methods for large, complex products will be a valuable tool in improving quality and reducing cost.

2.3 Review of Project Scheduling

Project scheduling is an effective tool for planning, studying and monitoring any project. In the past scheduling has been regarded as an art and done using simply the experience of the scheduler. Since scheduling is a complex problem, representations such as bar charts and CPM networks have been developed to aid schedulers. Based on such representations, algorithms have been developed to generate and optimize project schedules.
2.3.1 Schedule Representations

The primary representation tools which are used in scheduling are bar charts and logic diagrams. Bar charts or Gantt charts are widely used to create a visual representation of a project schedule and increase understanding of the schedule, particularly for those not directly involved in creating the schedule. Logic diagrams use arcs and nodes to represent tasks, task relations and events or milestones and contain more information which is useful for scheduling complex projects.

Logic Diagrams

There are three important methods of schedule representation using logic diagrams: Critical Path Method (CPM), Program Evaluation and Review Technique (PERT) and Precedence Diagramming Method (PDM) [Willis 86]. The scheduling method developed in this thesis is based upon the PDM to allow for greater schedule complexity, although CPM and PERT are more widely known and used.

Critical Path Method

The CPM is the oldest of the logic diagram scheduling methods, beginning development in the Integrated Engineering Control group at du Pont [O’Brien 69]. The goal of the CPM is to create a diagram which allows the scheduler to determine which tasks must be completed on schedule for the entire project to be completed on schedule. These tasks are the “critical tasks” and the series of critical tasks which link the beginning of project to completion is the “critical path.” The non-critical tasks have some float time which allows the task to be delayed without delaying the completion of the project. It is standard to calculate the early and late, start and finish times for the non-critical tasks.

The critical path method uses the activity on arrow method of representation, where activities in the project are represented by arrows and the events which relate activities are represented by nodes. Using the activity on arrow representation, the only logical relationship between tasks is that activity B can not start until activity A is completed as shown in Figure 2.10.
Program Evaluation and Review Technique

The PERT was developed in the late 1950's by the United States Navy for the Polaris program [O'Brien 69]. Like the CPM, PERT uses activity on arrow logic diagrams. The difference in the methods stems from the emphasis on events in PERT while the emphasis in CPM is on activities [Willis 86]. Another significant difference is that PERT uses probabilities to calculate "effective" task durations based on estimates of optimistic, pessimistic and mean event times. Based on the "effective" duration of the tasks, early and late event times are calculated for each event.

Precedence Diagramming Method

The PDM differs from CPM and PERT by using activity on node networks rather than activity on arrow networks. However, the PDM is similar to CPM in dealing chiefly with tasks rather than events and early and late, task start and finish times are calculated. PDM can be considered as a form of CPM [O'Brien 69]. The advantage of PDM over CPM is the ability to more clearly represent complex schedules due to the use of more relationship types, which can only be used with activity on node networks because they relate start and finish times of the tasks. The representation of the relations are shown in Figure 2.11.
2.3.2 Schedule Optimization Problems

There have been many solution methods created for a variety of resource-constrained project scheduling problem formulations. A comprehensive review of methods and problem formulations has been produced by Özdamar and Ulusoy [Özdamar 95].

The most significant characteristic of the problem formulation is the objective which is to be optimized. The objectives which are used are either time related, or cost related [Özdamar 95]. The most common objective is the minimization of project duration. Cost based objectives include maximizing the net present value of a project\(^1\) and minimizing the total cost of the project.

Another important characteristic of problem formulation is how tasks and resource usage are constrained. Generally tasks are constrained using the finish-to-start relation. The set of precedence diagramming relations are also used, referred to in schedule optimization literature as generalized precedence relations [Elmaghraby 92], time windows [Bartusch 88] and minimal and maximal time lags [Özdamar 1995]. The constraints on the resource are either renewable, non-renewable or doubly constrained [Özdamar 95]. Renewable resources are limited on period by period basis. Non-renewable resources are limited on a project basis. Doubly constrained resources are limited on a period by period and on a project basis.

\(^1\) To optimize the net present value of a project the problem model must include cash flows related with the activities and then optimize the schedule to increase the value of the project throughout the schedule.
2.3.3 Schedule Optimization Methods

There are several different algorithms or techniques which have been used for the basis of project schedule optimization methods. The oldest method is the use of heuristics to make decisions on resource allocation. There has been research into branch-and-bound algorithms with sophisticated rules for limiting the search space. The most recent methods developed involve the use of constraint logic programming and genetic algorithms.

Heuristic Project Scheduling Optimization

Heuristic project scheduling uses rules to make resource allocation decisions as the schedule is generated. Scheduling heuristics are applied by beginning at the start of the schedule and then scheduling tasks using the heuristics to allocate resources.

The use of heuristics in scheduling problems is popular due to the complexity in solving arithmetical models and has resulted in “hundreds of heuristic-based procedures” [Davis 75]. There are different bases for creating heuristic rules. Olaguibel and Goerlich [Olaguibel 89] categorize the bases which have been used to develop heuristics as activity, network, CPM, resource based rules and combinations. Some examples of heuristics used in scheduling are contained in Table 2.4. Although heuristic algorithms have been widely investigated and developed, they do not always find the optimal schedule which has lead to further research in schedule optimization.

Table 2.4 Examples of scheduling heuristics.

<table>
<thead>
<tr>
<th>Heuristic Name</th>
<th>Highest Priority Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Total Successors [Neumann 95]</td>
<td>Task with most successor tasks.</td>
</tr>
<tr>
<td>Latest Start Time [Neumann 95]</td>
<td>Task with the lowest latest start time.</td>
</tr>
<tr>
<td>Minimum Job Slack [Davis 75]</td>
<td>Task which has the least slack.</td>
</tr>
<tr>
<td>Greatest Resource Usage [Davis 75]</td>
<td>Task which results in maximum resource usage</td>
</tr>
</tbody>
</table>
Branch and Bound Schedule Optimization

The schedule optimization problem space can be searched to find the optimal schedule using the branch and bound search method. The branch and bound algorithm creates partial algorithms and then

- Expands partial solutions to find the optimal solution, and
- Eliminates partial solutions from the solution space which will not lead to the optimal solution.

To create a branch and bound procedure, branching and pruning methods are required for searching the solution tree. The branching method selects the variable to label for expanding the partial solution. The bounding or pruning method determines if a partial solution cannot lead to an optimal solution and the partial solution is no longer considered for expansion.

A widely discussed branch and bound algorithm has been produced by Stinson [Stinson 78]. In the algorithm the selection of decision choices or branching of the solution tree is made using several heuristics. A series of heuristics is used to break the ties which may exist according to a single heuristic. Using heuristics to control the branching of the solution tree is done so that good solutions will be obtained quickly, allowing more effective pruning of the solution tree. The algorithm uses dominance and lower bound pruning rules. The dominance rule states that if a partial solution contains an activity that could be scheduled earlier in time without violating any constraints, the partial schedule is dominated and can be pruned from the solution tree.

The lower bound pruning uses an optimistic estimate of the entire schedule length based on the partial schedule. If the optimistic estimate is longer then the shortest completed schedule, or the upper bound, the partial schedule is pruned from the solution tree. Stinson [Stinson 78] uses three lower bound estimates. The first finds the optimal schedule ignoring the resource constraints on the unscheduled activities. The second ignores the resource constraints on the activities. The third finds the completed schedule ignoring the unscheduled tasks which do not lie on the critical path to project completion. The longest of the three estimates is used as the lower bound of the partial schedule.

30
The limitation of branch and bound methods is due to the combinatorial nature of the scheduling problem [Ulusoy 94]. As the size of the scheduling problem increases, the solution space problem becomes too large and problems with computation time and memory requirements arise. Due to these constraints branch and bound methods are effectively limited to problems which contain 50 activities and 3 resources or less [Neumann 95].

**Constraint Logic and Schedule Optimization**

Constraint logic programming or constraint satisfaction techniques have been introduced into scheduling algorithms to deal with the complexity of resource and precedence constraints. Constraint logic programming reduces the problem space to be searched by enforcing constraints on variables before the variables are assigned values and through propagation of constraints as variables are labeled [Lever 95]. Although constraint logic is not an optimization routine in itself, it has been successfully combined with heuristic\(^1\) and branch and bound\(^2\) approaches to improve algorithm performance.

**Genetic Algorithms and Schedule Optimization**

Genetic algorithms are well suited to solving for combinatorial optimization problems and have been used for project schedule optimization. Genetic algorithms work by encoding problem solutions into chromosomes which are strings of values. Solutions are then combined and evaluated to search for the optimal solution.

The problem which needs to be considered in applying genetic algorithms is how to encode the schedule solution into the chromosome. A direct method is used by Cheng and Gen [Cheng 94] where the schedule solution is stored in the chromosome using the start times for the tasks. However, directly encoding the solution as start times creates difficulty when trying to generate schedules due to violating constraints. To avoid this difficulty it is possible to store solutions as the priority of activity during schedule creation. Chan et al [Chan 96] store the priorities for tasks as floating point numbers.

---

1 See Ulusoy and Özdamar [Ulusoy 94] and Crabtree [Crabtree 95].
2 See Dincbas et al [Dincbas 90] and Lever et al [Lever 95].
Pet-Edwards and Mollaghasemi [Pet-Edwards 95] have created an approach which determines scheduling priorities using the sum of six heuristics which are multiplied by weights determined using a genetic algorithm. Although the genetic algorithm based scheduling methods have been demonstrated to be successful, there are opportunities for research in this relatively new area.

2.3.4 Summary of Project Scheduling

During the late 1950's CPM and PERT were developed to create and represent project schedules. As a descendent of CPM, PDM was developed to represent schedules with complicated constraints between tasks. Although these techniques are widely used they can be made more effective by including resource considerations. The generation and optimization of schedules under resource constraints is a numerically complex problem and algorithms are needed to find solutions. Among the methods used as the basis for such algorithms are heuristics, branch and bound procedures, constraint satisfaction and genetic algorithms. Although project scheduling has been extensively investigated it is still an area of active research and progress can be made.
3. Design for Large Assembly

3.1 Introduction

This chapter presents guidelines developed to aid designers in the consideration of assembly during the design of large assemblies. The assemblies considered are one of a kind or low volume, complex projects compared to the mass production addressed by existing DFA methods. There is a need for a separate design for large assembly methodology because the assembly considerations required are different from those in smaller designs. An important difference is caused by the smaller production run which is typical of large products. When large production runs are possible, small refinements in the assembly process are economically feasible. In large projects the assembly process is complex and involves little process repetition. Therefore, in a design for large assembly methodology the focus needs to be on simplifying the assembly processes and the general design features which affect assembly.

The design for large assembly method proposed is comprised of design paradigms, assembly considerations and design guidelines. The design guidelines are the practical result of identifying relevant design paradigms and design features. The relation of paradigms, assembly considerations and guidelines is shown in Figure 3.1.

Basic improvements in assembly can be achieved by altering the goals and processes of design through the implementation of design paradigms. The use of design paradigms will focus the design effort so that improvements in constructability are made without directly dictating the technical aspects of the design and reduce the redesign required during assembly.
Design Paradigms

Concurrency
Simplicity
Modularity

Assembly Considerations

Joining
Handling
Positioning
Assembly Planning

Design for Large Assembly Guidelines

Figure 3.1 The design paradigms and assembly considerations of design for large assembly.

Identifying the assembly considerations or design decisions which influence how easily a design can be assembled is crucial for the success of a DFA methodology. The identification of assembly considerations was done by examining some common processes involved in assembling a large project, which often includes a variety of interacting processes. From the identification of the assembly considerations, the general strategies which are used to improve assembly were examined. Finally, guidelines for improving large structure assembly were developed. The guidelines, based on the paradigms and assembly knowledge, serve as the tool to implement the design for large assembly methodology.

3.2 Design Paradigms

The design for large assembly paradigm is a combination of existing design paradigms. The paradigms which have been identified as important for improving assembly are concurrent engineering, simplicity and modularity. The implementation of these three paradigms are tightly related due to the common goals of the paradigms.

3.2.1 Concurrent Engineering

Concurrent engineering and DFA are inseparable concepts. Concurrent engineering practice addresses life-cycle issues during the design process and DFA
practice considers assembly during design. Concurrent engineering is difficult to implement on large projects since design is divided between engineering disciplines and different organizations are often responsible for the project in different parts of the product life-cycle. Therefore, there are two major objectives:

1. Force designers to consider life-cycle issues, and
2. Ensure there is adequate interaction between design departments.

To improve assembly through concurrent engineering, the assembly processes needs to be developed and evaluated during the design phase. To develop and understand assembly processes, designers require access to assembly information and knowledge. Typically, assembly information is provided by giving designers access to assembly and construction planning information in databases and by including experts, such as manufacturing engineers and construction managers, in design teams. The advantage of integrating those responsible for the downstream realization of the project into the design process is realized in reduced redesign and production delays.

Implementing concurrent engineering in large product design requires designers from several disciplines to interact and cooperate to identify and solve configuration and assembly sequencing problems. The design disciplines which may be involved in large assembly engineering include mechanical, civil, electrical and chemical engineers involved in traditionally divided design problems. If design teams are formed by design area and proceed in isolation, design rework will be required to resolve assembly conflicts which may not be detected until assembly is underway. The benefit of implementing concurrent engineering through assembly consideration and design coordination is a simpler and more efficient design and assembly process.

**Implementing Concurrent Engineering**

To implement concurrent engineering, it is important that assembly related specifications are examined from the beginning of the project. The assembly specifications serve as the basis of evaluating design decisions for the designers. To estimate the assembly parameters for a unique project may be difficult. However, it is important to involve engineers with assembly and design knowledge in creating the
specifications so that the assembly of the product developed will be feasible. The specifications which influence the assembly related design decisions are:

- Production volume,
- Required assembly time or period,
- Projected cost for assembly,
- Assembly method to be implemented, and
- Modularity requirements.

The production volume is a significant parameter in determining the level of investment which can be made in assembly process development, assembly modeling and automation. Considerations of cost and production time are determining factors in product feasibility and since assembly constitutes a large portion of both, assembly cost and time need to be carefully examined from the start of the design.

For most designs the assembly method would be determined during the design of the product. In some cases however, a familiar product may be redesigned to take advantage of a particular assembly method. In these cases it is important that the proposed assembly method be studied so that specifications are compatible with the assembly process selected. A common example is the redesign of a casting as a weldment for which it is recommended the part be redesigned to suit the welding process [Connor 87].

3.2.2 Simplicity

The most important design goal which will produce an easily assembled design is simplicity. As was observed by Albert Einstien, "the best design is the simplest one that works." An easily assembled product is typically simple in both the design and the assembly processes. Evidence of the importance of simplicity is contained in existing DFA methods which encourage simplicity by the use of the scoring systems that penalize complex insertion and fastening processes. AT&T recognized the importance of the paradigm and named the DFA methodology developed at AT&T, "Design for Simplicity."
For large assemblies, simplification of design and assembly is achieved in two main ways:

- Reduction in the number of parts, and
- Standardization of components and processes.

Part Number Reduction

In the DFA guidelines developed by Boothroyd and Dewhurst, the primary method of improving assembly efficiency is the reduction in the number of parts. By reducing the number of parts in a design the end result is often a simpler design and faster assembly due to the reduced number of assembly operations. Boothroyd and Dewhurst [Boothroyd 91] propose the following guidelines to establish which parts are necessary and not targeted for elimination:

1. The part moves relative to all other parts already assembled during the normal operating mode of the final product. (Small motions that can be accommodated by elastic hinges do not qualify.)
2. The part must be made of a different material than all other parts assembled (for insulation, isolation, vibration damping etc.).
3. The part must be separate from all other assembled parts; otherwise the assembly of parts meeting one of the above criteria would be prevented.

Although the elimination of all parts which do not meet the guidelines is difficult and economically unfeasible, the guidelines are a valuable tool for driving the simplification of a design.

There are many examples of how the number of parts can be reduced in large assemblies. A simple example of part elimination is the bending of flange in a plate rather than welding a flange to a plate. Another example is the redesign of a structure which includes many braces to incorporate heavier main structural members and fewer braces. An example of the effectiveness of designing for simplicity is the Westinghouse AP600 pressurized water reactor. The nuclear ‘island’ of the reactor was redesigned to use “50 percent fewer valves, 80 percent less piping, and 70 percent less cable than the conventional midsize nuclear power plant” reducing capital costs 30 percent [Valenti 95].
Standardization

Simplicity can be increased in a design by standardizing the components and assembly processes. Standardizing components reduces the design effort. The repetition of assembly processes increases familiarity with the processes which leads to increases in efficiency. Standardization of assembly processes also makes examining the assembly process more economically feasible since any benefits realized will be multiplied. An example of repetitive process analysis is the CYCLONE (CYCLic Operations NEtwork) modeling system developed for use on construction projects [Halpin 92].

In many applications economic advantages can be achieved by limiting the number of different joining methods used. The economic advantage is realized primarily by making the maintenance and operation of the equipment simpler since staff gains greater experience with the equipment. There is also an advantage in that design and manufacturing engineers gain a greater understanding of how the process can be efficiently used. An example of designer familiarity is understanding what tolerances are required for welded joints. An example of standardizing the joining process is found at Flexi-Coil, a local producer of farm implements, which uses Gas Metal Arc Welding for nearly all of the welded joints it produces [Gerien 95].

3.2.3 Modularity

Modularity is a concept which provides assembly advantages in a several areas. The basis of modularity is to divide a design into functional groupings which have the least possible interaction between the modules. The functional independence and reduction of interactions allows the modules to be easily added, removed and substituted. One of the primary purposes for dividing a product into modules is to create a product in which the modules can be easily assembled. Some other common purposes for implementing modularity include creating product variety using combinations of modules and improving reusability or maintainability.

The major advantage of using modules regarding assembly is that modules can be assembled under more productive conditions due to the size of a module compared to the entire product. Also, once the project has been sectioned into modules, parallel development of the project is possible, speeding both design and assembly. When
designing a product it is important to carefully consider the advantages and disadvantages of modularity and identify the extra considerations which need to be taken into account.

The modularity paradigm is related to the two previously discussed design paradigms, simplicity and concurrent engineering, since an effective modular design requires a concurrent engineering approach and results in a simplified design and assembly process. To determine the modules effectively, the modules must be determined by a team who is able to understand all the functional requirements of the modules and how modules can be assembled together. The consideration of the product design while determining the module boundaries creates simplicity by detecting opportunities for using integrated equipment and produces a simplified equipment layout [Marcin 82].

Modular Assembly Advantages and Disadvantages

Modularity improves the ease of assembly by reducing the size and complexity of the assemblies. For large assemblies greater efficiency is usually achieved by producing modules in assembly yards instead of on-site. The initiative for beginning to design plants using modularity was to avoid expensive labor at remote locations and interruptions in assembly due to weather problems. Assembling modules off-site also provides access to more highly skilled labor and specialized equipment since the assembly yard can remain in continual production of modules for varied projects. Modularity also improves consistency in productivity, reducing the project risk and allowing a shortened schedule. The assembly of modules in assembly yards also speeds on-site work by reducing interference between on-site activities. An example of a reduced project schedule for a refinery due to modularization is shown in Figure 3.2.
Figure 3.2 Effect of modularity on project schedule of refinery [Glaser 83].

The decrease in the physical size of the assemblies due to modularity and reduced assembly complexity allows for greater access to all components, providing for more efficient performance of joints and inspection. The amount of work done on large assemblies at dangerous heights is reduced by decreasing assembly size. As the effort continues to be made to automate construction, the design of modules with the joint accessibility, size and simplicity required for automatic assembly provides the most feasible method [Perkowski 88].

When constructing a plant, moving work off-site also allows the work which must remain on-site to be simplified [Whittaker 84]. Performing assembly off-site removes equipment and raw materials from the actual plant site, making the coordination of onsite activities simpler. Testing of the modules can also be conducted off-site, reducing the amount of testing required and testing equipment on-site [Stubbs 90].

The primary disadvantage of modularization is the increased cost of handling and assembly process management. The most evident increased cost is due to the requirement for equipment to manipulate the modules and the cost of transportation. The assembly process also requires greater management to coordinate the assembly of modules and delivery of materials at several assembly locations.

The other significant disadvantage of employing modularity is the increased engineering effort required for design. The design of the product increases in difficulty due to the added requirements on the design. Such requirements include the
determination of the modules or grouping of components, and the transportation and positioning of the modules. A common transportation consideration is the addition of steelwork to accommodate the forces experienced by the modules during transportation. Although the engineering effort will be greater, the design time may be reduced due to the parallel design of modules and the reuse of module designs.

Implementation of Modularity

Modules are determined by separating the components of the design into functional groups. The goal in creating the modules can be described as [Ulrich 91]

- Similarity between the physical and functional structure of the design, and
- Minimizing unnecessary interactions between modules.

Although these goals are not directly related to assembly, the result of the focus on modular design results in a simpler design. From the design of chemical process plants, Marcin and Schulte [Marcin 82] recognize that, “It is through this focus that simplification opportunities in plant hardware and physical configuration can be recognized, identified, evaluated and implemented.”

The determination of modules is a difficult process since there are different possible modular groupings for the functional units which offer advantages for different design goals [Gu 97]. As the design becomes more detailed, the evaluation of how the modular groupings will affect the assembly efficiency can be completed. However, it is important to establish some goals during conceptual design, such as flexible module assembly sequences, module designs which allow testing before final installation and the ease of joining modules together.

Creating functional groups is a complex problem requiring careful examination for a problem with many possible solutions. A method for determining modules based on design goals has been developed by Gu et al [Gu 97]. The method requires an analysis of the importance of parts being placed in the same module using a broad variety of factors which are relevant to the design goals. The interaction information is used as the evaluation criteria with a genetic algorithm to generate optimal module arrangements.
For large construction projects, there may be a decision to pursue a modular design from the beginning of the project. In this case, the requirements and the constraints on the modules need to be determined before the design is divided into modules, so that the arrangement of modules will be optimal. The parameters which need to be determined for the modules are chiefly the maximum size and weight of the individual modules. These module parameters depend on the positioning equipment and the transportation method, equipment and routes available and the cost of using each. An example of such a decision was the transportation of modules, by either truck, rail or barge, for a oil processing facility in Prudhoe Bay, Alaska from the U.S. mainland [Bass 82]. The decision was made to use barge transport so that larger modules could be used even though barge transport was limited to six weeks per year due to ice flows.

3.3 Large Structure Assembly Considerations

The purpose of investigating assembly considerations is to identify the important, general processes which have significant influence on the ease of assembly in large complex projects. Once the important assembly considerations are identified, the general strategies for improving assembly and the characteristics of easily assembled designs can be discussed and evaluated for use in design for large assembly guidelines.

The general assembly considerations involved in large products are joining, handling, positioning and assembly planning. Joining considerations include the locations and processes selected. Handling includes the transportation of materials, assemblies and modules. Positioning refers to the processes to install parts or assemblies correctly. Assembly planning requires examination since it greatly affects the success of the other considerations and determines the production schedule and efficiency. To achieve maximum benefit from a design for large assembly program, all the areas need to be included: joining, handling, positioning and planning.

3.3.1 Joining

The selection and design of joining processes is a critical factor in determining assembly efficiency. Joining processes are typically carefully considered due to the amount of analysis required on joints and the complexity of the analysis. Joining
processes also need to be analyzed carefully with respect to the cost and difficulty in creating the joint. For example, joining process selection and joint location are often the largest variable cost in erecting steel structures [Parker 84]. The joining considerations which significantly affect the assemblability of large products are:

- Joining process selection, and
- Joint layout and design.

**Joining Processes Selection**

Selecting the joining process to use requires both considering the characteristics, the available joining methods and the level of automation that will be most economically advantageous. Common joining methods include:

- Threaded fasteners,
- Rivets,
- Welding, and
- Adhesives.

Although use of automation is often limited on large assemblies due to the assembly environment, if use of automation is considered during design it may offer advantages.

**Joining Methods**

*Threaded Fasteners*

Threaded fasteners are a very common and popular method of fastening, primarily due to ease of use and the simple tooling required for installation. The simplicity of threaded fasteners produces savings in labor costs, since the application of threaded fasteners requires less operator skill than riveting or welding. In many structural applications, threaded fasteners have replaced rivets also because of the ease of use and superior joint performance compared to rivets. Threaded fasteners allows products to be rapidly and simply assembled in the field. A simple example is the joining of legs to a table by threaded fasteners after shipping, giving significant cost advantages in shipping.
Another important advantage of threaded fasteners is that parts can be easily removed for maintenance or repair of the product. Also, compared to rivets, the tooling for threaded fasteners allows them to be installed in tighter spaces. A disadvantage of threaded fasteners is that holes are required for insertion, which create stress concentrations.

*Rivets*

In the past rivets were the primary method of fastening in structural applications. However, threaded fasteners and welding have nearly eliminated rivets in large structural applications. The problems with riveting in large applications were the large amount of labor required along with the excessive noise of impact riveting. As mentioned for threaded fasteners, the holes needed for applying rivets cause stress concentrations.

The advantage of rivets compared to threaded fasteners is that their simple shapes are more easily produced and the material used for rivets is usually cheaper than the material used for bolts. Also the insertion of rivets is more easily automated than the insertion of threaded fasteners and greater installation rates are achieved. Automation is difficult in large assemblies since access to both sides is required.

*Welding*

Welding includes a variety of methods of fusing components together and has become versatile in the joints which can be created and the materials which can be joined. Welding has replaced bolting and riveting in many structural applications due to the joint performance offered and the speed which joints can be created with improved welding technology. Welding is particularly suitable when the environment can be controlled.

Welding also has been used very effectively in designs which replace large castings. Cary [Cary 89] lists the following reasons for the superiority of welding:

1. The weldment is normally lighter in weight than cast or mechanically fastened structures; thus it requires less material.
2. The weldment design can be readily and economically modified to meet changing product requirements.
3. The production time for a weldment is usually less than that of other manufacturing methods. (This is a hidden cost savings benefit.)
4. The weldment will be more accurate with respect to dimensional tolerances than a casting (another hidden cost savings advantage).

5. Weldments are more easily machined than castings.

6. Weldments are tight and leak proof and will not shift or give like riveted structures.

7. The capital investment for producing a weldment is much lower than that for producing castings. Additionally, environmental controls are more easily adopted to the welding shop then to the foundry.

8. Weldments can be more pleasing to the eye than castings. They are cleaner in their lines and usually smoother and easily prepared for final use.

There are many different welding processes which have advantages in different applications. The selection of a welding process is based on the materials being joined, the joint size and location, economics and the circumstances of the assembly environment. Some common methods for general use and large structure assembly are Shielded Metal Arc Welding (SMAW), Gas Metal Arc Welding (GMAW), Gas Tungsten Arc Welding (GTAW), Submerged Arc Welding (SAW) and Oxyfuel Gas Welding (OFW). These processes are compared in the following sections (as described by the Welding Handbook [Connor 87]).

Shielded Metal Arc Welding

SMAW is performed with electrodes coated with flux, which provides a protective atmosphere for the weld pool. The electrode is manually manipulated by the welder, which slows the process considerably. The advantages of SMAW are the simple, low cost equipment and that many applications can be handled with a small variety of electrodes. The equipment is also easily portable, which is advantageous for maintenance and repair applications. Disadvantages of the process are that slag must be removed between each pass and the inefficient use of electrodes.

Gas Metal Arc Welding

GMAW uses continuously fed electrodes and externally supplied shielding gas. The major advantages of GMAW as compared to SMAW is that GMAW is easier to use, the deposition rates are faster and the continuous feeding of electrodes increases welder speed and efficiency.
Gas Tungsten Arc Welding

GTAW uses a non-consumable Tungsten electrode with externally supplied shielding gas with optional use of a filler material. The advantages of GTAW are that it can be used for a wider range of joint thickness, for more complex geometries and in more positions than GMAW or SMAW. The disadvantages are that it requires more welder training time, manual dexterity and welder coordination than other processes.

Submerged Arc Welding

SAW welding uses a continuously fed electrode which is submerged in a granular flux which provides a protective atmosphere. SAW is either used as a mechanized or a semi-automated process. The advantage of the process is that a high current can be used to obtain high deposition rates along with consistent weld quality. The process is limited since it can be used only in the flat and horizontal welding positions. In general, when SAW can be used it is among the least expensive processes.

Oxyfuel Gas Welding

OFW uses flames from the burning of oxygen and another fuel, usually acetylene, to melt base and filler material. The flame can be used to provide a protective atmosphere for the weld pool. The main advantages of OFW is that the equipment used is very simple and easily portable. The process is very operator dependent and highly skilled labor is required for critical welding jobs. The greatest disadvantage of OFW is that it is too slow to be practical for joining thick plates or for use on long production runs.

Adhesives

Adhesives are used for joining assemblies for a variety of reasons. Primarily, adhesives provide a lightweight method of joining, which is especially good for use with thin sections or fragile materials. The advantage of adhesives in thin sections is that the entire surface area of the components in the joint bear load, avoiding the stress concentrations which would result using other joining methods. The use of adhesives allows the creation of layered materials, such as plywood and the honey comb materials which are widely used in the aerospace industry. Adhesives often are the best method
for joining materials which are generally difficult to join, such as joining glass and rubber.

Adhesives may be used in conjunction with other joining process. Weldbonding is the term given to the combination of adhesives and resistance spot welding. The advantage realized is greater fatigue resistance than spot welding alone and the adhesive seals the joint, preventing crevice corrosion. The need to seal joints has caused adhesives to be used along with mechanical fasteners to produce sealed joints. Adhesives are also used to secure threaded fasteners in position which is particularly appropriate where fasteners may be loosened due to vibration.

There are a number of factors to consider when selecting adhesives as outlined by W. A. Lees in Adhesives in Engineering Design [Lees 84]. The considerations include the materials being joined, the joint configuration and function. The effects of other operations which need to be considered, such as variations in temperature and stress in the joint before curing has completed. In the application of the adhesives the preparation of the mating surfaces and curing of the joint are crucial. The method of the holding the joint materials in position during the curing of the adhesive needs to be provided. Some options are clamps, fixtures and the other joining methods.

**Joining Automation**

Automation can often offer significant cost savings due to the increased speed of production and the improved consistency of the work. Automation naturally offers greater advantages where large joints, many joints or long production runs are required. Using either threaded fasteners, rivets or welding, further increases in speed can be achieved by using multiple tool heads. An example of the efficiency increase which can be produced is the welding of rub bars to barges reported by Vogt et al [Vogt 82]. The attachment of the 162 ft long bars requires a \( \frac{1}{4} \) inch fillet weld on both sides. The switch to a tractor mounted SAW welder with two welding heads reduced the labor required by 63% compared to a manual SMAW process.

The greatest concern when designing for automatic joining is that greater clearances are required than for manual joining. An example of designing for automatic joining is the arrangement of piping networks so that efficient orbital pipe welders can
be used. Reducing the complexity of joints also allows for easier automation. For example automated welding machines, which run on rails attached to assemblies, can be used to efficiently produce long straight welds.

Joint Layout and Design

As the embodiment design proceeds, the details of the joint locations and specifications are developed. The greatest concern in improving assembly efficiency is adequate access to the joint location so that quality joints can be created efficiently. Significant improvement can also be achieved by using subassemblies and joint optimization.

Joint Accessibility

During assembly it is critical that the joints to be created can be adequately accessed by the tooling required to perform the joints. For example, trying to create joints in tight corners is difficult and will result in imperfect joints. If joint access cannot be improved in a difficult situation, the process may need to be changed to insure that a joint of sufficient quality and consistency is efficiently produced. An instance were this type of alteration of joints is made is replacing rivets with threaded fasteners, which are easily installed.

When using welding as the joining method, the efficiency of the process is heavily dependent on the welding position used. The design of a weldment should be examined to ensure that wherever possible vertical and overhead welds are replaced by horizontal or flat welds. The welding positions are shown in Figure 3.3. If the welding joints cannot be altered to an efficient position, the use of positioners to reorient the assembly should be considered and implemented depending on the size and number of joints to be performed.
Figure 3.3 Groove welding positions.

In some instances, ease of assembly can be improved by altering the type of joint used from one which requires access from both sides to a joint which requires access from only one side of the joint to perform. Examples of single-sided joints are blind rivets and single groove welds. An example of improving joint accessibility by the removal of double sided joints is shown in Figure 3.4. The assembly shown is a large turbine component which consists of several half-ring pieces inside a cylindrical shell.
Figure 3.4 Schematic showing joint design changes made to improve accessibility [Maloney 95].

The accessibility of joint locations is heavily dependent on the assembly sequence. By altering the sequence of assembly processes the equipment, processes and schedule which are possible are altered by the change in accessibility. An example of altering assembly sequences is the General Dynamics Electric Boat Division production of the Seawolf submarine [Mraz 93]. The traditional method had been to construct the hull and then install the equipment inside. The improved method was to assemble equipment in modules and then slide the assemblies inside the hull, cutting costs up to 70%. The increased accessibility allowed a greater number of workers to operate simultaneously reducing construction time, improved quality control, provided easier installation in difficult locations and allowed installation of long lead time components later in the schedule.

Joint Optimization

A common method of reducing the time required to perform joints is joint redesign or joint design optimization. Joints can be optimized by changing the joint parameters or altering the process which is used. An example of altering the joint parameters used is the modification of a joint from a single bevel fillet weld to a double bevel fillet weld. The two types of joint are shown in Figure 3.5. The double bevel fillet weld requires less than 60 percent of the deposited weld material compared to the single bevel joint [Coxhill 82].
In some cases, improving the ease with which the joint is created can be achieved by combining joining methods. For example, GMAW may be used to lay a root weld in a large joint and then use SMAW, which has a greater deposition rate, to complete the joint. When using adhesives as the primary method of creating a joint, mechanical fasteners or welds may be used to maintain the position of the components during curing of the adhesive.

**Subassemblies**

An effective way of improving the layout and design joints is to use subassemblies. Subassemblies simplify and permit the use of a more efficient joining process. An example of significant redesign to simplify assembly, is the design of the space frame [Edwards 82] shown in Figure 3.6. The frame is constructed of tubing which is welded together. Since the joints where the tubes intersect are complex, joint subassemblies were fabricated as shown in Figure 3.7 in a shop so that the joints made on-site were less difficult.
3.3.2 Handling

In the assembly of any product handling of components and materials is an important factor in the assembly efficiency. Handling can be a significant problem large assemblies which require the use of particularly heavy duty cranes, large positioners or other equipment which is expensive to use and difficult to obtain.

The simplest method of improving handling is the addition of features to components and assemblies for use during handling. The addition of handling features involves planning the method of handling that will be used. An example of including features for handling is specifying eyes or lugs for lifting which “almost always saves field time and cost” [Oglesby 89]. In some cases where large assemblies must be
handled, temporary bracing [Bolt 82] or extra steelwork [Whittaker 84] may be required. Finally, when packaging is required for transporting components, efficiency and cost reductions for the packaging should be considered [Oglesby 89].

**Handling Modules and Subassemblies**

Some handling factors to consider when designing modules are the strength of the modules and the location of the center of gravity. The required strength for modules is dependent on the forces applied during transport and positioning. For ocean transport, the module will be exposed to the rolling and flexing of the transporting barge [de Vilder 82]. Positioning modules by crane can be avoided by using skids or jacks, which have the advantage of supporting the module from below reducing the need for extra steelwork. If cranes are used to place the modules, the modules may be damaged during lifting and the design and use of lifting frames be necessary. The center of gravity is a design concern because it determines how stable the module will be during handling and positioning. If the center of gravity is not suitable for lifting and placing the module, a lifting frame which incorporates balancing tanks or blocks may be used [Bolt 82].

### 3.3.3 Positioning

Positioning requires consideration during design due to the complexity, importance and cost of the positioning processes. Positioning of large components with tight tolerances is a difficult process which may require coordination between pieces of equipment or the design of custom fixtures. An example of the importance of positioning is the alignment of structural steel, which is exceedingly expensive to correct misalignments in, once work has progressed to another area [Parker 84]. To ease positioning of assemblies the following strategies should be considered:

- Use of positioners,
- Aiding positioning through joint design, and
- Use of fixtures.

**Positioners**

In some large assemblies, production can be made more efficient by reorienting the assembly while creating joints. During product design it is important to consider what kinds of positioners are available and which positioners will be required. The most
common positioners rotate the work piece about a single axis, such as the headstock-tailstock positioner shown in Figure 3.8. A typical use of a headstock-tailstock positioner is the completion of welds in a frame of tubular members which has been tack welded together, then is placed in the positioner and rotated so that all welds are easily performed. Rotating turntables are also widely used to allow workers and equipment to remain stationary, speeding up work.

Figure 3.8 Headstock-tailstock positioner [Connor 87].

Tilting turntables, such as the one shown in Figure 3.9, are also common and can rotate the assembly about two axes. A typical use would be to position an assembly so that vertical or overhead welds become flat welds.
In large scale assemblies it is important to also consider what equipment is required for the workers to reach the joints rather than repositioning the assembly. Common examples of how workers are positioned are ladders, scaffolding and mechanical lifts. To speed assembly and improve flexibility during assembly easily repositioned rolling scaffolding and hydraulic lifts should be used.

Aiding Positioning through Joint Design

The positioning of large components and modules is a difficult and expensive operation, particularly where precision is an important factor. The basic methods for designing joints to aid positioning components is the specifying of clearances and chamfers. An example in large assembly is steel erection where the beam length is limited so that the clearance allows the beam to be swung into place between columns.

When selecting joints for large assemblies, joint configurations which aid in the aligning components should be used. An example is the use of seated connections versus framed connections in steel beam erection. Diagrams of the connections are shown in Figure 3.10. The seated joint has the advantages of supporting the beam while it is aligned, providing greater clearances when framing into column webs and reducing problems in aligning the holes for the bolted connections [Glidden 94].
Figure 3.10 Typical bolted beam connections [Glidden 94].

An example of considering the location of components during joint design is the girders and bridge piers of the Confederation Bridge [Gilmour 97]. The main girders for the bridge are 192 m long and were cast on-shore then transported and placed on the piers. To aid in the positioning of the girders a pier-top template with alignment keys was included in the design. The template was match cast to the pier segment of the girder on-shore then positioned and grouted to the pier. Therefore, the alignment of the girders on top of the piers were ensured to be correct.

Fixtures

The design of fixtures needs to be considered since the design of fixtures affects product quality and assembly efficiency. Fixtures are important in assembly because they provide more consistent and accurate part location, allow tighter tolerances and increase assembly speed compared to the manual location of parts. The typical method of employing fixtures is to tack components together in the fixture and then remove the assembly from the fixture for the completion of the joint.
The following guidelines for the desirable features of fixtures are given in the American Welding Society Welding Handbook [Connor 87]:

1. Weld joints must be accessible in the fixture.
2. The fixture must be more rigid than the assembly.
3. Holddowns, clamps, and threads of bolts and nuts should be protected from weld splatter.
4. The fixture should allow assembly of the work with a minimum of temporary welds that are visible on completion.
5. The work piece must be easy to remove from the fixture after welding is complete.

Although taken from a welding handbook, points 1 and 5 apply when using any joining method. The accessibility of joints is particularly important when using automated tooling which is not as agile as the manual equivalent. Point 5 becomes more important when using indexed automated tooling where assemblies must be inserted and removed manually as quickly as the tooling is able to perform the joints.

Often fixtures used for large structures are assembled along with the product and then are destroyed once the assembly is finished. Although this process may be appropriate for some products it is important that a reusable fixture be considered. Where fixtures need to be dismantled, using threaded fasteners rather than welding the fixture together should be considered. The motive for designing reusable fixtures is to save money especially on long production runs.

Economic advantage can be realized by designing assemblies so that the same fixture can be used with several similar assemblies. For example, if the variation in an assembly is the length of a frame then an adjustable fixture can be designed to accommodate different frame lengths by moving clamps mounted using pins or threaded fasteners. A simple diagram illustrating this idea is shown in Figure 3.11. For assemblies where part location is critical and usually requires manual adjustment, fixtures should be designed to allow fine tuning of location, perhaps by using threads. The reuse of fixtures should be considered both during product and fixture design.
Figure 3.11 Fixture design for fixture reuse.

3.3.4 Assembly Planning

Perhaps the greatest difference between the requirements of a design for large assembly and existing DFA methods is the need for addressing the design of the assembly process. Due to low volume projects being relatively unique, the combination of assembly processes and sequence will be unique and influenced by the product design. The fundamental design work which affects assembly planning is the selection of the assembly concept. As the design progresses further, the specific assembly plan requirements are determined and the assembly plans need to be developed. The assembly planning issues which need to be considered are:

- Sequencing,
- Scheduling,
- Resource allocation,
- Material handling, and
- Process planning.

Assembly Concepts

As large products are designed the process which will be used to assemble the product is largely determined. Therefore in projects which involve a level of originality, the process required for assembly may need to be developed from a primitive level. Assembly concepts which need to be examined include the level of modularity, arrangement of modules and the assembly sequence. An example of introducing modularity into a design is found in ship building which reduces design and assembly...
time. The strategy is to "pre-outfit ships - that is, design and build them in blocks that are outfitted with standard components (pipes, raceways, and similar equipment) and then welded together inside the finished hull" [Deitz 95].

Advantages can be realized in considering the sequence of assembly and integrating the assembly in the processes. General examples are the installation of stairs, mechanical lifts and cranes which are required for operation and can be used during assembly by placing them early in the assembly sequence [Perkowski 88]. An excellent example of exploiting the assembly sequence is the development of the "Monkey" for assembling the legs of an offshore drilling vessel reported by Johnson and Krog [Johnson 79]. The vessel has three legs, which are open trusses made of tubular steel. The legs are assembled above the vessel deck in 48 ft high, 100 ton sections to a total leg length of 410 ft. The leg sections are positioned by the Monkey which can climb previously placed leg sections using a rack and pinion. The Monkey uses clamps to secure the leg section being lifted and is able to position the section over the last installed section for welding.

Assembly Sequencing

An important component of design for large assembly is the consideration of the assembly process and therefore the assembly sequence needs to be determined. The assembly sequence determines the joint accessibility, the flow of materials and the allocation of resources. The general factors which have been identified by Echeverry et al [Echeverry 91] which restrict construction sequences are:

- Physical relationships among building components,
- Trade interaction,
- Path interference, and
- Code regulations.

The physical relationships are the most obvious constraints, including restrictions such as spatial requirements and support or protection requirements. Trade interactions restrict what processes can be concurrently undertaken due to space, resource and safety restrictions. Path interference delays the installation of a component due to the inability to transport it to the installation location. Code regulations identified which constric
sequences are safety regulations and inspection requirements. Clearly the assembly sequence is heavily related to the already discussed considerations of joining, handling and positioning. The sequencing considerations for DFA include accessibility, resource allocation and assembly process design.

**Scheduling and Sequencing**

The generation of the schedule and the assembly sequence is a critical part of assembly planning. The schedule and sequence ensures that assembly will be efficient and avoids rework due to incorrect assembly. The process of creating the assembly sequence is important because it will reveal assembly problems before construction begins. An example of efficient scheduling and sequencing is scheduling tasks which will cause delays in other tasks in periods of low activity, such as on weekends. This strategy was successfully implemented on a nuclear power station where a 450-ton reactor vessel and two 750-ton generators set in place on three successive weekends, with each lift completed in a single shift [Parker 84].

**Resource Allocation and Selection**

The allocation of resources is difficult to consider during design but is a significant assembly concern. The main goal during design is to reduce the requirements and critical activities which are dependent on heavily allocated resources. The allocation of resources is particularly critical in assemblies which incorporate large modules and therefore require extremely heavy or specialized equipment. An example determining the location and selection equipment is the area of heavy lift optimization and module transportation.

**Material Handling and Assembly Site Layout**

Due to the low volume production associated with large, complex products, the planning of handling and storing materials is often overlooked. This has been recognized as a common and serious problem in the construction industry [Tommelein 94]. However, material handling is an important part of assembly planning. The arrangement and flow of materials and equipment needs to be determined before production begins. In a construction area, the layout of temporary structures and material storage also needs to be determined [Oglesby 89]. When using a modular
design, the possible locations of the various assembly yards should be examined and planning of routes and access for the large transport vehicles required must be considered [Varghese 95].

Assembly Process Planning

The planning of assembly processes is a difficult task due to the number and variety of processes involved. The planning of typical assembly processes, such as the process for performing welds, is relatively straightforward but important for ensuring quality. Some processes will depend more on the structure of the assembly being created. An example of an activity which needs careful planning is the use of large equipment which is difficult and expensive to move such as cranes and concrete pumps [Lin 96a]. When using such equipment the tasks must be planned so that the optimal equipment locations are determined. Some tasks require detailed analysis due to the cost, complexity and importance in the assembly sequence. The placement of the modules requires careful planning and the use of 3D CAD programs for visualization is common practice by large companies [Lin 96b].

3.4 Design For Large Assembly Guidelines

The goal of the creation of design for large assembly guidelines is to aid designers in considering assembly issues. However, it is recognized for the program to be effective it must be organized appropriately. A poorly organized set of guidelines would be likely seen as nuisance rather than an aid. Therefore the guidelines have been organized into the stages of systematic design [Pahl 88]:

- Clarification of the task
- Conceptual design
- Embodiment design, and
- Detail design.

By arranging the guidelines in this manner, the guidelines allow the designer to efficiently address assembly concerns which ensures that assembly options are examined.
3.4.1 Clarification of the Task

In the first stage of design the requirement and constraints for the product are identified. Once the requirements are identified, the technical specifications of the project are developed. The important consideration during this stage is that the concurrency paradigm be implemented by clarifying the assembly requirements for the design.

1. Define assembly requirements. The assembly requirements which need to be determined are:
   - Production volume,
   - Required assembly time or period,
   - Projected cost for assembly,
   - Assembly method to be implemented, and
   - Module size limit specifications.

   These specifications then need to be established as the basis for assembly related design decisions.

3.4.2 Conceptual Design

The goal of the conceptual design phase is the establishment of function structures and the development of conceptual design solutions. The greatest assembly advantage at this abstract phase of the design is the investigation of how the function structures can be divided into modules. As the design becomes less abstract, concepts for the assembly of the product should also be developed. The development of assembly concepts serves as a tool for implementing the concurrent design paradigm, by involving a variety of specialists in the development of the design.

1. Develop assembly concepts for the design concepts which are considered. As part of design evaluation the assembly needs to be included as part of the concurrent engineering effort.
   i. Determine if the use of modules or subassemblies will produce advantages.
   ii. Examine the process and equipment requirements including the handling and position requirements.
iii. Develop and evaluate advantages of different sequences for major assembly tasks.

2. **Determine the modules and layout.** The assembly components need to be grouped into modules. During conceptual design the important concerns are the sizes of the modules and the layout of modules.

3.4.3 **Embodiment Design**

During the embodiment design phase the greatest number of assembly considerations are involved. The embodiment design phase consists of creating and comparing possible layouts and forms for products. The layout and form of the design is the largest factor determining the assembly efficiency. The assembly considerations which need to be examined include:

- Design simplification,
- Modularity considerations,
- Joining process selection,
- Joint layout and design,
- Positioning requirements, and
- Handling requirements.

1. **Simplify the design.**
   i. Eliminate unnecessary parts based on the Boothroyd-Dewhurst guidelines.
   ii. Identify components and processes which could be standardized.

2. **Account for the requirements of using modules and subassemblies.**
   i. Consider the effects of component and module size, mass and center of gravity during handling and positioning.
   ii. Ensure that joints can be created with the required quality between modules.

3. **Consider assemblability during joint layout and design.**
   i. **Process Selection.** Where possible use welding. In cases where access is difficult or to control labour costs use bolting. Use of riveting and adhesives should be limited to cases where welding or bolting are not satisfactory.
ii. **Process Automation.** Consider the use of automated equipment for performing large or repeated joints and where consistency requirements are high. Design for sufficient joint simplicity and access.

iii. **Joint Accessibility.** For minor problems joint redesign can improve accessibility. In cases where accessibility is a significant problem, the assembly should be divided into subassemblies.

iv. **Joint Type Selection.** Ensure that the joint type selected can be efficiently performed. The joint type should also aid in positioning, particularly for joining large components.

4. **Consider positioning requirements.**
   i. Use joint layouts which can make the most effective use of positioners.
   ii. Design part configurations and fixtures which allow fixtures to be reused. If possible fixtures should be designed to handle variations on a assembly configuration.

5. **Consider handling requirements.**
   i. Include handling specific features in subassemblies such as eyes for lifting or lugs for securing the subassembly during transport.
   ii. Limit component and subassembly dimensions and mass.

3.4.4 **Detail Design**

In the detailed phase of design the final details of the arrangement and components are specified and the required technical documents are produced. The effect of the specification of design details will not have as large an affect on the assembly efficiency as long as good design as specified by standards is carried out. The typical design details which would be of concern during detail design are the specifications of tolerances and of joints.

In designing large assemblies consideration needs to be made of the assembly process. Since at the detail design phase most of the design is completed, an assembly plan can be developed which may reveal areas where design changes need to be made.
1. Optimize joining processes.
   i. Optimize joint details such as weld dimensions and bolting patterns.
   ii. Consider combining joining processes. For example different welding process may be used for the root pass or adhesives may be combined with another joining process in a single joint.
   iii. Consider the access requirements for joining equipment. Use joints where access from only a single side is required where necessary.

2. Develop an assembly plan.
   i. Process Planning. Consider which processes are the most complex and important. For example the positioning of large assemblies or processes which are repeated.
   ii. Material Handling and Site Layout. Examine the alternatives for site layout and the resulting material handling requirements.
   iii. Sequencing and Scheduling. An assembly sequence and schedule need to be developed to identify problems and which processes are critical for completing the assembly on time.
   iv. Resource Allocation. Determine the resource requirements for assembly, including the amount, capacity and flexibility of the required equipment.

3.5 Discussion

The design for large assembly guidelines developed are a valuable tool for aiding designers in considering assembly during design. The design guidelines identify the areas of design which affect assembly and some general strategies for addressing these issues. However, the guidelines do not compare in maturity to the DFA methodologies which exist for smaller products. The DFA methodologies also incorporate methods for providing a quantitative evaluations of the ease of assembly. To develop a quantitative evaluation method for a large assembly will require an enormous amount of research, due to the variety and variability in the processes used.
4. Project Scheduling

4.1 Introduction

The purpose of this thesis is to develop methods for improving large structure assembly and construction by reducing cost and project duration. In this chapter, an algorithm for determining optimal or near optimal schedules for large projects is discussed. Since scheduling algorithms already exist, the goal in the development of this method was to produce an algorithm which was robust, flexible and able to handle complex problems.

The algorithm was developed in an effort to deal with large scheduling problems which are too numerically complex to be solved efficiently by existing algorithms and heuristics do not produce optimal results.1 The nature of the project scheduling problem is combinatorial, such that the computational requirements increases faster than problem size as more parameters are added. The technical description of the problem is NP-hard [Neumann 95]. To deal with the complexity of the scheduling problem a genetic algorithm was selected to perform the search. Genetic algorithms in addition to being suited to combinatorial problems are a robust and flexible search method. Another characteristic of project scheduling, is that the possible solutions are often highly constrained. In order to satisfy the scheduling constraints, constraint satisfaction techniques were also incorporated into the algorithm.

---

1 Similar reasoning and objectives led Chan et al [Chan 96] to identify genetic algorithms as an appropriate basis for project scheduling.
There are many different formulations of the project schedule optimization problem. Therefore the characteristics of the problem to be solved need to be defined. Schedule optimization problems can be characterized by the task, resource and constraint types used and the objective of the optimization. The specific problem considered in this study is generally known as the minimum duration, resource constrained scheduling problem. The tasks which are used to create the schedule have fixed durations and resource requirements\(^1\). The resources used are daily renewable and have fixed availabilities. The constraints used in the developed algorithm are from the Precedence Diagramming Method, referred to as the generalized precedence relations, which allow more complex task relations then are commonly used in schedule optimization.

Before the developed scheduling algorithm is presented, the concepts on which the algorithm is based are discussed. The primary basis of the algorithm is genetic algorithms. Constraint satisfaction techniques are also included in the algorithm for creating schedules.

4.2 Genetic Algorithms and Constraint Satisfaction

4.2.1 Genetic Algorithms

The genetic algorithm has become widely popular as an optimization tool, based on the robust nature and simple implementation of the algorithm. The basis of the algorithm is the analogy to natural selection theory in biology and genetics. The analogy to natural processes is quite loose although very powerful for solving certain kinds of problems. The power and looseness of the analogy has led to the development of many problem specific variations of the algorithm. To illustrate the principles of the algorithm, a simple example created by Goldberg [Goldberg 89] will be used.

The most unique feature of genetic algorithms is the use of several solutions during each iteration of the optimization. The analogy is that the set of solutions represents a biological population and each iteration of the optimization produces a new generation. The optimization proceeds by combining selected population members, or

---

\(^1\) Tasks with more complex resource requirements can be modeled using several simple tasks and constraints between them
solutions, of the current generation to produce the next generation of solutions. The optimization continues until a specified number of generations are completed.

**Chromosome Representation**

The critical decision in applying genetic algorithms to solve a problem is the encoding of the problem parameters in a string or 'chromosome.' Each string or chromosome then represents a solution to the problem being solved. The positions in the string are analogous to genes. For example, in the encoding of a number into a binary string, each digit in the string is a gene. The allowable values that can be held in a position are analogous to alleles in natural systems. For the binary string example, the alleles for each position are 0 and 1.

**Evaluation**

Once a population of chromosomes has been created, the solutions must be evaluated to produce a measurement of fitness for each solution. For example, consider optimizing the square of a binary number. The fitness of the solution can be determined by converting the number to base 10 and calculating the square. Most often the evaluation function will not be as simple as this, since for such simple problems other optimization methods will be more efficient.

**Selection**

The selection of chromosomes for producing a new generation is performed using the fitness score of the chromosomes to determine the probability of selection. The selection method is commonly referred to as the roulette wheel because having a larger fitness score improves the chances of selection similar to having a larger space on a roulette wheel.

**Crossover**

To produce the next generation of chromosomes, existing chromosomes are combined using a crossover. To produce two new chromosomes, two existing parent chromosomes are selected. A point between the genes in a chromosome is then randomly selected for the crossover point. The genes before the crossover point are then combined with the genes after the crossover point of the other parent chromosome to
produce a new chromosome. In this way two offspring chromosomes are formed. The process is illustrated in Figure 4.1.

![Figure 4.1 Schematic of a simple crossover [Goldberg 89].](image)

**Mutation**

Mutation is implemented in genetic algorithms to introduce new characteristics or diversity into a population. The effect is to increase the variety of the chromosomes in the population and to avoid the loss of positive characteristics due to premature convergence of the population. The mutations are performed by randomly selecting a chromosome and position and then replacing the current allele with a randomly selected allele. The rate of mutation is normally low, such that a small percentage of the chromosomes are mutated in each generation. For the example of a chromosome representation which is a binary, a mutation may be performed by randomly selecting a position in the chromosome and changing the value at the position.

**4.2.2 Constraint Satisfaction**

Constraint satisfaction techniques are comprised of a broad and varied set of methods. The scope of the methods is so large because of general nature of constraint satisfaction problems and the importance of having appropriate algorithms. The constraint satisfaction problem is defined by Tsang [Tsang 93] as

a problem composed of a finite set of variables, each of which is associated with a finite domain, and a set of constraints that restricts the values the variables can simultaneously take. The task is to assign a value to each variable satisfying all the constraints.
The search methods which are employed in the developed scheduling algorithm are part of the general backtracking and lookahead methods.

**Backtracking**

Backtracking is an intuitive method for solving constraint satisfaction problems. In the backtracking algorithm, the constraints are checked for satisfaction as each variable is assigned a value or labeled. If the label does not satisfy the constraints with the previously labeled variables, the label is changed. Once all possible labels for a variable have been tried the algorithm backtracks to the last successfully labeled variable and searches for another valid label, while the previous label is marked invalid. If the algorithm backtracks to the first variable labeled and all the possible values are marked invalid, the problem cannot be solved.

To improve the efficiency of the backtracking algorithm, the structure of the problem constraints can be used. The resulting algorithm is back jumping which works in the same way as backtracking, except that more than a single variable is back tracked at a time. In back jumping, when a variable cannot be validly labeled, the algorithm back jumps to the last validly labeled variable which is related to the current invalid variable by a constraint. The use of knowledge of the problem structure contained in the constraints improves the efficiency of the algorithm by avoiding relabeling variables which are not related to the constraint which has been violated.

**Lookahead Algorithm**

The lookahead algorithm attempts to improve efficiency by reducing the problem space as variables are labeled. The characteristic of lookahead strategies is the removal of values from the domains of unlabelled variables which are incompatible with the labeled variables. The variations in lookahead algorithms are determined by what degree the unlabeled variables are inspected. The lookahead algorithm implemented in the scheduling algorithm does not ensure that valid solutions remain after labeling each variable, and therefore backtracking is included as part of the algorithm to ensure that successful solutions can be found. To ensure that a lookahead algorithm is successful without using backtracking, after each variable is labeled, the unlabeled variables must
be inspected to ensure that the remaining possible values in the domains can satisfy the constraints between the unlabelled variables.

4.3 The Developed Algorithm

4.3.1 General Strategy

The developed algorithm for solving the project scheduling problem is based on a combination of genetic algorithms and constraint satisfaction. However the use of a genetic algorithm alone is unlikely to create valid schedules by selecting random task start times. To ensure that valid schedules are generated, a method was developed so that the chromosomes in the genetic algorithm contain the order in which the tasks are considered for scheduling and constraint satisfaction is used to determine the task start and finish times. The concept for combining the genetic algorithm with constraint satisfaction is illustrated in Figure 4.2. The fundamental concept behind this algorithm is that the order in which the tasks are considered for scheduling will determine the schedule produced. The differences in the schedules which are produced is due to the change in priority of tasks for resource allocation, as determined by task position in the chromosome.

![Diagram](image)

**Figure 4.2** Concept for combining genetic algorithm and constraint satisfaction techniques.

To demonstrate how altering priority changes the schedule, consider scheduling the three tasks listed in Table 4.1 under only resource constraints. Some sample
schedules are shown in Figure 4.3 which vary due to the priorities assigned to tasks for scheduling.

Table 4.1 Tasks for demonstration example.

<table>
<thead>
<tr>
<th>Task Number</th>
<th>Duration</th>
<th>Resources Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>A &amp; B</td>
</tr>
</tbody>
</table>

Figure 4.3 Sample schedules for demonstration example.

The constraint satisfaction technique used in the algorithm is lookahead constraint propagation. The technique is implemented by propagating the effects of scheduling a task to the available time domains of the unscheduled tasks through the precedence and resource constraints. Since the lookahead technique implemented was not capable of producing valid schedules from all chromosome arrangements, a method of altering the chromosome was implemented which is similar to the back jumping technique.

4.3.2 Chromosome Representation

The chromosome representation used for the scheduling algorithm is a list of tasks in the order which the tasks are considered for scheduling. To represent the tasks in the chromosomes each of the tasks are assigned a task number. Therefore the
chromosomes are vectors of task numbers. The initial chromosomes for the genetic algorithm are created by randomly placing the task numbers in the chromosomes.

4.3.3 Schedule Generation

The schedules are generated by labeling the tasks in the order that the task numbers are contained in the chromosome. When a task is selected to be scheduled, the task is labeled with the earliest start and finish times which are available in the domain of start and finish times of the task. Selecting the earliest times remaining in the domain reduces the number of schedules which may be generated, however it is only important that the optimal and near optimal schedules remain. Once the task is labeled the constraints are checked to see if any constraints have been violated. If none of the problem constraints have been violated, the effects scheduling the task are propagated through the precedence and resource constraints. If a constraint violation is detected, the algorithm back tracks and modifies the chromosome by changing the order of the tasks. The modified chromosome is then attempted to be scheduled.

4.3.4 Task Representation

A task is defined by the resources required and the duration of the task. As part of the implementation, the task data includes the references to the precedence constraints related to the task, a set of valid possible start and finish times for the task and the assigned start and finish times for the task. The domain of the start and finish times is altered by the propagation of constraints. The maximum and minimum values in the domain can be altered and windows of time can be removed, creating gaps in the domain.

4.3.5 Constraint Propagation

Each task and resource has a constraint propagation routine associated with it. Once a task has been scheduled, the resource constraint propagation routine for all the resources required for that task are called. The resource first allocates the amount required for the task from the resource time domain. The routine then checks all unscheduled tasks which require the resource to determine if the allocation of the resource makes it impossible for a task to be scheduled in the valid time domain of the unscheduled task. If time is removed from the time domain of a task, the precedence
constraint routine of the task is called once the resource constraint propagation is completed.

The task constraint propagation routine checks the scheduled start and finish times or the earliest and latest possible start and finish times in the valid time domain of the unscheduled tasks which are related by precedence constraints. If time needs to be removed from the time domain of a related task, the change is made and the constraint propagation routine of the modified task is called. The calling of task constraint propagation routines continues until the effect of scheduling the task is propagated through the entire network.

4.3.6 Chromosome Modification

Once a constraint conflict has been detected, the new position of the unsuccessfully labeled or invalid task must be determined. The basis of the genetic algorithm is that the solutions found by crossover from a previous generation are better than random chromosomes. Therefore the smallest change in ordering which will result in a successful schedule is desired. However, moving the task a single position forward at a time is extremely inefficient. Therefore positions which are likely to make scheduling possible are searched for.

The furthest position considered is the position of the task which was related to the invalid task by the violated precedence constraint. Each position between this initial position and the present position of the invalid task is a candidate if the task in that position meets the following conditions:

- The task is at least partially scheduled during the same time as the invalid task and uses a common resource,
- The task is ahead of the invalid task in the unmodified chromosome, and
- The task has not been selected for modification before by the invalid task during this generation.

The candidate position closest to the present position of the invalid task is selected as the position to insert the invalid task into the list. The modification of a chromosome is shown in Figure 4.4.
Figure 4.4 Illustration of chromosome modification.

Once the modification is complete the scheduling of tasks begins again starting at the new position of the previously invalid task. The labels for all the tasks ahead of this position are left the same as for the previously unsuccessful scheduling attempt. When a modified gene has been successfully scheduled the modified gene replaces the original gene in the population.

4.3.7 Scoring Function

The evaluation of the chromosomes begins by determining the durations of all the schedules in the population. Since the objective is to minimize the schedule duration, the schedule lengths cannot be used directly as the scores for the function. The scoring function used to determine the chromosome score is

\[
\text{score} = \int \left[ 1000 \left( A + B \frac{\text{time}_h - \text{time}}{\text{time}_h - \text{time}_l} + C \left( \frac{\text{time}_h - \text{time}}{\text{time}_h - \text{time}_l} \right)^2 \right) \right],
\]

where:

- \( \text{score} \) = the score for the current chromosome,
- \( \text{time} \) = the schedule length of the current chromosome,
- \( \text{time}_h \) = the longest schedule length in the current generation,
- \( \text{time}_l \) = the shortest schedule length in the current generation, and
- \( A, B, C \) = user defined variables.
The variable A provides a base line score for all chromosomes so that all chromosomes have a minimum probability of being picked. The B variable linearly rewards improvements in schedule and C quadratically rewards improvements in schedule. Experimenting with the program determined only very small values of C relative to B and the A term were required or else the algorithm would too quickly converge to a non-optimal value. However, a C term was required in some cases where the schedule lengths of the good and poor chromosomes were near the same value. Typical values used for A, B and C are 0.2, 0.6 and 0.1 respectively. A more conventional scoring function could be used if an upper bound was always known for the problem being solved.

4.3.8 Crossover Routine

An important characteristic of genetic algorithms is the combination of solutions to generate new solutions through the use of the crossover. Since the encoding of the chromosome is based on the ordering of tasks in the chromosome, typical crossover routines do not generate useful or valid chromosomes. A crossover routine was developed which creates offspring chromosomes which reflect the ordering of tasks in the parent chromosomes.

The parent chromosomes are selected using the conventional roulette wheel method. The created crossover routine only develops a single offspring chromosome. The offspring chromosome is generated by inserting task numbers starting at the highest priority position. For each position a parent chromosome is randomly selected and the highest priority unselected task is used. The process is shown in Figure 4.5.
Developed crossover method to combine the gene ordering information in the parent chromosomes.

### 4.3.9 The Algorithm Implementation

The purpose of this section is to discuss in detail the implementation of the algorithm. A flowchart illustrating the algorithm implementation is shown in Figure 4.6. To explain the implemented algorithm the routines are described.

**Input Data:** The data input to the algorithm is contained in a text data file. The data file includes resource, task and constraint data as well as parameters which determine the execution of the algorithm. The parameters include the number of resources, tasks and constraints, population size, number of generations, the parameters for the scoring equation and a random seed. The resources and tasks are stored in arrays of objects of the resource and task classes.

**Determine Default Time Domains:** Default time domains are determined for the tasks by propagating the precedence constraints without any task having start and finish times set. Times where it is impossible for the task to be scheduled are eliminated.

**Create Chromosomes:** The chromosomes are created by randomly selecting a task from the task array and storing the position of the task in the array used to represent the chromosome.

**Copy Chromosome:** The copy chromosome function is used to make a copy of chromosome which is changed during scheduling. Once the chromosome is successfully scheduled, the modified chromosome is copied into the population.
Figure 4.6 Flowchart of the implemented algorithm.
Schedule Task: To schedule a task, the schedule routine contained in the task class is called. The schedule routine assigns the earliest possible start and finish times to the task. The routines for applying constraints between tasks and the resource constraints are then called, propagating the effects of the task being scheduled to the unscheduled tasks.

Modify Gene: If a task cannot be scheduled without violating a constraint, the modify gene routine is called. The modify gene moves the task which could not be scheduled to an earlier position in the chromosome. The position to which the task will be moved is determined by analyzing the tasks which have been successfully scheduled. The invalid task is moved ahead of the nearest task which meets the following conditions:

- The task and the invalid task use a common resource type,
- The scheduled duration of the task and the invalid task overlap, and
- The invalid task has not been moved ahead of the task previously in the scheduling of the current chromosome.

Store Times for Scheduled Tasks: The start and finish times of the successfully scheduled tasks are stored in arrays. The times are stored so that the tasks which are not moved in the chromosome do not have the start and finish times determined again.

Reset Tasks and Resources: The reset functions for the tasks and resources are called, resetting the time domains to the default values.

Recreate Valid Portion of Schedule: For the tasks which were successfully scheduled and did not change position in the chromosome, the start and times are restored and the constraint propagation routines for each scheduled task is called.

Determine Schedule Length: The tasks are searched for the latest task finish time. If the chromosome has produced the shortest schedule recorded, the task start and finish times are written to output.

Generate Scores: The chromosomes are assigned scores using the scoring equation previously discussed.

Crossover Chromosomes: The chromosomes are crossed over using the crossover method previously discussed. The parent chromosomes are selected using the roulette wheel technique.
**Mutate Chromosomes:** the chromosomes are mutated using the previously discussed mutation method.

### 4.4 Algorithm Demonstration

In order to develop and demonstrate the schedule optimization algorithm, the algorithm was implemented and tested. The algorithm was implemented using C++, and using object-oriented programming to represent the tasks in the schedule, as previously described.\(^1\) The algorithm was demonstrated using two benchmark problems. The first benchmark problem used was “The Bridge Problem” created by Bartusch [Bartusch 88]. The second was a randomly generated problem by Stinson et al [Stinson 78].

#### 4.4.1 First Case Study

**The Bridge Problem**

The bridge problem is the scheduling of the simplified construction of a bridge, developed by Bartusch. The bridge is shown in Figure 4.7. The construction of the bridge consists of 44 tasks and which are related by relatively complicated precedence constraints. A list of the tasks, durations and resource requirements is given in Table 4.2. The basic precedence requirements are simple, such as the completion of the masonry work on the columns before the preformed bearers can be placed. In addition to these constraints there are the following time constraints:

1. The time between the completion of a particular formwork and the completion of its corresponding concrete foundation is at most 4 days.
2. There are at most 3 days between the end of a particular excavation (or foundation piles) and the beginning of the corresponding formwork.
3. The formworks must start at least 6 days after the beginning of the erection of the temporary housing.
4. The removal of the temporary housing can start two days before the end of the last masonry work.

---

\(^1\) The g++ compiler for the LINUX operating system was used.
5. The delivery of the preformed bearers occurs exactly 30 days after the beginning of the project.

The resources available for the project are limited to one of each type.

![Diagram of a simple five segment bridge](image)

**Figure 4.7** Simple five segment bridge.

**Solution**

The implemented algorithm was able to find the known optimal solution of 104 days. The optimal solution was often found on the second or third generations. This is likely a result of the tasks being highly constrained in time, reducing the number of possible near optimal schedules to be searched. The number of generations required to find the optimal solution in ten trial runs of two different population sizes is shown in Table 4.3. Note that with a population size of 30, the algorithm stopped without reaching the optimal answer three times, because all members of the population had the same schedule length. By increasing the population size the diversity in the genetic pool is maintained for a greater number of generations, allowing the optimal solution to be found.
Table 4.2 Task specifications for the bridge problem.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Duration</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA</td>
<td>Beginning of Project</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>Excavation (abutment 1)</td>
<td>4</td>
<td>Excavator</td>
</tr>
<tr>
<td>A2</td>
<td>Excavation (pillar 1)</td>
<td>2</td>
<td>Excavator</td>
</tr>
<tr>
<td>A3</td>
<td>Excavation (pillar 2)</td>
<td>2</td>
<td>Excavator</td>
</tr>
<tr>
<td>A4</td>
<td>Excavation (pillar 3)</td>
<td>2</td>
<td>Excavator</td>
</tr>
<tr>
<td>A5</td>
<td>Excavation (pillar 4)</td>
<td>2</td>
<td>Excavator</td>
</tr>
<tr>
<td>A6</td>
<td>Excavation (abutment 2)</td>
<td>5</td>
<td>Excavator</td>
</tr>
<tr>
<td>P1</td>
<td>Foundation piles 2</td>
<td>20</td>
<td>Pile-driver</td>
</tr>
<tr>
<td>P2</td>
<td>Foundation piles 3</td>
<td>13</td>
<td>Pile-driver</td>
</tr>
<tr>
<td>UE</td>
<td>Erection of temporary housing</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>Formwork (abutment 1)</td>
<td>8</td>
<td>Carpenter</td>
</tr>
<tr>
<td>S2</td>
<td>Formwork (pillar 1)</td>
<td>4</td>
<td>Carpenter</td>
</tr>
<tr>
<td>S3</td>
<td>Formwork (pillar 2)</td>
<td>4</td>
<td>Carpenter</td>
</tr>
<tr>
<td>S4</td>
<td>Formwork (pillar 3)</td>
<td>4</td>
<td>Carpenter</td>
</tr>
<tr>
<td>S5</td>
<td>Formwork (pillar 4)</td>
<td>4</td>
<td>Carpenter</td>
</tr>
<tr>
<td>S6</td>
<td>Formwork (abutment 2)</td>
<td>10</td>
<td>Carpenter</td>
</tr>
<tr>
<td>B1</td>
<td>Concrete foundation (abutment 1)</td>
<td>1</td>
<td>Concrete-mixer</td>
</tr>
<tr>
<td>B2</td>
<td>Concrete foundation (pillar 1)</td>
<td>1</td>
<td>Concrete-mixer</td>
</tr>
<tr>
<td>B3</td>
<td>Concrete foundation (pillar 2)</td>
<td>1</td>
<td>Concrete-mixer</td>
</tr>
<tr>
<td>B4</td>
<td>Concrete foundation (pillar 3)</td>
<td>1</td>
<td>Concrete-mixer</td>
</tr>
<tr>
<td>B5</td>
<td>Concrete foundation (pillar 4)</td>
<td>1</td>
<td>Concrete-mixer</td>
</tr>
<tr>
<td>B6</td>
<td>Concrete foundation (abutment 2)</td>
<td>1</td>
<td>Concrete-mixer</td>
</tr>
<tr>
<td>AB1</td>
<td>Concrete setting time</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AB2</td>
<td>Concrete setting time</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AB3</td>
<td>Concrete setting time</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AB4</td>
<td>Concrete setting time</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AB5</td>
<td>Concrete setting time</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AB6</td>
<td>Concrete setting time</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>Masonry work (abutment 1)</td>
<td>16</td>
<td>Bricklaying</td>
</tr>
<tr>
<td>M2</td>
<td>Masonry work (pillar 1)</td>
<td>8</td>
<td>Bricklaying</td>
</tr>
<tr>
<td>M3</td>
<td>Masonry work (pillar 2)</td>
<td>8</td>
<td>Bricklaying</td>
</tr>
<tr>
<td>M4</td>
<td>Masonry work (pillar 3)</td>
<td>8</td>
<td>Bricklaying</td>
</tr>
<tr>
<td>M5</td>
<td>Masonry work (pillar 4)</td>
<td>8</td>
<td>Bricklaying</td>
</tr>
<tr>
<td>M6</td>
<td>Masonry work (abutment 2)</td>
<td>20</td>
<td>Bricklaying</td>
</tr>
<tr>
<td>L</td>
<td>Delivery of the preformed bearers</td>
<td>2</td>
<td>Crane</td>
</tr>
<tr>
<td>T1</td>
<td>Positioning (preformed bearer 1)</td>
<td>12</td>
<td>Crane</td>
</tr>
<tr>
<td>T2</td>
<td>Positioning (preformed bearer 2)</td>
<td>12</td>
<td>Crane</td>
</tr>
<tr>
<td>T3</td>
<td>Positioning (preformed bearer 3)</td>
<td>12</td>
<td>Crane</td>
</tr>
<tr>
<td>T4</td>
<td>Positioning (preformed bearer 4)</td>
<td>12</td>
<td>Crane</td>
</tr>
<tr>
<td>T5</td>
<td>Positioning (preformed bearer 5)</td>
<td>12</td>
<td>Crane</td>
</tr>
<tr>
<td>UA</td>
<td>Removal of the temporary housing</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>V1</td>
<td>Filling 1</td>
<td>15</td>
<td>Caterpillar</td>
</tr>
<tr>
<td>V2</td>
<td>Filling 2</td>
<td>10</td>
<td>Caterpillar</td>
</tr>
<tr>
<td>PE</td>
<td>End of project</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.3 Number of generations required to find bridge problem optimal solution.

<table>
<thead>
<tr>
<th>Population Size</th>
<th>Trial Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
</tr>
</tbody>
</table>

The running time for the program using a population of thirty is approximately 10 seconds for completing 50 generations. The population that requires the greatest time is always the first, due to the greater modification of the chromosome required to create a valid schedule. The number of modifications required for the initial generation is approximately 22 per member of the population. The subsequent generations require less than three modifications per member of the population.

4.4.2 Second Case Study

Randomly Generated Problem

The second problem used to test the algorithm was randomly generated by Stinson et al [Stinson 78]. The specifications for the problem are shown in Figure 4.5. The problem differs from the bridge problem in two significant ways. First the randomly generated problem, only uses finish-to-start resource constraints. Secondly, the tasks in the randomly generated problem require multiple resource types, and each task requires at least one of each type, except the start and finish of the project.

\[1 \text{ N/A indicates the optimal solution was not found in 100 generations.}\]
<table>
<thead>
<tr>
<th>Activity Number</th>
<th>Duration</th>
<th>Resource Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0 0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3 2 4 2</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>2 3 3 1</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>5 4 3 2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1 3 3 1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>4 1 6 2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>4 1 4 5</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>1 2 2 2</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>1 5 2 3</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>6 1 5 3</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>6 1 6 3</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>5 1 6 6</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>3 4 4 3</td>
</tr>
<tr>
<td>14</td>
<td>7</td>
<td>1 2 2 3</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1 1 4 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity Number</th>
<th>Duration</th>
<th>Resource Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1</td>
<td>1 2 2 3</td>
</tr>
<tr>
<td>17</td>
<td>9</td>
<td>1 3 5 2</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>4 2 3 1</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>5 1 4 3</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>2 2 5 3</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>3 5 6 5</td>
</tr>
<tr>
<td>22</td>
<td>5</td>
<td>2 6 6 1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>3 2 4 1</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>2 6 4 3</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>1 1 2 4</td>
</tr>
<tr>
<td>26</td>
<td>5</td>
<td>6 3 6 1</td>
</tr>
<tr>
<td>27</td>
<td>6</td>
<td>6 2 2 6</td>
</tr>
<tr>
<td>28</td>
<td>3</td>
<td>5 1 5 1</td>
</tr>
<tr>
<td>29</td>
<td>3</td>
<td>3 3 2 4</td>
</tr>
<tr>
<td>30</td>
<td>7</td>
<td>4 3 4 3</td>
</tr>
<tr>
<td>31</td>
<td>5</td>
<td>2 5 3 3</td>
</tr>
<tr>
<td>32</td>
<td>2</td>
<td>1 3 5 5</td>
</tr>
<tr>
<td>33</td>
<td>3</td>
<td>2 3 3 2</td>
</tr>
<tr>
<td>34</td>
<td>6</td>
<td>4 1 4 1</td>
</tr>
<tr>
<td>35</td>
<td>7</td>
<td>5 5 5 4</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>1 4 5 4</td>
</tr>
<tr>
<td>37</td>
<td>3</td>
<td>1 5 1 2</td>
</tr>
<tr>
<td>38</td>
<td>2</td>
<td>5 3 1 3</td>
</tr>
<tr>
<td>39</td>
<td>2</td>
<td>2 4 3 3</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>6 3 2 3</td>
</tr>
<tr>
<td>41</td>
<td>2</td>
<td>2 6 5 4</td>
</tr>
<tr>
<td>42</td>
<td>3</td>
<td>3 3 2 4</td>
</tr>
<tr>
<td>43</td>
<td>0</td>
<td>0 0 0 0</td>
</tr>
</tbody>
</table>

Figure 4.8 Specifications for the randomly generated problem.
Solution

The algorithm was able to find the 74 day solution which is known to be optimal. The optimal solution was found using a population of 200 and maximum number of generations of 50. Using these parameters, the optimal solution was found four of five times. The amount of time required to complete the 50 generations was 30 minutes, running on a 200 MHz Pentium Pro. The average values for these trials are shown in Figure 4.9. A second set of trials was run using a population of 100, however, in all five cases the optimal solution was not found in 50 generations.

The effect of the mutation can be seen in the fluctuation of the average value of the populations from generation 18 to generation 50 in Figure 4.9. The mutation causes poor schedules to be produced and the average is kept from approaching closer to the optimum schedule length.

![Graph showing schedule duration vs generation number](image)

Figure 4.9 Average results for the randomly generated problem for 5 trials using a population of 200.

A major factor in the length of time required to run the program is the number of modifications which are required to produce feasible schedules from the initial set of chromosomes. The average number of modifications for each chromosome from the 200 population trials is shown in Table 4.4. The number of modifications for scheduling an initial chromosome is nearly three times as large as the number of tasks in the problem.
This result would seem to indicate that better rules for gene modification could be developed. The speed of the demonstration program could also be improved implementing the algorithm in C rather than C++ due to the greater code optimization available from C compilers.

Table 4.4 Average number of modifications per chromosome.

<table>
<thead>
<tr>
<th>Generation</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Modifi cations</td>
<td>120</td>
<td>12.7</td>
<td>10.0</td>
<td>7.5</td>
<td>5.1</td>
<td>3.5</td>
<td>3.4</td>
<td>3.2</td>
<td>2.8</td>
<td>2.6</td>
</tr>
</tbody>
</table>

4.5 Discussion

The results of the test problems demonstrates that the developed algorithm is able to optimize project schedule networks. The progress of the genetic algorithm in the optimization is relatively rapid, with the best solution that is found during a run of the program appearing after only a few generations. For solving more complex problems, a greater population may be required to provide more diversity in the population so that the search for the optimal solution does not converge before the optimal solution is found. To make the use of larger populations more practical, the speed of the demonstration program needs to be improved. As mentioned previously, the speed of the program could be improved by a more efficient implementation or refining the algorithm for modifying chromosomes which cannot be scheduled.

Since the purpose in developing the program is to improve actual project assembly schedules, the program needs to be tested using an actual project schedule. Through testing with an actual schedule, the algorithm performance can be evaluated for problems with more realistic numbers of tasks and constraints. It is by allowing variation in the numbers of tasks and constraints that the strength of using a genetic algorithm will be demonstrated. The use of the program for generating partial project schedules, by completely constraining completed tasks, could also be investigated in testing with actual scheduling networks.
5. Conclusion

Large structure assembly is a complex, long and expensive process which is critical in determining the success of a project. These characteristics of large structure assembly are caused by the complexity of the individual assembly processes, the number of processes and resources required and the complicated relationships between the processes. Two tools have been developed to address these problems and improve assembly efficiency:

- Design for large structure assembly guidelines, and
- A resource-constrained project schedule optimization algorithm.

Design for assembly methodologies have been extensively developed for smaller products and have significantly improved the assembly efficiency of designs. To produce similar results for large structure assembly, three paradigms need to be applied. First, a concurrent design process is required to consider assembly during design and to integrate the design process across engineering disciplines. Simplicity must be a requirement of the design, which can be created by reducing the number of parts in a design and standardizing parts and assembly processes. Finally, a modular design needs to be investigated as an aid for applying concurrency and simplicity and increasing the ease of assembly.
To make the application of the paradigms more practical, the following considerations were identified as being important in improving assembly efficiency:

- Joining,
- Handling,
- Positioning, and
- Assembly planning.

To address these considerations the guidelines were developed. To make the guidelines easier to apply they were organized according to the design process.

A. Clarification of the Task
   1. Define the assembly requirements.

B. Conceptual Design
   1. Develop assembly concepts for the design concepts which are produced.
   2. Determine the modules and layout.

C. Embodiment Design
   1. Simplify the design.
   2. Account for the requirements of using modules and subassemblies.
   3. Consider assemblability during joint layout and design.
      i. Process selection
      ii. Process automation
      iii. Joint accessibility
      iv. Joint type selection
   4. Consider positioning requirements.
   5. Consider handling requirements.
D. Detail Design

1. Optimize joining processes.
2. Develop an assembly plan.
   i. Process planning
   ii. Material handling and site layout
   iii. Sequencing and scheduling
   iv. Resource allocation.

The developed guidelines can be an effective tool for forcing designers to consider assembly during design. However, the guidelines can become a more effective tool if implemented into a quantitative methodology similar to those for existing DFA methods.

Resource constrained project scheduling is numerically complex problem. To optimize project schedules an algorithm has been developed based on genetic algorithms and constraint satisfaction techniques. The genetic algorithm is used to determine the priority of tasks for scheduling. Constraint satisfaction techniques are used to generate schedules using the task priorities.

To demonstrate the algorithm it was implemented as a computer program. The implementation successfully determined the known optimal schedule for two problems taken from project scheduling literature. The first was the "Bridge Problem" which consisted of tasks requiring a single resource, but which were related by complex constraints. The second problem solved was a randomly generated network of tasks related by simple constraints, but requiring multiple resources.

Although both problems were successfully solved, the amount of time required to solve the first generation for the randomly generated problem was greater than fifteen minutes for a population of 200. To improve the speed of the program either a more efficient implementation or refinement of the schedule generating algorithm should be considered. The algorithm could also be further developed by testing with real project schedules.
5.1 Research Contributions

There are two significant research contributions made in this thesis. First, a set of guidelines for considering assembly during the design of large structures have been created. The guidelines were developed by applying DFA principles used for smaller products to large structures. The set of guidelines are unique, although the considerations included may have been made informally in many existing projects.

Secondly, genetic algorithms and constraint satisfaction techniques have been combined in an algorithm for resource constrained project scheduling under generalized precedence relations. The use of the genetic algorithms for determining the priority of tasks is not original. However, using this technique for schedules under generalized precedence relations is original. To generate the schedules under the complex task relations, constraint satisfaction was incorporated into the algorithm.

5.2 Future Research

The ultimate purpose of the research direction of this thesis is the development of an integrated tool for large structure assembly design and planning. The major portions of such a tool would be assembly process selection and construction simulation. The development of the design for large structure guidelines is only the beginning of an assembly process selection system. Further work needs to be in compiling data on assembly process characteristics, times, cost models and resource requirements, and then organizing the data into an expert system.

Project scheduling is only one component required for a construction simulator. An important component of a simulator is integration with a computer-aided design system to visually display the construction with 3D graphics. Also needed for an effective construction simulator is a sequencing tool for automatically generating sequence requirements. Further development of the project scheduling algorithm to incorporate probabilistic task durations should also be considered.
6. References


<table>
<thead>
<tr>
<th>Reference</th>
<th>Author(s)</th>
<th>Title</th>
<th>Journal/Book Details</th>
</tr>
</thead>
</table>


