Agile manufacturing system scheduling using genetic algorithms and simulated annealing

Sherif Masoud

Follow this and additional works at: https://fount.aucegypt.edu/retro_etds

Recommended Citation

APA Citation

MLA Citation

This Thesis is brought to you for free and open access by AUC Knowledge Fountain. It has been accepted for inclusion in Archived Theses and Dissertations by an authorized administrator of AUC Knowledge Fountain. For more information, please contact mark.muehlhaeusler@aucegypt.edu.
AGILE MANUFACTURING SYSTEM SCHEDULING USING GENETIC ALGORITHMS AND SIMULATED ANNEALING

by

SHERIF ALI MASOUD

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Engineering

with specialization in:

Industrial Engineering

under the supervision of:

Dr. Lotfi K. Gaafar
Associate Professor, Mechanical Engineering Department

Fall 2003
I would like to express my deep gratitude to Dr. Lotfi Gaafar, my adviser, for his valuable advice, motivation, and feedback. He suggested to research agile manufacturing scheduling, and he was continuously available for consultation. I thank Dr. Ashraf Nassef for teaching a graduate course on genetic algorithms and simulated annealing. I profoundly thank my family, specially my mother and father, for their solid encouragement and precious support. I also thank all my professors and everybody who helped me in one way or another during my thesis work.
ABSTRACT
The American University in Cairo
Agile Manufacturing System
Scheduling Using Genetic Algorithms and Simulated Annealing
Sherif Ali Masoud
Supervisor: Dr. Lotfi Gaafar

Agile manufacturing is concerned with thriving in prevailing market conditions by quickly introducing new or modified products. This research deals with the scheduling of an agile manufacturing system (AMS), which performs both machining and assembly, with the objective of minimizing the makespan. The AMS allows the production of high varieties of modular products in small batches and at low costs. This problem is difficult to solve optimally and was solved in literature by heuristic algorithms. In the current research, four novel, genetic algorithms and simulated annealing-based, heuristics – General Genetic Algorithm, General Simulated Annealing, Heuristic Assisted Genetic Algorithm, and Heuristic Assisted Simulated Annealing – are developed to address this scheduling problem. A 2³ factorial experiment, replicated twice, is conducted to compare the performance of the proposed and existing heuristics and identify the significant factors that affect the resulting percentage deviation from the lower bound and the frequency of resulting in the best solution. The results show the superiority of the developed heuristics to those existing in literature in addition to identifying the significant factors and interactions.
# TABLE OF CONTENTS

LIST OF TABLES .................................................................................................................. viii  
LIST OF FIGURES ............................................................................................................... ix  
LIST OF ACRONYMS ......................................................................................................... xiv  
LIST OF SYMBOLS ............................................................................................................ xv  
CHAPTER 1 ......................................................................................................................... 1  
   INTRODUCTION............................................................................................................. 1  
      1.1 Background ....................................................................................................... 1  
      1.2 Problem Description ....................................................................................... 2  
         1.2.1 Problem Statement ................................................................................. 2  
         1.2.2 Assumptions ........................................................................................... 5  
         1.2.3 Applications ......................................................................................... 6  
      1.3 Objectives ....................................................................................................... 8  
      1.4 Scope .......................................................................................................... 8  
      1.5 Significance ................................................................................................. 9  
      1.6 Report Overview ....................................................................................... 10  
CHAPTER 2 .................................................................................................................... 11  
   LITERATURE REVIEW ............................................................................................... 11  
      2.1 Introduction ................................................................................................... 11  
      2.2 Agile Manufacturing Systems ........................................................................ 11  
         2.2.1 Origin and Definition ........................................................................... 11  
         2.2.2 Why Agile Manufacturing ................................................................. 12  
         2.2.3 How to Achieve Agile Manufacturing .............................................. 13  
         2.2.4 Lean Versus Agile ............................................................................. 16  
         2.2.5 Literature on Agile Manufacturing .................................................... 16  
      2.3 Machining and Assembly System Scheduling ............................................. 17  
      2.4 Agile Manufacturing System Scheduling .................................................. 22  
         2.4.1 Product Representation ....................................................................... 23  
         2.4.2 Mathematical Formulation .................................................................. 27  
         2.4.3 Lower Bound ..................................................................................... 28  
         2.4.4 Heuristic Algorithms ......................................................................... 31  

v
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Varities of several items in the U.S. market</td>
<td>13</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Principles of lean production and agile manufacturing</td>
<td>16</td>
</tr>
<tr>
<td>Table 2.3</td>
<td>Classification scheme and number of citations for AMS research</td>
<td>17</td>
</tr>
<tr>
<td>Table 2.4</td>
<td>Scheduling measures</td>
<td>18</td>
</tr>
<tr>
<td>Table 2.5</td>
<td>Machining and assembly time sets for the problem example</td>
<td>30</td>
</tr>
<tr>
<td>Table 2.6</td>
<td>A sample of GA/SA-scheduling published research</td>
<td>41</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Machining and assembly time sets for a product example</td>
<td>45</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>The $2^3$ factorial design of the computational experiment showing the two levels of every factor</td>
<td>58</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Experimental results</td>
<td>66</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Statistics for percentage deviation from the lower bound</td>
<td>68</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Number of instances with DLB belonging to four ranges</td>
<td>69</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>FBS and %FBS for the five heuristics</td>
<td>70</td>
</tr>
<tr>
<td>Table 4.5</td>
<td>Experimental results in two replicates</td>
<td>71</td>
</tr>
<tr>
<td>Table 4.6</td>
<td>Significant factors and interactions</td>
<td>72</td>
</tr>
<tr>
<td>Table 4.7</td>
<td>fGGA ANOVA</td>
<td>74</td>
</tr>
<tr>
<td>Table B.1</td>
<td>dHA ANOVA</td>
<td>90</td>
</tr>
<tr>
<td>Table C.1</td>
<td>dGGA ANOVA</td>
<td>95</td>
</tr>
<tr>
<td>Table D.1</td>
<td>dGSA ANOVA</td>
<td>100</td>
</tr>
<tr>
<td>Table E.1</td>
<td>dHAGA ANOVA</td>
<td>105</td>
</tr>
<tr>
<td>Table F.1</td>
<td>dHASA ANOVA</td>
<td>110</td>
</tr>
<tr>
<td>Table G.1</td>
<td>fHA ANOVA</td>
<td>115</td>
</tr>
<tr>
<td>Table H.1</td>
<td>fGSA ANOVA</td>
<td>119</td>
</tr>
<tr>
<td>Table I.1</td>
<td>fHAGA ANOVA</td>
<td>122</td>
</tr>
<tr>
<td>Table J.1</td>
<td>fHASA ANOVA</td>
<td>127</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1.1 The agile manufacturing system under consideration ........................................ 3
Figure 1.2 A sample product with its assembly sequence .................................................... 4
Figure 1.3 The digraph for the sample product in Figure 1.2 ............................................. 4
Figure 1.4 Products with their digraphs ........................................................................... 5
Figure 1.5 An AMS for assembling ready-made parts ..................................................... 7
Figure 2.1 Prevailing market trends .................................................................................. 13
Figure 2.2 The structure of agile manufacturing enterprises ............................................ 14
Figure 2.3 Digraph standardization .................................................................................. 25
Figure 2.4 Converting $NN$ digraphs to a single digraph ................................................ 26
Figure 2.5 Digraph for the problem example .................................................................... 30
Figure 2.6 Heuristic Algorithm 1 ................................................................................... 31
Figure 2.7 Heuristic Algorithm 2 ................................................................................... 31
Figure 2.8 Heuristic Algorithm 3 ................................................................................... 32
Figure 2.9 Schedule obtained using Heuristic Algorithm 1 with a makespan of 38 ... 33
Figure 2.10 Schedule obtained using Heuristic Algorithm 2 with a makespan of 42. 33
Figure 2.11 Schedule obtained using Heuristic Algorithm 3 with a makespan of 39. 33
Figure 2.12 The generic GA ......................................................................................... 35
Figure 2.13 Two chromosome examples ....................................................................... 36
Figure 2.14 An example of mutating a binary chromosome ........................................... 37
Figure 2.15 An example of applying a crossover to two binary parents ......................... 38
Figure 2.16 The simple SA for the minimization of discrete optimization problems. 39
Figure 3.1 Standardized digraph of the product referred to in Table 3.1 ......................... 46
Figure 3.2 A sample chromosome for the product in Figure 3.1 .................................... 46
Figure 3.3 Schedule resulting from the chromosome in Figure 3.2 with a makespan of 46 .................................................................................................................. 46
Figure 3.4 GGA roulette wheel and elitist selection ....................................................... 47
Figure 3.5 An example of applying LOX crossover ....................................................... 48
Figure 3.6 An example of applying PMX crossover ....................................................... 49
Figure 3.7 An example of applying CO1 crossover ....................................................... 50
Figure 3.8 An example of implementing NABEL crossover ......................................... 50
Figure 3.9 An example of SWAP operator ................................................................. 51
Figure 3.10 An example of INV operator ................................................................. 51
Figure 3.11 An example of INS operator ................................................................. 51
Figure 3.12 GSA algorithm ..................................................................................... 53
Figure 3.13 An example of PS1 .............................................................................. 54
Figure 3.14 An example of PS2 .............................................................................. 54
Figure 3.15 An example of PS3 .............................................................................. 54
Figure 3.16 An example of PS4 .............................................................................. 54
Figure 3.17 An example of PS5 .............................................................................. 54
Figure 3.18 An example of PS6 .............................................................................. 54
Figure 3.19 The experimental low complexity digraph with 16 part nodes and two
assembly levels ....................................................................................................... 59
Figure 3.20 The experimental high complexity digraph with 16 part nodes and three
assembly levels ....................................................................................................... 60
Figure 3.21 The experimental low complexity digraph with 32 part nodes and two
assembly levels ....................................................................................................... 61
Figure 3.22 The experimental high complexity digraph with 32 part nodes and four
assembly levels ....................................................................................................... 62
Figure 4.1 Box plots of average DLB over 100 replicates for the eight experimental
runs .......................................................................................................................... 69
Figure 4.2 FBS for the five heuristics ....................................................................... 70
Figure 4.3 Half normal plot of effects for fGGA ....................................................... 74
Figure 4.4 Normal plot of residuals for fGGA .......................................................... 75
Figure 4.5 Plot of residuals vs. predicted values for fGGA ....................................... 76
Figure 4.6 Plot of Outlier T for fGGA ..................................................................... 77
Figure 4.7 A - C interaction graph at B = low for transformed fGGA ..................... 77
Figure 4.8 A-C interaction graph at B = high for transformed fGGA ....................... 78
Figure B.1 Half normal plot of effects for transformed dHA ................................... 89
Figure B.2 Normal plot of residuals for transformed dHA ....................................... 91
Figure B.3 Plot of residuals vs. predicted for transformed dHA .............................. 91
Figure B.4 Plot of Outlier T for transformed dHA .................................................. 92
Figure B.5 A - C interaction graph at B = low for transformed dHA ..................... 92
Figure B.6 A -C interaction graph at B = high for transformed dHA ....................... 93
Figure C.1 Half normal plot of effects for dGGA .................................................... 94
Figure C.2 Normal plot of residuals for dGGA ....................................................... 96
Figure C.3  Plot of residuals vs. predicted values for dGGA .......................................... 96
Figure C.4 Plot of Outlier T for dGGA........................................................................ 97
Figure C.5 A - C interaction graph at B = low for dGGA ........................................... 97
Figure C.6 A - C interaction graph at B = high for dGGA ........................................ 98
Figure D.1 Half normal plot of effects for dGSA............................................................. 99
Figure D.2 Normal plot of residuals for dGSA............................................................. 101
Figure D.3 Plot of residuals vs. predicted values for dGSA........................................ 101
Figure D.4 Plot of Outlier T for transformed dGSA...................................................... 102
Figure D.5 A - C interaction graph at B = low for transformed dGSA ......................... 102
Figure D.6 A - C interaction graph at B = high for transformed dGSA ................. 103
Figure E.1 Half normal plot of effects for dHAGA...................................................... 104
Figure E.2 Normal plot of residuals for dHAGA.......................................................... 106
Figure E.3 Plot of residuals vs. predicted values for dHAGA..................................... 106
Figure E.4 Plot of Outlier T for transformed dHAGA.................................................. 107
Figure E.5 A - C interaction graph at B = low for transformed dHAGA...................... 107
Figure E.6 A - C interaction graph at B = high for transformed dHAGA ................. 108
Figure F.1 Half normal plot of effects for transformed dHASA.................................... 109
Figure F.2 Normal plot of residuals for transformed dHASA..................................... 111
Figure F.3 Plot of residuals vs. predicted values for transformed dHASA............... 111
Figure F.4 Plot of Outlier T for transformed dHASA .................................................. 112
Figure F.5 A - C interaction graph at B = low for transformed dHASA ................. 112
Figure F.6 A - C interaction graph at B = high for transformed dHASA ................. 113
Figure G.1 Half normal plot of effects for transformed fHA...................................... 114
Figure G.2 Normal plot of residuals for transformed fHA......................................... 116
Figure G.3 Plot of residuals vs. predicted values for transformed fHA.................. 116
Figure G.4 Plot of Outlier T for transformed fHA....................................................... 117
Figure H.1 Half normal plot of effects for fGSA........................................................... 118
Figure H.2 Normal plot of residuals for transformed fGSA......................................... 119
Figure H.3 Plot of residuals vs. predicted values for fGSA....................................... 120
Figure H.4 Plot of Outlier T for fGSA....................................................................... 120
Figure I.1 Half normal plot of effects for transformed fHAGA ................................... 121
Figure I.2 Normal plot of residuals for transformed fHAGA...................................... 123
Figure I.3 Plot of residuals vs. predicted values for transformed fHAGA.............. 123
Figure I.4 Plot of Outlier T for transformed fHAGA .................................................. 124
Figure I.5  A - C interaction graph at $B = \text{low}$ for transformed $fHAGA$ ................... 124
Figure I.6  A - C interaction graph at $B = \text{high}$ for transformed $fHAGA$ .................. 125
Figure J.1  Half normal plot of effects for transformed $fHASA$................................. 126
Figure J.2  Normal plot of residuals for transformed $fHASA$ .................................... 127
Figure J.3  Plot of residuals vs. predicted values for transformed $fHASA$ ............... 128
Figure J.4  Plot of Outlier T for transformed $fHASA$............................................... 128
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMS</td>
<td>agile manufacturing system</td>
</tr>
<tr>
<td>ASAP</td>
<td>as soon as possible (scheduling rule)</td>
</tr>
<tr>
<td>AS</td>
<td>assembly station</td>
</tr>
<tr>
<td>GA</td>
<td>genetic algorithm</td>
</tr>
<tr>
<td>GGA</td>
<td>general genetic algorithm</td>
</tr>
<tr>
<td>GSA</td>
<td>general simulated annealing</td>
</tr>
<tr>
<td>HA</td>
<td>heuristic algorithm</td>
</tr>
<tr>
<td>HAGA</td>
<td>heuristic assisted genetic algorithm</td>
</tr>
<tr>
<td>HASA</td>
<td>heuristic assisted simulated annealing</td>
</tr>
<tr>
<td>INS</td>
<td>insert mutation operator</td>
</tr>
<tr>
<td>INV</td>
<td>inverse mutation operator</td>
</tr>
<tr>
<td>LB</td>
<td>lower bound (of the makespan value)</td>
</tr>
<tr>
<td>LOX</td>
<td>linear order crossover</td>
</tr>
<tr>
<td>PMX</td>
<td>partially mapped crossover</td>
</tr>
<tr>
<td>PS</td>
<td>perturbation scheme</td>
</tr>
<tr>
<td>SA</td>
<td>simulated annealing</td>
</tr>
<tr>
<td>TA</td>
<td>total assembly time (associated with a digraph)</td>
</tr>
<tr>
<td>TM</td>
<td>total machining time (associated with a digraph)</td>
</tr>
<tr>
<td>TSA</td>
<td>total subassembly time (associated with a digraph)</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS

$\Delta f$ deterioration of the objective function value in simulated annealing

$A_d$ dummy assembly node with assembly time equal to zero in a single digraph corresponding to several digraphs standardized

$A_i$ assembly operation i

$AP$ set of assembly operations with preceding part nodes in a digraph

$C_{\text{max}}$ makespan or the maximum completion time of all operations performed in the agile manufacturing system

$CO1$ a crossover operator

$CT(A_i)$ completion time of assembly operation $A_i$; $CT(A_N)$ is the same as $C_{\text{max}}$

$CTj(A_i)$ variable in the mixed integer programming formulation for the scheduling problem

$d$ percentage deviation from the lower bound for a heuristic

$DLB$ percentage deviation from the lower bound

$f$ frequency of resulting in the best solution for a heuristic

$\text{FBS}$ frequency of resulting in the best solution

$f_i$ evaluation function value corresponding to existing or new solution in a simulated annealing algorithm

$G$ directed graph (digraph)

$g$ current generation number in a genetic algorithm

$i, j, k$ positive integers

$IP(A_i)$ the set of all assembly operations immediately preceding $A_i$

$\text{iter}$ current iteration number in SA

$M$ Arbitrary large number in the mixed integer programming formulation for the scheduling problem
\( \min^{(k)} \) the \( k \)th smallest number in the set

\( N \) number of assembly nodes in a digraph

\( n \) number of part nodes in a digraph

\( nt \) number of tasks to be scheduled on a single machine

\( NA(A_i) \) the set of assembly operations neither preceding nor succeeding \( A_i \)

\( NN \) number of digraphs converted to a single digraph

\( P \) population set in a genetic algorithm

\( P_i \) machining operation \( i \)

\( PH_i \) set of subassembly nodes belonging to the direct path from the part node \( P_i \) to the root node \( AN_i \), \( i = 1, \ldots, n \)

\( \text{Popsize} \) population size in a GA

\( pr \) probability of accepting the new solution generated in SA

\( q \) the number of identical assembly stations in the second stage of the AMS (\( q \geq 2 \))

\( q_{ij} \) 0-1 variable in the mixed integer programming formulation for the scheduling problem

\( r \) a random number between zero and one

\( R_i \) range \( i \) of percentage deviation from the lower bound

\( T \) SA control parameter

\( T_i \) total subassembly time of subassembly nodes belonging to the direct path from \( P_i \) to the root node \( AN \)

\( t(\mathbf{\cdot}) \) time of operation \( \mathbf{\cdot} \)

\( t_i \) SA initial temperature (constant)

\( t_k \) SA cooling schedule

\( X_i \) existing or new solution in a simulated annealing algorithm

\( x_{ij} \) 0-1 variable in the mixed integer programming formulation of the scheduling problem
$y_{ij}$ auxiliary 0-1 variable in the mixed integer programming formulation of the scheduling problem

$Z$ objective function, to be minimized, in the mixed integer programming formulation for the scheduling problem considered

$z_{ij}$ auxiliary 0-1 variable in the mixed integer programming formulation of the scheduling problem

$\gamma$ SA constant parameter
CHAPTER 1
INTRODUCTION

1.1 Background

Manufacturing commenced about 5000 to 4000 B.C. with the production of articles of wood, ceramic, stone, and metal (Kalpakjian 1997). In the earliest days and for thousands of years, goods were manufactured using craft production: highly skilled workers using simple tools to produce small quantities of customized goods. Starting the late eighteenth century, many remarkable contributions, such as the division of labor and interchangeable parts, gradually transformed the shape of manufacturing from craft production to mass production: lower skilled workers using specialized machinery to produce high volumes of standardized goods. Afterwards, as the market competition intensified, mass production appeared to be wasteful. Consequently, many efforts were put forth trying to reduce the inefficiencies associated with mass production, ultimately resulting in lean production: a system that uses minimal amounts of resources to produce large volumes of high-quality goods (Stevenson 1996).

In 1991, an industry-led study, sponsored by the United States Navy Mantech Program, and supported by the Iacocca Institute at Lehigh University in the United States, was conducted in order to originate a new paradigm for successful manufacturing enterprises in the year 2006. The report of that study, entitled “21st Century Manufacturing Enterprise Strategy”, coined the term “agile manufacturing” to describe a new emerging manufacturing paradigm, expected to replace mass
production. (Groover 2001, Kidd 1994). Agile manufacturing can be defined as an enterprise level manufacturing strategy of introducing new products into rapidly changing markets or as an organizational ability to thrive in a competitive environment characterized by continuous and sometimes unforeseen change (Groover 2001). Agile manufacturing systems (AMSs) symbolize the latest development in manufacturing engineering and management.

Agile manufacturing, the latest manufacturing paradigm, needs a lot of research efforts to outline how it can be achieved. Scheduling plays a decisive role in the context of agile manufacturing due to its considerable effect on the performance of such manufacturing systems. Furthermore, Sanchez and Nagi (2001) reported that AMS scheduling received little attention in the literature. Consequently, the current research addresses the problem of scheduling an AMS.

1.2 Problem Description

1.2.1 Problem Statement

The problem considered in this research deals with scheduling an AMS that is made up of two stages. The first stage is the machining stage, consisting of a flexible machine tool that is capable of machining a variety of parts. Machined parts are successively delivered to the second stage, assembly, which is comprised of two or more identical assembly stations, where they are assembled in prescribed orders to ultimately manufacture a variety of products. Figure 1.1 shows the AMS under investigation (He and Babayan 2002).

The objective of this scheduling problem is to minimize the makespan \( C_{\text{max}} \) – the maximum completion time of all operations – of the system for given sets of machining and assembly operations. For stage 1, machining, the optimal sequence of
processing parts on the single machine tool needs to be determined. For stage 2, assembly/subassembly operations need to be assigned to assembly stations and the timing of starting to perform them needs to be determined.

![Diagram of Machining Stage and Assembly Stage](image)

**Figure 1.1 The agile manufacturing system under consideration**  
Source: He and Babayan (2002)

There are precedence relations that govern the order of assembling parts to form subassemblies and that of putting subassemblies together to result in final products. Kusiak (1989, 1990) defined a digraph, G, in order to simply communicate those precedence relations. Figures 1.2 and 1.3 (Kusiak 1990) show a sample product, which consists of three parts, and its digraph, respectively. First, part 1 and part 2 are assembled together to make subassembly A1. Then A1 is assembled with part P3 to make the final assembly A2, which is also the final product.
Generally, there are two types of products with such a structure: simple products and complex products. Simple products are represented by simple digraphs, and these are digraphs that contain no more than one assembly node in any assembly level. On the other hand, complex products are represented by complex digraphs, which contain at least two assembly nodes in at least one assembly level. Figure 1.4 shows a simple product and a complex product with their corresponding digraphs. Also, the digraph in Figure 1.3 represents another example of a simple product.
For the AMS configuration described above, the optimum solution of the simple-digraph scheduling problem can always be obtained by the method presented by Kusiak (1989). Only three heuristic algorithms, introduced by He and Babayan (2002), were proposed to solve the complex-digraph scheduling problem, and they are not guaranteed to obtain optimum schedules. Hence, the current research addresses the complex-digraph scheduling problem.

1.2.2 Assumptions

The assumptions for the problem studied are specified below:

- Machining operation times and assembly operation times are fixed and known in advance, i.e. the problem is deterministic
- At most one part is machined in the first stage of the system at any instance of time
• At most one assembly operation is performed on any assembly station in the second stage of the system and no more than one assembly station is allowed to work on any assembly operation at any instance of time
• Preemption of machining or assembly operations is not allowed
• Assembly operations must follow the precedence constraints given for every product, as shown in its digraph
• The capacity of buffer storage zone between the machining stage and the assembly stage is unlimited
• The capacities of buffer storage zones between assembly stations in the second stage are unlimited
• Set-up times are negligible compared to machining and assembly times
• Handling times between assembly stations are also negligible

1.2.3 Applications
The AMS described above fulfills the goals of agile manufacturing by allowing the manufacturing of a high variety of modular products in small batches at low costs, satisfying the customers’ requirements. This is achieved by taking advantage of machining common parts and then assembling them to make various products. The resulting products are easier to maintain, suitable for upgrades, and capable of changing functions. The advantages of such products are described by He and Kusiak (1996, 1997, and 1998) and Kusiak (2000).

The AMS described above can be found in a variety of contexts, such as automotive part manufacturing (He and Babayan 2002), furniture manufacturing, and automated assembly of ready-made parts. In automotive part manufacturing, components are machined by a multifunctional machine tool and delivered to multiple assembly stations for final assembly (He and Babayan 2002). In furniture
manufacturing, the same shop structure can be found where common and uncommon parts are machined in stage one and then sent to stage two, where they are assembled to form multiple products, such as closets and drawer cabinets of different styles and dimensions.

A third potential application of the AMS studied in this thesis is the automated assembly of ready-made parts. In this system, the machine tool in stage one is replaced with an automated storage/retrieval system (AS/RS) and automated guided vehicle (AGV) to pick up the ready made parts from their locations in the warehouse and deliver them to stage two, where they are assembled to form final products. Thus the system consists of a delivery stage and an assembly stage. Figure 1.5 schematically shows the AMS that performs the automated assembly of ready-made parts.

![Figure 1.5 An AMS for assembling ready-made parts](image-url)
The machining time set for the original shop structure is replaced with a delivery time set for the modified shop structure. This application is proposed here for the first time in conjunction with the AMS under investigation and it is verified to conform to the characteristics of modern assembly systems delineated by Kusiak (2000).

1.3 Objectives

The objectives of this research are to:

- Construct genetic algorithms (GAs) to solve the scheduling problem of the AMS under research involving complex products with the objective of minimizing the makespan
- Develop simulated annealing (SA) algorithms to solve the same problem
- Plan, design, and implement a $2^3$ factorial experiment, replicated twice, to:
  - Systematically compare the performance of proposed heuristic algorithms and the existing ones, introduced by He and Babayan (2002)
  - Decide on the significant factors affecting the performance of all heuristics studied

1.4 Scope

The current research is concerned with constructing new heuristic algorithms that are based on GAs and SA in order to apply them to the scheduling problem of an AMS with the objective of minimizing the makespan. The proposed algorithms are General Genetic Algorithm, General Simulated Annealing, Heuristic Assisted Genetic Algorithm, and Heuristic Assisted Simulated Annealing.
represents the old algorithms, introduced by He and Babayan (2002), which can solve the same problem. In order to objectively evaluate the performance of those new algorithms and systematically compare the performance of the new and old algorithms, a $2^3$ factorial experiment, replicated twice, is performed. In this experiment, three factors – the number of part nodes in a digraph; digraph complexity; and average machining time per part to average subassembly time per subassembly operation ratio– are varied over two levels. The responses selected are the percentage deviation from the lower bound and the frequency of resulting in the best solution. The results of such an experiment are expected to reveal which algorithms are most effective in solving the scheduling problem. Also, the analysis will show whether any of the three factors stated above or any possible interaction among them significantly affects the performance of any of the available five algorithms that can solve the problem specified.

1.5 Significance

The originality of this research comes from the facts that the problem studied has not been solved using GAs or SA before and that the published research dealing with machining and assembly system scheduling using GAs or SA is very limited. The results obtained by the heuristics proposed in this thesis are better than those found in literature. Consequently, this work presents scheduling methods that, when applied, improve the performance of the manufacturing system considered due to the fact that better schedules result in less manufacturing costs. Another significant contribution of this research is finding applications for the problem studied that did not exist in literature before.
1.6 Report Overview

In the remaining parts of this report, chapter two reviews the literature on AMSs; machining and assembly system scheduling; AMS scheduling; GAs; and SA. Chapter three describes the GAs and SA heuristics developed in addition to presenting the methodology of the numerical experiment that studies the heuristics proposed and the pre-existing ones to compare their performance. Chapter four presents and discusses the results of the numerical experiment. Finally, Chapter five provides the whole research conclusions, recommendations, and directions for future research.
CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

The scheduling of an agile manufacturing system (AMS), which carries out machining and assembly operations, plays an indecisive role in determining the efficiency of that system, and two of the leading optimization methods that can be used to accomplish this task effectively are GAs and SA. This chapter reviews the literature on AMSs, machining and assembly system scheduling, AMS scheduling, GAs, and SA.

2.2 Agile Manufacturing Systems

2.2.1 Origin and Definition

In 1991, a study led by thirteen industrial companies, was sponsored by the United States Navy Mantech Program and supported by the Iacocca Institute at Lehigh University in the United States with the objective of pointing out the characteristics of successful manufacturing companies in the year 2006. Ultimately, more than 100 companies participated in addition to the original thirteen, publishing the report “21st Century Manufacturing Enterprise Strategy.” The term “agile manufacturing” was coined in that report to describe a new emerging manufacturing paradigm, recognized to replace mass production (Groover 2001, Kidd 1994).

Agile manufacturing can be defined as an enterprise level manufacturing strategy of introducing new products into rapidly changing markets or as an organizational ability to thrive in a competitive environment characterized by continuous and sometimes unforeseen change (Groover 2001).
Groover (2001) presented the key findings of the year 1991 study, including:

- A new competitive environment is emerging that is forcing changes in manufacturing systems and organizations
- Agile companies will have competitive advantage in that environment
- Agility requires the integration of:
  - Flexible production technologies
  - Knowledgeable workforce
  - Management structures that encourage cooperative initiatives both internally and externally

2.2.2 Why Agile Manufacturing

In order to appreciate why agile manufacturing was introduced, the prevailing market trends need to be studied. The market trends that have been dominating since the 1970s and through the 1990s are shrinking production volumes, increasing product varieties, shortening product life cycles, and decreasing rates of repeat orders. This is shown in Figure 2.1 (Kidd 1994). These market trends resulted from the fact that manufacturers have been striving to keep their products up to date and respond to customers' demands. Furthermore, many companies nowadays are trying hard to market closer and closer to customers' individual tastes, which is resulting in plenty of choices in many cases. For example, in the US, more and more options are available for customers who are looking to purchase products such as vehicles, bicycles, and even milk, as shown in Table 2.1 (Cox and Alm 1998). As a result, agile manufacturing evolved to function as the manufacturing paradigm expected to allow companies to thrive in such a tough market.
TABLE 2.1  Varities of several items in the U.S. market
Source: Cox and Alm (1998)

<table>
<thead>
<tr>
<th>Item</th>
<th>Variety</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle models</td>
<td>Early 70s</td>
</tr>
<tr>
<td>Bicycle types</td>
<td>140</td>
</tr>
<tr>
<td>TV screen sizes</td>
<td>8</td>
</tr>
<tr>
<td>Bottled water brands</td>
<td>5</td>
</tr>
<tr>
<td>Milk types</td>
<td>16</td>
</tr>
<tr>
<td>Magazine titles</td>
<td>339</td>
</tr>
</tbody>
</table>

2.2.3  How to Achieve Agile Manufacturing

Agile manufacturing can be attained by considering it a generic structure within which every company can develop its own strategies and products. The structure is supported by three pillars: innovative management structures and organizations, a skill base of knowledgeable and empowered people, and flexible-intelligent technologies. Integration reinforces these three primary resources in order to achieve innovation and supply customers with high quality customized products. This concept is shown in Figure 2.2 (Kidd 1994). Agile manufacturing could require recourses that are beyond
the reach of a single company. Therefore sharing resources among companies becomes a must, and the competitive ability of an enterprise depends on its ability to form proper relationships with suppliers, customers, and even other competing companies (Sanchez and Nagi 2001).

Singh (1996) presented some of the characteristics of agile manufacturing according to the Agile Manufacturing Enterprise Forum, affiliated with the Iacocca Institute at Lehigh University, including:

- Greater product customization
- Rapid introduction of new or modified products

Figure 2.2 The structure of agile manufacturing enterprises
Source: Kidd (1994)
• Products that can be upgraded
• Advanced inter-enterprise networking technology
• Greater use of flexible production technologies
• Increased emphasis on knowledgeable, highly trained, empowered workers
• Interactive customer relationships
• Dynamic reconfiguration of production processes
• Rapid prototyping
• An open systems information environment
• Innovative and flexible management structures
• Product pricing based on value to the customer
• Commitment to environmentally benign operations and product designs

On the other hand, many currently held truths should be unlearned (Singh 1996), including:

• That cooperation is less desirable than succeeding on one’s own
• That labor-management relations must be adversarial
• That information is power and can be shared only to one’s detriment
• That trust makes one vulnerable
• That there are single technological solutions to complex problems
• That breakthroughs are the only targets worth aiming at
• That markets will appear by themselves once better mousetraps are invented
• That infrastructure requirements will take care of themselves once pioneers have thrown up superstructures
• That standards are constraining and their formulation dull work
• That only parts can be invented, not whole systems
2.2.4 Lean Versus Agile

It is important to note that “lean production”, which existed before 1991, and agile manufacturing are two different concepts, and the two terms should not be used interchangeably. While lean production is a collection of operational techniques concerned with doing everything with less, agile manufacturing is an overall enterprise strategy, for goods or services, focused on thriving in an environment of unpredictable change (Kidd 1994, Sanchez and Nagi 2001). In general, a company needs to be lean first to facilitate becoming agile. Table 2.2 compares the principles of lean production to those of agile manufacturing (Groover 2001).

<table>
<thead>
<tr>
<th>Lean production</th>
<th>Agile manufacturing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize waste</td>
<td>Enrich the customer</td>
</tr>
<tr>
<td>Perfect first-time quality</td>
<td>Cooperate to enhance competitiveness</td>
</tr>
<tr>
<td>Flexible production lines</td>
<td>Organize to master change</td>
</tr>
<tr>
<td>Continuous improvement</td>
<td>Strengthen the impact of people and information</td>
</tr>
</tbody>
</table>

2.2.5 Literature on Agile Manufacturing

Since 1991, agile manufacturing has been attracting an increasing amount of attention from both the academic and industrial communities (Sanchez and Nagi 2001). Some books, such as “Agile Manufacturing: Forging New Frontiers” (Kidd 1994), and many papers have studied agile manufacturing concepts in order to eventually realize an agile industry (Sanchez and Nagi 2001).

Sanchez and Nagi (2001) reviewed 73 papers, which deal with agile manufacturing, from premier scientific journals and conferences, and they introduced
a classification scheme for the survey on AMSs with nine major categories. Table 2.3 shows the classification scheme with the number of citations for every research area. It is deduced that AMS research is still in the development phase so that many AMS research projects are expected to appear. The number of citations for the whole area of production planning, scheduling, and control are only four out of 73; hence, the need for the current research is crucial.

### TABLE 2.3 Classification scheme and number of citations for AMS research
Source: Sanchez and Nagi (2001)

<table>
<thead>
<tr>
<th>AMS research area</th>
<th>Number of citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product and manufacturing systems design</td>
<td>9</td>
</tr>
<tr>
<td>Process planning</td>
<td>5</td>
</tr>
<tr>
<td>Production planning, scheduling, and control</td>
<td>4</td>
</tr>
<tr>
<td>Facilities design</td>
<td>8</td>
</tr>
<tr>
<td>Material handling and storage systems</td>
<td>1</td>
</tr>
<tr>
<td>Information systems</td>
<td>21</td>
</tr>
<tr>
<td>Supply chain</td>
<td>13</td>
</tr>
<tr>
<td>Human factors</td>
<td>3</td>
</tr>
<tr>
<td>Business practices and processes</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>73</td>
</tr>
</tbody>
</table>

### 2.3 Machining and Assembly System Scheduling

Scheduling is defined as the allocation of resources over time to perform a set of tasks (Blazewicz et al. 1994). In other words, a schedule should reveal:

- The resources responsible for doing the given set of tasks
- The exact start and finish times of performing the given set of tasks
Scheduling aims to minimize a certain measure, and Table 2.4 (Brah et al. 1991) lists the most common of such measures. In the context of agile manufacturing, one of the most important scheduling measures is the makespan, since minimum makespans are expected to lead to fastest product deliveries to customers.

**TABLE 2.4 Scheduling measures**  
Source: Brah et al. (1991)

<table>
<thead>
<tr>
<th>Category</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures related to completion time</td>
<td>Maximum completion time (makespan)</td>
</tr>
<tr>
<td></td>
<td>Maximum flow time</td>
</tr>
<tr>
<td></td>
<td>Total completion time</td>
</tr>
<tr>
<td></td>
<td>Total flow time</td>
</tr>
<tr>
<td></td>
<td>Mean completion time</td>
</tr>
<tr>
<td></td>
<td>Mean flow time</td>
</tr>
<tr>
<td></td>
<td>Weighted sum of completion time</td>
</tr>
<tr>
<td></td>
<td>Weighted sum of flow time</td>
</tr>
<tr>
<td></td>
<td>Jobs waiting time</td>
</tr>
<tr>
<td></td>
<td>Weighted job waiting time</td>
</tr>
<tr>
<td>Measures related to due dates</td>
<td>Maximum lateness</td>
</tr>
<tr>
<td></td>
<td>Maximum tardiness</td>
</tr>
<tr>
<td></td>
<td>Maximum earliness</td>
</tr>
<tr>
<td></td>
<td>Total lateness</td>
</tr>
<tr>
<td></td>
<td>Total tardiness</td>
</tr>
<tr>
<td></td>
<td>Total earliness</td>
</tr>
<tr>
<td></td>
<td>Mean lateness</td>
</tr>
<tr>
<td></td>
<td>Mean tardiness</td>
</tr>
<tr>
<td></td>
<td>Mean earliness</td>
</tr>
<tr>
<td></td>
<td>Weighted sum of lateness</td>
</tr>
<tr>
<td></td>
<td>Weighted sum of tardiness</td>
</tr>
<tr>
<td></td>
<td>Weighted sum of earliness</td>
</tr>
<tr>
<td></td>
<td>Number of tardy jobs</td>
</tr>
<tr>
<td></td>
<td>Number of early jobs</td>
</tr>
<tr>
<td></td>
<td>Number of jobs in the system</td>
</tr>
<tr>
<td>Measures related to cost &amp; utilization</td>
<td>Machine idle time</td>
</tr>
<tr>
<td></td>
<td>Weighted machine idle time</td>
</tr>
<tr>
<td></td>
<td>Manpower idle time</td>
</tr>
<tr>
<td></td>
<td>Manpower weighted idle time</td>
</tr>
<tr>
<td></td>
<td>Utilization or mean utilization</td>
</tr>
<tr>
<td></td>
<td>Setup time</td>
</tr>
</tbody>
</table>
It is not usually easy to solve a scheduling problem because the number of different possible schedules is, most of the time, very large. In other words, scheduling involves complex combinatorial optimization. For example, scheduling a set of $nt$ tasks on a single machine involves selecting a schedule from $nt!$ possible different schedules. For example, if there are ten tasks to be scheduled, the number of possible schedules equals $10!$ or $3.6288 \times 10^6$. Consequently, polynomial time algorithms cannot solve most scheduling problems. These problems are called NP-hard (Pinedo 1995). So even a computer usually needs unacceptably large amounts of time, such as years, to optimally solve the medium or large sized of such scheduling problems (Parker 1995).

Most scheduling literature deals with machining and assembly shops independently (Yokoyama 2001). For instance, Blazewicz (1991) reviewed mathematical programming formulations for machining scheduling without assembly while Sawik (1999) dealt with assembly system scheduling without machining. Nevertheless, research on hybrid systems that include both machining and assembly stations is indispensable, specially for agile manufacturing. Since the AMS investigated in this thesis belongs to that category, literature on the scheduling of machining and assembly systems is reviewed next.

Kusiak (1989 and 1990) presented an aggregate scheduling problem of a machining subsystem and an assembly subsystem, which are linked by a material handling carrier, such as an AGV. The problem is represented by complex and simple digraphs. Optimal scheduling algorithms to solve the single product, the single batch, the multiple products, and the multiple batches were developed. That research investigated a system with a single machine at stage one and only a single assembly
station at stage two, so more complex manufacturing system structures need to be investigated.

Fry et al. (1989) studied the effects of product structure and sequencing rule on machining and assembly shop performance. Ten bill-of-materials (BOMs) and fourteen dispatching rules were selected and experimented with. Mean flow time, mean tardiness, and mean absolute lateness were the shop measures chosen. The results indicate that the interaction between the product BOM and the dispatching rule is significant. However, the effects of factors, such as machining and assembly time sets, on the shop performance were not examined.

Doctor et al. (1993) addressed the problem of scheduling multiple jobs in a machining and assembly shop. The objective was to maximize the machine utilization subject to satisfying job due date requirements. They developed a heuristic algorithm and presented an example problem in addition to a computational study with sub-optimal results. However, the effects of factors, such as product structure, on the shop performance were not examined.

Lee and Vairaktarakis (1998) compared the throughput performance of several flexible machining and assembly shops. They presented heuristic algorithms and worst-case error bounds, showing that the performance of their heuristics is near optimal. Nevertheless, the products considered include only two succeeding operations: one at the machining stage and one at the assembly stage. The complex product structure considered in this thesis changes the nature of the problem considerably since it fulfills the goals of agile manufacturing.

Yokoyama (2001) considered the scheduling problem of machining and assembly operations in a production system, with the objective of minimizing the weighted sum of completion time for all products scheduled. In that paper, the author
provided a technique to get better lower bounds and a calculation procedure using the branch and bound method to yield optimal and near optimal solutions. The machining stage is a flow shop and the assembly stage consists of a single assembly station. The structure considered in the current research is different since it consists of a single machine at stage one and several identical assembly stations at stage two.

Mohanasundaram et al. (2002) investigated the performance of scheduling rules—existing and proposed—for shops that manufacture multilevel jobs, with the objective of minimizing the maximum flow time and standard deviation of flow time as well as maximum tardiness and standard deviation of tardiness. They conducted a simulation study for a shop consisting of nine work centers, each work center consisting of two identical machines and two identical assembly stations. It was shown that the proposed rules are quite effective for scheduling the system studied. However, in the context of agile manufacturing the makespan scheduling measure is of prime importance, so it is studied in the current research.

Sun et al. (2003) considered minimizing the makespan in the fixed three-machine assembly-type flow shop-scheduling problem. They proposed a series of heuristic algorithms to solve that problem. Only two components are manufactured at stage one, which consists of two non-identical machines, and then assembled by a single assembly station at stage two. Such a shop structure does not permit the manufacturing of complex products, which are crucial for agile manufacturing and studied in the current research.

The next section reviews three journal papers that deal with either the same problem or very similar problems to the one studied in the current research.
2.4 Agile Manufacturing System Scheduling

He et al. (2001) defined, formulated, and solved scheduling problems associated with the assembly-driven product differentiation strategy in an AMS. The system consists of two stages: machining and assembly. At the machining stage, multiple identical machines produce parts, which are then assembled at the assembly stage to form customized products. The product structures are represented by digraphs, and the objective for scheduling the system is to minimize the makespan. Two heuristic algorithms that provide optimal and near optimal solutions were developed. First, an optimal aggregate schedule is determined by solving a two-machine flowshop problem. Next, the optimal aggregate schedule is decomposed by solving a simple integer-programming model. The computational experiment shows that the heuristics result in optimal and near-optimal solutions. However, the factors that significantly affect whether the optimal solution is reached or not were not investigated. Also, the results provided do not show a comparison of the performance of the heuristics, since the two developed heuristics were not applied comparatively in that research.

He and Babayan (to appear) presented a general framework for AMS scheduling utilizing the agent-based approach, a method referred to in that paper. They initiated a methodology for the development of a negotiation mechanism to improve scheduling flexibility and robustness. Lower and upper bounds for measuring the effectiveness of the scheduling system were also constructed. The developed model integrates data associated with several entities within a scheduling system, resulting in optimal and near optimal schedules. The AMS consists of two stages: machining and assembly. At the machining stage, multiple identical machines produce parts, which are then assembled by multiple identical assembly stations at the assembly stage to form customized products. The product structures are represented
by digraphs, and the objective for scheduling the system is to minimize the makespan. The limitation on that research is that the factors that significantly affect whether the optimal solution is reached were not investigated.

He and Babayan (2002) made a few contributions regarding the scheduling problem addressed in the research. They presented a unifying way of product representation, mathematical formulation, lower bound computation, and four heuristics to solve the problem. They ultimately presented computational experience showing that their proposed heuristics result in optimal and near optimal solutions. However, they did not investigate the factors affecting whether the optimum is reached or not. The items presented in that paper are reviewed next. Also, it is worth noting that the three heuristics proposed in that paper are experimented with in Chapter four in this thesis.

2.4.1 Product Representation

As mentioned in Chapter one, the precedence relations among the parts and assemblies for the products manufactured in the AMS are represented by digraphs and this thesis deals only with complex products. Complex products are represented by complex digraphs, which contain at least two assembly nodes in at least one assembly level. Since these digraphs could take considerably diverse shapes according to the structure of the product studied, it was necessary to decide on a standardization procedure that makes all digraphs look similar. This strategy is necessary to facilitate solving the scheduling problem. First, dummy assembly nodes with assembly time of zero are inserted so that only assembly nodes at the highest assembly level have preceding part nodes. Assembly levels are counted from right to left with the root node (final assembly) at assembly level one. Second, all part nodes that precede the same subassembly node are aggregated together to form a single part node. This step
will not affect the problem since the AMS has a single machine at stage one. The machining time for that aggregate part node becomes the sum of machining times of the aggregated nodes. Figure 2.3 shows an example of the digraph standardization procedure described above. First, dummy node $A_2$ is added to satisfy condition one. Second, nodes $P'_1$ and $P'_2$ and nodes $P'_4$ and $P'_5$ are replaced by nodes $P_1$ and $P_3$, respectively, to satisfy condition two.

Moreover, for the case of manufacturing $NN$ products simultaneously in the AMS, all product digraphs are converted to a single digraph by adding a dummy assembly node $A_d$ with assembly time $t(A_d)$ of zero. In scheduling the manufacturing of $NN$ products in the AMS, those products, separately, can be either complex or simple. However, the resulting single graph – according to the method outlined above – will always be complex. This is true since it will at least have two assembly nodes at the same assembly level. The approach of converting $NN$ digraphs to a single digraph is depicted in Figure 2.4. Following this step, the resulting single digraph must be subjected to the application of the digraph standardization procedure (Figure 2.3) outlined at the beginning of this section, so that it is compatible with the available solution methods.
Figure 2.3 Digraph standardization

(a) original digraph

(b) after adding assembly node A2

(c) after aggregating part nodes
Figure 2.4 Converting \( NN \) digraphs to a single digraph

Source: He and Babayan (2002)

Note: The resulting single digraph must be subjected to the application of the digraph standardization procedure (Figure 2.3)
2.4.2 Mathematical Formulation

The mixed-integer programming formulation for the problem under investigation was introduced by He and Babayan (2002) as follows:

\[
\min Z = CT(A_N) 
\]

subject to

\[
CT(A_i) - t(A_i) \geq \sum_{j=1}^{k} q_{ij} t(P_j) + t(P_i) \quad \text{for } i, A_i \in AP, k = |AP| 
\]

\[
q_{ij} + q_{ji} = 1 \quad \text{for } i, j, A_i \in AP, A_j \in AP, i \neq j 
\]

\[
CT(A_i) \geq CT(A_j) + t(A_i) \quad \text{for } i = 1, \ldots, N, A_i \in IP(A_i) 
\]

\[
\sum_{j=1}^{q} x_{ij} = 1 \quad i = 1, \ldots, N 
\]

\[
\sum_{j=1}^{q} CT_j(A_i) = CT(A_i) \quad i = 1, \ldots, N - 1 
\]

\[
x_{ij} \leq M (1 - z_{ij}) 
\]

\[
CT(A_i) - CT_j(A_i) \leq M z_{ij} \quad \text{for } i = 1, \ldots, N - 1, j = 1, \ldots, q 
\]

\[
CT_j(A_i) - CT_j(A_k) + x_{ij} t(A_k) \leq M y_{ikj} 
\]

\[
CT_j(A_k) + x_{ij} t(A_i) - CT_j(A_i) \leq M (1 - y_{ikj}) 
\]

for \( i = 1, \ldots, N - 1, j = 1, \ldots, q, A_k \in NA(A_i) \)

\[
x_{ij} = \begin{cases} 
1, & \text{if assembly operation } i \text{ is assigned to assembly station } j \\
0, & \text{otherwise} 
\end{cases} 
\]

\[
q_{ij} = \begin{cases} 
1, & \text{if machining operation } i \text{ is schedule before machining operation } j \\
0, & \text{otherwise} 
\end{cases} 
\]

\[
CT_j(A_i) = \begin{cases} 
CT(A_i), & \text{if assembly operation } i \text{ is assigned to machine } j \\
0, & \text{otherwise} 
\end{cases} 
\]

\[
y_{ikj} \text{ and } z_{ij} = 0 \text{ or } 1 
\]
Variables that are not defined in (2.11) to (2.14) are non-negative \hspace{1cm} (2.15)

The objective function (2.1) minimizes the makespan, the completion time of the final assembly operation $A_N$. Constraint (2.2) guarantees that every assembly operation with a preceding part node will start only after the corresponding machining operation is completed. Constraint (2.3) ensures that for any two part-nodes, $P_i$ and $P_j$, either $P_i$ is worked on before $P_j$ or $P_j$ is worked on before $P_i$. Constraint (2.4) guarantees that assembly operations having immediately subassembly operations will not start before the preceding subassembly operations are completed. Constraint (2.5) ensures that every assembly operation is worked on by only one assembly station. Constraints (2.6) – (2.8) guarantee that if assembly operation $A_i$ is worked on by assembly station $j$, then the completion time of $A_i$ on $j$ is equal to the completion time of that assembly operation $A_i$. Constraints (2.9) and (2.10) ensure that there is no overlapping of assembly operations on any assembly station. Constraints (2.11) to (2.15) define the allowed values for all variables.

The model (2.1 – 2.15) is difficult to solve optimally, specially for medium and large sized problems, due to its computational complexity, as the problem studied can be considered as an extended parallel machine makespan scheduling problem (He and Babayan 2002). Consequently, the computations of a lower bound and several heuristics are introduced in the next sections with examples.

2.4.3 Lower Bound

The lower bound presented in this section was introduced by He and Babayan (2002) in order to assist in evaluating the quality of solutions resulting from the four heuristics they developed. The lower bound for a certain problem instance is a value below which no makespan value exists for any possible schedule of that particular
instance. The importance of the lower bound cannot be overestimated. The lower bound computed for a given instance can point out if the optimum solution is reached since if a solution is equal to the lower bound, then that solution is optimum. Moreover, the percentage deviation from the lower bound is an effective measure that can be used to compare the quality of solutions generated by different algorithms.

The lower bound (LB) is stated in equations (3.16) – (3.18).

\[
LB_1 = t(A_N) + TM + \min \left\{ \sum_{A_j \in PH_i} t(A_j) \right\} \\
LB_2 = t(A_N) + \frac{\text{TSA} + \sum_{k=1}^{q} (q-k+1) \min_{i=1}^{n} \{t(P_i)\}}{q} \\
LB = \max \{LB_1, LB_2\}
\]  

Where:

\( A_N \) = root assembly node in a digraph

\( TM \) = total machining time associated with a digraph

\( PH_i \) = set of subassembly nodes belonging to the direct path from the part node \( P_i \) to the root node \( A_N \), \( i = 1, \ldots, n \), \( n \) = number of part nodes in a digraph

\[ \sum_{A_j \in PH_i} t(A_j) = \text{total subassembly time for } PH_i \]

\( A_j \) = assembly operation \( j \)

\( \text{TSA} \) = total subassembly time associated with a digraph

\( q \) = the number of identical assembly stations in the second stage of the AMS

\( \min_{i=1}^{n} \{t(P_i)\} = \text{the } k^{th} \text{ smallest number in the set } \{t(P_i)\} \)
As an illustration, the lower bound for the scheduling problem associated with the digraph in Figure 2.5 and a manufacturing system with a structure of a single machine in stage one and two assembly stations in stage two \((q = 2)\) is computed next. Table 2.5 lists the given machining and assembly time sets (He and Babayan 2002).

![Figure 2.5 Digraph for the problem example](image)

**TABLE 2.5** Machining and assembly time sets for the problem example

<table>
<thead>
<tr>
<th>Part</th>
<th>(P_1)</th>
<th>(P_2)</th>
<th>(P_3)</th>
<th>(P_4)</th>
<th>-</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Machining Time</strong></td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Subassembly / Assembly</strong></td>
<td>(A_1)</td>
<td>(A_2)</td>
<td>(A_3)</td>
<td>(A_4)</td>
<td>(A_5)</td>
<td>(A_6)</td>
<td>(A_7)</td>
</tr>
<tr>
<td><strong>Assembly Time</strong></td>
<td>7</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

\[
TM = 3 + 4 + 2 + 5 = 14; \quad TSA = 7 + 11 + 9 + 8 + 10 + 10 = 55
\]

\[
PH_1 = \{A_1, A_5\} \quad \text{total subassembly time for } PH_1 = 7 + 10 = 17
\]

\[
PH_2 = \{A_2, A_5\} \quad \text{total subassembly time for } PH_2 = 11 + 10 = 21
\]

\[
PH_3 = \{A_3, A_6\} \quad \text{total subassembly time for } PH_3 = 9 + 10 = 19
\]
PH₄ = {A₄, A₆}  \implies \text{total subassembly time for PH₄} = 8 + 10 = 18

LB₁ = 6 + 14 + \min\{17, 21, 19, 18\} = 37

LB₂ = 6 + (55 + 2 \times 2 + 1 \times 3) / 2 = 37

LB = \max\{LB₁, LB₂\} = 37.

2.4.4 Heuristic Algorithms

The three heuristic algorithms that are designed to work on standardized digraphs, described in section 2.4.1, which were proposed by He and Babayan (2002) are reviewed in Figures 2.6-2.8. This is followed by a description of the fourth heuristic algorithm.

Heuristic Algorithm 1

1. Schedule the machining operations at the machining stage with the shortest processing time first (SPTF) rule
2. Schedule assembly operations on available assembly stations as soon as possible (ASAP)

Heuristic Algorithm 2

1. Apply Algorithm 1 in Kusiak (1989) – presented in Appendix A– to obtain an aggregate minimum makespan schedule. Schedule part nodes on the machine at the machining stage according to the aggregate schedule
2. Schedule assembly operations on available assembly stations ASAP

Figure 2.6 Heuristic Algorithm 1
Source: He and Babayan (2002)

Figure 2.7 Heuristic Algorithm 2
Source: He and Babayan (2002)
Heuristic Algorithm 3

1. For each part node $P_i$, calculate total subassembly time $T_i$ of subassembly nodes belonging to the direct path from $P_i$ to the root node $A_N$.
2. Schedule part nodes on the machine at the machining stage with the descending order of $T_i$.
3. Schedule assembly operations on available assembly stations ASAP.

Figure 2.8 Heuristic Algorithm 3
Source: He and Babayan (2002)

Heuristic Algorithm 4 is for scheduling the manufacturing of several products simultaneously in the AMS, which is achieved by converting the several digraphs to a single digraph by adding a dummy node with assembly time of zero, then applying one of the three heuristic algorithms outlined above. Since it does not significantly add new information, Heuristic Algorithm 4 is not considered in the next example.

The calculations of Heuristic Algorithms 1–3 are shown by applying them to the data presented in Figure 2.5 and Table 2.5. Applying Heuristic Algorithm 1, the sequence for part machining is: $P_3$, $P_1$, $P_2$, $P_4$. The schedule has a makespan of 38 and is shown in Figure 2.9. Applying Heuristic Algorithm 2, the sequence for part machining is: $P_3$, $P_4$, $P_1$, $P_2$. The schedule has a makespan of 42 and is shown in Figure 2.10. Applying Heuristic Algorithm 3, the sequence for part machining is: $P_2$, $P_3$, $P_4$, $P_1$. The schedule has a makespan of 39 and is shown in Figure 2.11.

The lower bound computed for the same instance is 37. By exhaustive search, it was found out that the schedule with the makespan of 38, resulting from Heuristic Algorithm 1, is optimal although its makespan is not equal to the lower bound. This illustrates the point that for a solution to be optimal, its makespan does not necessarily equal the lower bound.
Figure 2.9 Schedule obtained using Heuristic Algorithm 1 with a makespan of 38

Figure 2.10 Schedule obtained using Heuristic Algorithm 2 with a makespan of 42

Figure 2.11 Schedule obtained using Heuristic Algorithm 3 with a makespan of 39
After solving few instances using Heuristic Algorithm 1 – 3, it was realized that although sometimes the optimum solution is reached, some other times solutions that significantly deviate from their lower bounds are obtained. This indicates the possibility of developing other heuristics to obtain better solutions, and two of the most effective optimization methods that can be utilized for that are genetic algorithms (GAs) and simulated annealing (SA), which are reviewed in the next section.

2.5 Genetic Algorithms and Simulated Annealing

2.5.1 Genetic Algorithms

GAs, as they are known today, were first proposed by John Holland (1975) in his book, “Adaptation in Natural and Artificial Systems.” They are search algorithms that are based on the biological laws of natural selection and genetics, such as ‘survival of the fittest.’ GAs provide a very powerful method that efficiently utilizes historical information to evaluate new search points with expected better performance (Goldberg 1989).

GAs are proven to provide robust search in complex spaces, and many papers and dissertations establish their validity in function optimization. Moreover, the number of applications of GAs is growing rapidly because these algorithms are simple yet powerful in their search for improvement. Unlike other optimization techniques, they are not restricted by assumptions about the search space, such as continuity, existence of derivatives, unimodality, and others (Goldberg 1989). Michalewicz (1992) provided a highly succinct verbal description of GAs:

“In such algorithms a population of individuals (potential solutions) undergoes a sequence of unari (mutation type) and higher order (crossover type) transformations. These individuals strive for survival: a selection scheme, biased towards fitter individuals, selects the next generation. After some
number of generations, the program converges – the best individual hopefully represents the optimum solution.”

In a generic GA, first the generations counter, \( g \), is set equal to zero, and the initial population is generated and evaluated. A generation, \( P(g) \), consists of a number of individuals, known as chromosomes, which mathematically represent potential solutions to the problem being solved. After these initial steps, the loop of generations starts, which ends with a certain termination condition, such as a pre-specified number of generations. In every generation, the generations counter increases by one, and the current generation’s population is selected from the previous generation’s population, using some sort of a selection scheme, such as roulette wheel selection. Then the current generation’s population undergoes reproduction, which typically occurs by mutations and crossovers, and then the reproduced population is evaluated. The generic GA is shown in Figure 2.12, as presented by Michalewicz (1992).

```
Generic GA

begin
  g \rightarrow 0
  initialize \( P(g) \)
  evaluate \( P(g) \)
  while (not termination-condition) do
    begin
      g \rightarrow g + 1
      select \( P(g) \) from \( P(g-1) \)
      recombine \( P(g) \)
      evaluate \( P(g) \)
    end
  end

Figure 2.12 The generic GA
Source: Michalewicz (1992)
```
GAs have five elements (Goldberg 1989; Michalewicz 1992; Venkataraman 2002), which are reviewed next:

- Representation
- Initial population
- Evaluation function
- Genetic operators
- Control Parameters

Representation deals with how the GA chromosomes are mathematically symbolized, and in general either binary encoding or real encoding is used. Binary encoding is associated with the earlier work of GAs, but real encoding is recommended for most problems (Venkataraman 2002). For example, Figure 2.13 shows two chromosome examples, of the types described above, which can be used to represent the sequence of machining parts on a single machine. In that example, every entry in a chromosome represents a part number. It is worth mentioning that several methods of decoding chromosomes into solutions exist and that the efficiency of every method is problem dependent. For example, Shittu et. al (2003) used binary encoding to apply GAs to the deterministic time-varying fixed quantity lot-sizing problem.

**a) binary coded chromosome**  
<100101, 000101, ..., 101011>

**b) real coded chromosome**  
<69, 5, ..., 43>

*Figure 2.13 Two chromosome examples*

How to initialize the GA population is a key decision. Initializing the population can be done by randomly generating a number of chromosomes, each one representing a different solution to the problem; by generating chromosomes using
sub-optimal heuristics; and by a combination of these methods (Wang and Zheng 2003; Nepalli et al. 1996).

The evaluation function is the objective function associated with the problem studied, such as the makespan, maximum lateness, or machine idle time for a manufacturing scheduling problem. Genetic operators consist of (Michalewicz 1992):

- **Selection**: temporary population is formed by probabilistically selecting individuals from the current population. Fitter individuals have more chance of being selected. Temporary population undergoes crossover and mutation to yield the next generation.

- **Mutation**: this genetic operator randomly alters one or more positions in a chromosome from the temporary population (generation) to yield a new individual for the next generation, with a probability equal to a pre-specified mutation rate. When applying a mutation, a random number between zero and one is generated; if that number is below the mutation rate, the chromosome picked undergoes mutation; otherwise, the chromosome is copied to the new population as is. Figure 2.14 shows an example of mutating a binary chromosome. In this example, the shaded, bold digit is changed from 1 to zero.

```
Original chromosome: <10111, 000101, 101011>
Mutated chromosome: <100111, 000101, 101011>
```

**Figure 2.14 An example of mutating a binary chromosome**

- **Crossover**: this genetic operator is performed by choosing two individuals (parents) from the temporary generation to yield two new individuals (children) for the next generation, with a probability equal to a pre-specified crossover rate. When applying a crossover, a random number between zero and one is
generated; if that number is below the crossover rate, the chromosomes picked undergo crossover; otherwise, the chromosomes are copied to the new population as they are. Figure 2.15 shows an example of applying a crossover to two binary parents, where the xs and ys are 0s and 1s but coded in that way for clarity. As described by Reeves (1995), crossover and mutation criteria should be suitable for the problem representation to avoid illegitimate offsprings (generated chromosomes).

Parents: \(<010101, 110101, 000011> & <xxxxyy, xxyxxy, yxyxyy>\)

Children: \(<010101, xxxxxy, yxyxyy> & <xxxxyy, 110101, 000011>\)

**Figure 2.15 An example of applying a crossover to two binary parents**

GA control parameters include the population size, stopping criteria, crossover rate, and mutation rate. Selection of such parameters has an effect on the performance of GAs (Yoon and Ventura 2002). It is import to note that several GA forms that are different from the generic GA in Figure 2.12 are encountered in literature, with the intention of improving performance.

### 2.5.2 Simulated Annealing

Using SA to solve combinatorial optimization problems was first introduced by Kirkpatrick et al. (1983), and the applications studied include problems dealing with the physical design of computers, such as the placement of components to minimize wiring connections. The annealing term refers to the process of gradually cooling metals after raising their temperature to achieve a distinct crystalline configuration at their minimum energy state, and SA aims to reach the minimum state (optimum) of a
certain function. Nevertheless, using SA as an optimization method is mostly heuristic. Fink and Vob (1999) depicted SA as follows:

“Successively, a candidate move is randomly selected; this move is accepted if it leads to a solution with a better objective function value than the current solution, otherwise the move is accepted with a probability that depends on the deterioration $\Delta f$ of the objective function value. The probability of acceptance is computed according to the Boltzman function as $e^{(-\Delta f/T)}$, using a temperature $T$ as a control parameter.”

The Boltzman function is a probability distribution function. The term $\Delta f$ is the difference in the objective function value due to the proposed move, and the term $T$ is a constant that is selected as a control parameter in SA. Analogous to the simple SA for the maximization of continuous optimization problems presented by Venkataraman (2002), the simple SA for the minimization of discrete optimization problems is shown in Figure 2.16.

<table>
<thead>
<tr>
<th>Simple SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

Figure 2.16 The simple SA for the minimization of discrete optimization problems
SA has the ability to escape from local optima by accepting solutions that temporarily deteriorate the objective function. This results in great potential of obtaining high quality solutions when SA is applied to various kinds of combinatorial optimization problems (Zegordi et al. 1995). Like GAs, SA implementation depends on the problem representation, initial solution, evaluation function, and control parameters. Deciding on proper control parameters include determining the values of the constant parameters in Figure 2.4. Furthermore, how the new solution is computed in SA should be compatible with the problem representation. It is worth mentioning that several variants of the simple SA in Figure 2.4, with the objective of improving its performance, are established in literature.

2.5.3 Machining and Assembly System Scheduling Using Genetic Algorithms and Simulated Annealing

Conducting the literature review for the current research, it has been noticed that most GA/SA-scheduling published research studies machining systems and assembly systems separately. Table 2.7 presents a categorization for a sample of the literature on manufacturing scheduling using GAs or SA. It is worth mentioning that many published papers study the famous assembly line balancing (ALB) problem, which is categorized under the scheduling area of assembly, using GAs or SA although Table 2.6 does not cite many such papers because this topic is significantly different from the current research topic. It is important to highlight that research dealing with the scheduling of integral machining and assembly systems using GAs or SA is remarkably scarce. Nevertheless, two journal papers that belong to that category— the same category to which the current research belongs— are reviewed next.
TABLE 2.6 A sample of GA/SA-scheduling published research

<table>
<thead>
<tr>
<th>Author</th>
<th>Algorithm</th>
<th>Scheduling area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GA</td>
<td>SA</td>
</tr>
<tr>
<td>Ishibuchi et al. (1995)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Reeves (1995)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Zegordi et al. (1995)</td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Cheng et al. (1996)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Kim and Kim (1996)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Nepalli et al. (1996)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Shi (1997)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Portmann et al. (1998)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Cheng et al. (1999)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Tian et al. (1999)</td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Khoo et al. (2000)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Negenman (2001)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Wang and Zheng (2001)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Kho and Loi (2002)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Pongcharoen et al. (2002)</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

Kim and Kim (1996) considered a short term scheduling problem, in a machining and assembly shop environment, for products with multi-level structures, with the objective of minimizing the weighted sum of earliness and tardiness of parts, subassemblies, and final products. They applied GAs and SA and compared their performance to a commonly used method called finite loading. The results show that GAs and SA perform better than the conventional algorithm and that SA outperforms GAs. However, it was assumed that scheduling time is divided into periods and that an item can be processed only within a single period. Therefore that problem solving
Pongcharoen et al. (2002) determined efficient GA parameters for scheduling the machining and assembly of complex products in a capital good manufacturing environment. The algorithm aims to minimize the penalties due to the early supply of components and the late delivery of final products simultaneously, considering capacity utilization. Moreover, a factorial experiment has been conducted to determine appropriate values for the GA control parameters, including the crossover rate and the number of generations, within a given execution time. The proposed GA provides significant improvements to the delivery performance and resource utilization in a case study. However, the GA proposed utilizes a representation of the problem which, when subjected to genetic operators, result in illegitimate schedules of the system. Although a repair process is presented to overcome this deficiency, this approach is not favorable, especially in agile manufacturing, because it adds computational difficulty to the problem, increasing the computer time required to reach acceptable solutions.

2.6 Conclusion

An agile company is one that is capable of thriving in an environment of unpredictable change in markets, technologies, business relations and all other aspects of a company for goods or services. The research dealing with AMSs is still in an early stage, and little research has been done to investigate innovative techniques for scheduling AMSs. On the other hand, GAs and SA provide powerful heuristics for solving difficult scheduling problems. The problem investigated in the current research appeared in a single published journal paper in which three heuristics were proposed,
sometimes resulting in optimum solutions. However, the factors affecting whether those heuristics arrive at the optimum or not were not investigated. Consequently, the current research proposes four novel heuristics based on GAs and SA in an attempt to obtain better solutions. Also, an experiment is conducted to study the proposed and existing heuristics to determine the factors that significantly affect their performance. The methodology is presented in the next chapter.
CHAPTER 3
METHODOLOGY

3.1 Introduction

It is noted from the literature review that the algorithms introduced by He and Babayan (2002) can solve the problem studied and they even sometimes result in optimum solutions, solutions with makespans equal to the corresponding lower bounds. It is so known because when a makespan is equal to the lower bound, then the solution associated with this makespan is optimal. However, along with that, sometimes they result in solutions that are quite far away from the lower bound, which indicates the possibility of finding better solutions. For this reason, four algorithms are proposed in this research.

The performance of these proposed algorithms will be compared to the existing ones through a designed experiment. In order to facilitate the comparison of the proposed algorithms to the old three algorithms in the experiment, the latter will be counted as only one algorithm, calling it Heuristic Algorithm (HA). This means that the HA solution for a specific instance is the minimum of solutions resulting from all three He and Babayan heuristics.

This chapter presents the four proposed heuristics: General Genetic Algorithm (GGA), General Simulated Annealing (GSA), Heuristic Assisted Genetic Algorithm (HAGA), and Heuristic Assisted Simulated Annealing (HASA). These four heuristics are novel variants of GAs and SA that are proposed for the first time in this research as solution methods to the problem studied. In addition, the design, planning, and procedure of the computational experiment are described.
3.2 General Genetic Algorithm

The overall organization of GGA is similar to the generic GA outlined in Figure 2.12. The five basic components of GGA - representation, initial population, evaluation function, genetic operators, and control parameters - are depicted next.

The representation of the problem is realized by a string of unrepeated numbers, of size $n$, that determines the sequence of machining parts in the first AMS stage coupled to a simple rule: schedule assembly operations on available assembly stations ASAP. At every instant of time when there are vacant assembly stations, all assembly operations are scanned, the ones with satisfied precedence constraints are identified, and the ready assembly operations are assigned to vacant assembly stations accordingly. When there are more ready assembly operations than vacant assembly stations, the assembly operations with higher orders (closer to the left end then to the top of the digraph) are scheduled first. Representing the problem in this way confines the search space to $n!$, where $n$ is the number of part nodes in a standardized digraph.

For example, for the machining and assembly time sets in Table 3.1 and the standardized digraph in Figure 3.1, the chromosome in Figure 3.2 represents a schedule (Figure 3.3) for an AMS with a single machine and two assembly stations (AS1 and AS2). The sequence of machining parts on the single machine is as dictated by the chromosome. The assembly stations schedule is developed as described above.

**TABLE 3.1 Machining and assembly time sets for a product example**

<table>
<thead>
<tr>
<th>Part</th>
<th>P₁</th>
<th>P₂</th>
<th>P₃</th>
<th>P₄</th>
<th>P₅</th>
<th>P₆</th>
<th>P₇</th>
<th>P₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Assembly</td>
<td>A₁</td>
<td>A₂</td>
<td>A₃</td>
<td>A₄</td>
<td>A₅</td>
<td>A₆</td>
<td>A₇</td>
<td>A₈</td>
</tr>
<tr>
<td>Time</td>
<td>3</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Assembly</td>
<td>A₉</td>
<td>A₁₀</td>
<td>A₁₁</td>
<td>A₁₂</td>
<td>A₁₃</td>
<td>A₁₄</td>
<td>A₁₅</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.1 Standardized digraph of the product referred to in Table 3.1

Figure 3.2 A sample chromosome for the product in Figure 3.1

<table>
<thead>
<tr>
<th>Machine</th>
<th>P6</th>
<th>P1</th>
<th>P4</th>
<th>P8</th>
<th>P3</th>
<th>P7</th>
<th>P2</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>AS2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
</tr>
</tbody>
</table>

Figure 3.3 Schedule resulting from the chromosome in Figure 3.2 with a makespan of 46
The GGA initial population is generated randomly; in other words, the initial population consist of \( \text{Popsize} \) chromosomes that are just random permutations of the set of numbers: \( \{1, 2, \ldots, n-1, n\} \). The evaluation function for GGA is the makespan of a schedule. The GGA selection scheme and seven genetic operators (crossover and mutation) are vital constituents.

In GGA, the selection scheme is the roulette wheel (Goldberg 1989) with elitist selection, which is outlined in Figure 3.4. This scheme guarantees that the best solution in a generation will never degrade over generations. Four crossover operators -LOX, PMX, C1, and NABEL- and three mutation operators -SWAP, INV, and INS- are applied (Wang and Zheng 2003). Every crossover operator is applied to two chromosomes (parents) and results in two new ones (children). Every mutation operator is applied to one chromosome and results in a different chromosome.

\[
\begin{array}{|l|}
\hline
\text{Roulette wheel and elitist selection} \\
\hline
\text{1} & \text{Evaluate the objective function for the generated initial population.} \\
\text{2} & \text{Compute chromosome fitness as maximum objective function value minus chromosome objective function value, for every chromosome} \\
\text{3} & \text{Sort chromosomes in an ascending order according to their fitness} \\
\text{4} & \text{Compute the chromosome probability of being selected as its fitness divided by total fitness of all chromosomes. Then compute the cumulative chromosome probability for every chromosome.} \\
\text{5} & \text{Generate a random number } r \text{ between zero and one; scan sorted chromosomes; if } r < \text{ chromosome cumulative probability, this chromosome is selected to temporary population, otherwise the next chromosome is checked and so on.} \\
\text{6} & \text{5 is repeated } \text{Popsize–}1 \text{ times and the chromosome with the highest fitness (elitist) is retained for the next generation.} \\
\hline
\end{array}
\]

\text{Figure 3.4 GGA roulette wheel and elitist selection}
LOX (Linear Order Crossover)

In LOX, two cutting points in the two parents (P1 and P2) are chosen randomly, and the entries between these two cutting points in one parent are removed from the other parent leaving some vacant entries. Those are then slid, one by one, towards the center until they cover the middle, originally cut, part. While sliding, no temporarily vacant entries (Vs) are allowed to exceed other Vs. Finally, the Vs are substituted by the ones that originally existed between the two cut points in the other parent to obtain the two children (C1 and C2) (Wang and Zheng 2003). Figure 3.5 illustrates LOX with an example, showing the two parents -P1 and P2-, the intermediate stages, and the resulting children -C1 and C2. In this instance, the 2, 1, and 8 in P1 are temporarily replaced with Vs. Then the Vs are slid, one by one, towards the center until they cover the middle section. Last, the Vs are replaced by 2, 1, and 8 to obtain C1. The same process is applied to the second parent, P2 to obtain C2.

$$
\begin{array}{cccc|cccc|cccc|cccc}
\text{P1:} & 2 & 6 & 4 & 7 & 3 & 5 & 8 & 1 \\
\text{V:} & 6 & 4 & 7 & 3 & 5 & \text{V:} & 5 & 2 & 1 & 8 \\
6 & 4 & \text{V:} & 5 & 2 & 1 & 8 & 7 & 3 & 5 \\
\text{C1:} & 6 & 4 & 2 & 1 & 8 & 7 & 3 & 5 \\
\text{P2:} & 4 & 5 & 2 & 1 & 8 & 7 & 6 & 3 \\
\text{V:} & 5 & 2 & 1 & 8 & \text{V:} & 6 & \text{V:} & 5 & 2 & 1 & 8 & 7 & 6 & 3 \\
\text{C2:} & 5 & 2 & 4 & 7 & 3 & 1 & 8 & 6 \\
\end{array}
$$

Figure 3.5  An example of applying LOX crossover (V = temporarily vacant position)

PMX (Partially Mapped Crossover)

Two cutting points in the two parents are chosen randomly and the entries between these two cutting points are exchanged to form the middle parts of the two children.
The rest of children entries are filled up by partial mapping –position wise exchanges (Goldberg 1989). Figure 3.6 shows an example of applying PMX. In this example, after exchanging the entries between the two cut sections, in $C_1$ the 8 and 4 are copied from $P_1$ as they are since they are not in the middle section. But the 1 cannot be copied because it is already in the middle section. Therefore the 5 is inserted in stead due to its similar position in $P_1$ to that of the 1 in $C_1$. The rest of the process proceeds similarly.

![Figure 3.6 An example of applying PMX crossover](image)

**CO1 (Crossover 1)**

This operator chooses one point in one parent randomly, copies the subsection of the parent before that point to the corresponding child, and then fills up the rest of the child chromosome by taking every legitimate point in the other parent in order (Wang and Zheng 2003). Figure 3.7 illustrates CO1 with an example. In this example, the first three entries in $C_1$, the 2, 6, and 4, are copied from $P_1$ as they are. To insert the fourth entry, $P_2$ is scanned from left to right, finding out that 4 cannot be copied since it already exists in $C_1$, as the third entry. So the 5 is chosen to be the fourth entry in $C_1$. Following that, the rest of $P_2$ is scanned again from left to right to realize that the 2 is not legitimate since it already exists in $C_1$, so the 1 is selected to be the fifth entry in $C_1$. The rest of the process proceeds similarly.
Figure 3.7 An example of applying CO1 crossover

NABEL

This operator is implemented by filling up one child by entries from one parent that are in positions with orders equal to corresponding entries in the other parent (Wang and Zheng 2003). Figure 3.8 shows an example of applying NABEL. In this example, the first entry in C1 is 5 because the first entry in P1 is 2 and the second entry in P2 is 5. Also, the second entry in C1 is 7 since the second entry in P1 is 6 and the sixth entry in P2 is 7. The rest of the process proceeds similarly.

Figure 3.8 An example of implementing NABEL crossover

SWAP

This mutation operator selects two distinct entries randomly and swaps them. Figure 3.9 shows an example of an original chromosome (OC) and a mutated chromosome (MC).
INV (Inverse)

This mutation operator inverts the entries between two random positions (Figure 3.10).

INS (Insert)

This mutation operator chooses two entries randomly and inserts the back one before the front one (Figure 3.11).

In GGA, the population size is selected to be 56: 10 chromosomes result from every one of the four crossover operators; 5 chromosomes are generated by every one of the three mutation operators; and the remaining chromosome is the best one in the preceding generation. The stopping criterion occurs when 54 generations are
reproduced or if the lower bound is reached since the latter condition means that the optimum is reached. This means that 3024 evaluations of the objective function are permitted, which is quite reasonable so that the execution time is not too long. The same number, 3024, of objective function evaluations is used for GSA for fair comparison.

The crossover rate is 1.0 and the mutation rate is 0.1. When applying a mutation, a random number between zero and one is generated; if that number is below the mutation rate, the chromosome picked undergoes mutation; otherwise, the chromosome is copied to the new population as is. When applying a crossover, the two chromosomes picked always undergo crossover since the rate is 1.0. The GGA parameters were selected based on the values recommended in literature and the results of pilot runs.

3.3 General Simulated Annealing

The problem representation for GSA is the same as the one for GGA, a permutation of numbers with a length equal to the number of parts in a digraph. The schedule is obtained in the same way as in GGA. That is a potential solution, \( X \), represents the sequence of parts to be machined in stage one of the AMS, coupled to the rule of scheduling assembly operations ASAP. The objective function is also the same, the makespan of the resulting AMS schedule.

GSA has a number of different characteristics from the simple SA reviewed in section 2.4.2. The GSA algorithm is outlined in Figure 3.12. The two fundamental, distinct features of this algorithm are the way of proposing a new solution and the criteria for deciding on accepting or rejecting that solution in case it is worse than the current solution. GSA computes 6 different neighbor permutations to the current
solution and selects the best, the one with the least corresponding makespan. This approach, called the best move strategy, provides the advantage of freeing the GSA performance from its possible dependence on the cooling schedule, the function $t_k$ in GSA (Ishibuchi et al. 1995). The six perturbation schemes (PSs) employed in GSA were published in Tian et al. (1999) as possible SA generation mechanisms, but they were experimented with independently, in different SA algorithms, unlike here. In other words, the best move strategy was not applied in that research. Figures 3.13 – 3.18 provide examples of PS1-6 with a brief description of every one.

**Figure 3.12  GSA algorithm**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Randomly choose starting solution $X_0$. Calculate $f_0 = f(X_0)$ (stopping criterion is the number of iterations). Specify $t_1$.</td>
</tr>
<tr>
<td>2</td>
<td>Obtain six neighbor permutations ($X_1$-$X_6$) from $X_0$ using Perturbation Schemes (PS1-PS6).</td>
</tr>
<tr>
<td>3</td>
<td>Compute objective function values for $X_1$-$X_6$. The least objective function value is $f_1$ and the solution associated with it is the proposed solution.</td>
</tr>
<tr>
<td>4</td>
<td>Calculate the current cooling schedule and the difference in the objective function value $t_k = \gamma^{iter} \times t_1$; $\Delta f = f_1 - f_0$</td>
</tr>
<tr>
<td>5</td>
<td>Determine the probability of accepting the proposed solution If $\Delta f \leq 0$, then $pr = 1$ Else $pr = e^{(-\Delta f / t_k)}$</td>
</tr>
<tr>
<td>6</td>
<td>A random number $r$ ($0 \leq r \leq 1$) is generated. If $r \leq pr$, then the step is accepted and the current solution is updated Else, no change is made to the current solution. Go to 2</td>
</tr>
</tbody>
</table>
OC: 1 3 2 4 5 7 6 8
MC: 1 3 2 5 4 7 6 8
   Two adjacent entries are swapped randomly

Figure 3.13 An example of PS1

OC: 1 3 2 4 5 7 6 8
MC: 1 7 2 4 5 3 6 8
   Two entries are swapped randomly

Figure 3.14 An example of PS2

OC: 1 3 2 4 5 7 6 8
MC: 1 6 3 2 4 5 7 8
   A single entry is moved randomly

Figure 3.15 An example of PS3

OC: 1 3 2 4 5 7 6 8
MC: 1 3 7 6 2 4 5 8
   A subsequence is moved randomly

Figure 3.16 An example of PS4

OC: 1 3 2 4 5 7 6 8
MC: 1 3 5 4 2 7 6 8
   A subsequence is reversed randomly

Figure 3.17 An example of PS5
The cooling schedule \( t_k \) in GSA is selected to be the same as the one published in Negenman (2001), as shown in Equation 3.1.

\[
t_k = \gamma^{\text{iter}} \times t_1
\]

This cooling schedule forces the probability of accepting a worse solution to decrease over time, which is desired since only fine tuning is required as the number of iterations increase, so that the gained proximity to the optimum is not lost. Following the recommended values in literature and utilizing the results of some pilot runs, the initial temperature \( t_1 \) and the parameter \( \gamma \) are selected to be 150 and 0.9, respectively.

The number of iterations is specified to be 504, which results in a maximum total of 3024 objective function evaluations, the same figure was specified for GGA, leading to a fair comparison.

### 3.4 Heuristic Assisted Genetic Algorithm

All of HAGA components are the same as those of GGA except the initial population.

In HAGA, the initial population, which consists of \( \text{Popsize} \) chromosomes, contains a chromosome resulting from HA. This implies that first HA is run to yield a solution; this solution is converted to a chromosome; and then this chromosome is included in...
the initial population. The rest of the initial population, \(Popsize – 1\) chromosomes, are generated randomly. These are random permutations of the set of numbers: 
\(\{1, 2,\ldots,n-1, n\}\), where \(n\) is the number of part nodes in a standardized digraph. Since the best chromosome in every generation is retained for the following generation, HAGA will always result in solutions that are never worse than the solution obtained by HA. Therefore HAGA plays the role of an additional GA that is expected to reach solutions that are better than those obtained by HA.

### 3.5 Heuristic Assisted Simulated Annealing

All of HASA components are the same as those of GSA except the initial solution. In HASA, the initial solution results from HA, instead of being generated randomly. HASA is meant to serve as an additional SA that is expected to reach solutions that are better than those obtained by HA.

### 3.6 Computational Experiment

The purpose of the computational experiment is to draw conclusions regarding the performances of the proposed and existing heuristics and how they are affected by several factors. The approach is to use a \(2^3\) factorial experiment, replicated twice, which is efficient in realizing the goal mentioned above (Hicks 1993). In factorial experiments, all of the possible combinations of factor levels are examined in each replication. The responses and factors are described, in details, next.

#### 3.6.1 Experimental Planning and design

This research addresses the problem of scheduling the AMS specified with the objective of minimizing the makespan. Two responses are chosen to evaluate the performance of every one of the five available algorithms: HA, GGA, GSA, HAGA,
and HASA. The two responses are the percentage deviation from the lower bound (DLB) and the frequency of resulting in the best solution (FBS). Ultimately, there are ten responses that are to be analyzed in order to make inferences about the performance of the algorithms available for solving the problem studied. DLB is calculated according to equation 3.2. FBS is the number of instances for which an algorithm obtains the best solution known.

\[
DLB = \left( \frac{\text{solution} - \text{lower bound}}{\text{lower bound}} \right) \times 100
\]  

(3.2)

The fixed effects model is chosen to describe the observations in the factorial experiment (Montgomery 2001). The factors to be varied are the number of part nodes in a digraph (A); the complexity of a digraph (B); and the ratio of average machining time per part to average subassembly time per subassembly operation (C). Two levels are selected for every factor, as shown in Table 3.2. The levels of factors A and C are quantitative, where the levels of factor B are qualitative. Digraph complexity (factor B) is either low (L) or high (H). The factors with their levels, especially for factors A and C, and the instance data generation are chosen in accordance with the information in Boothroyd (1992) and He and Babayan (2002), to intensify the practicality of this research. In the computational experiment, there are 8 runs per replicate. Table 3.2 lists the 8 combinations of factor levels in the $2^3$ factorial design used.

Figures 3.19-3.22 show the four digraphs experimented with: low complexity with 16 part nodes; high complexity with 16 part nodes; low complexity with 32 part nodes; and high complexity with 32 part nodes. From these four figures, it is noted that higher complexity is realized by three attributes: more subassembly nodes; more assembly levels; and loss of symmetry in the digraph.
The number of instances per combination for the experiment in this work is set equal to 100, yielding a total of 800 instances for the eight combinations. The number of instances per combination used in this research was statistically verified to be adequate for representing the performance of all heuristics studied. This was achieved by plotting the cumulative average makespans versus the number of replicates, and making sure that adding more replicates than 100 would not significantly affect the locations of the cumulative averages. For every combination, the 100 instances are divided into two replicates of 50 instances each before analyzing the results, as discussed in the next chapter. For the same combination, what differentiate an instance from another are the machining and assembly time sets, which are initially generated from discrete uniform distributions between (2, 25) and (1, 30), respectively. Following that initial step, the machining and assembly time sets are either scaled up or down to reach the desired level of factor C, the ratio of average machining time per part to average subassembly time per subassembly operation, 1/2 or 2. The 800 instances are solved five times using the five available algorithms: HA, GGA, GSA, HAGA, and HASA. Then the two responses are computed for every algorithm. The procedure is described next.

**TABLE 3.2 The $2^3$ factorial design of the computational experiment showing the two levels of every factor**

<table>
<thead>
<tr>
<th>Run #</th>
<th>Factor</th>
<th>Run label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16 L</td>
<td>1/2 (1)</td>
</tr>
<tr>
<td>2</td>
<td>32 L</td>
<td>a</td>
</tr>
<tr>
<td>3</td>
<td>16 H</td>
<td>1/2 b</td>
</tr>
<tr>
<td>4</td>
<td>32 H</td>
<td>1/2 ab</td>
</tr>
<tr>
<td>5</td>
<td>16 L</td>
<td>2 c</td>
</tr>
<tr>
<td>6</td>
<td>32 L</td>
<td>2 ac</td>
</tr>
<tr>
<td>7</td>
<td>16 H</td>
<td>2 bc</td>
</tr>
<tr>
<td>8</td>
<td>32 H</td>
<td>2 abc</td>
</tr>
</tbody>
</table>
Figure 3.19  The experimental low complexity digraph with 16 part nodes and two assembly levels
Figure 3.20 The experimental high complexity digraph with 16 part nodes and three assembly levels
Figure 3.21 The experimental low complexity digraph with 32 part nodes and two assembly levels
Figure 3.22 The experimental high complexity digraph with 32 part nodes and four assembly levels
3.6.2 Procedure

The five algorithms experimented with in the current research were coded in MATLAB® 6.1 (The Math Works, Inc. 2001), and the codes were run on an Intel® Pentium® II, 233 MHZ computer. The Design-Expert® 6 (Stat-Ease 2001) software was used to aid in analyzing the results. The procedure of experimentation is described in the following points:

[1] Generate the machining time sets and the assembly time sets for the 800 problems experimented with according to the factor combinations associated with every run.

[2] Run the codes to compute makespans resulting from HA, GGA, GSA, HAGA, and HASA, in addition to computing the lower bound.

[3] Compute DLB and FBS associated with all five algorithms for every instance.

[4] For every combination, divide the 100 instances into two replicates of 50 instances each and compute the averages of the results over the 50 instances.

[5] Enter the results of the above to Design-Expert® 6 (Stat-Ease 2001) to start the analysis (Montgomery 2001):

  o Form initial model
  o Perform statistical testing
  o Refine model if necessary
  o Analyze residuals
  o Interpret results

The next chapter presents and discusses the results of the computational experiment.
CHAPTER 4
RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents and explains the results of the experiment described in Chapter Three. In this research, there are five algorithms for solving the problem studied and two criteria (responses) for testing the efficiency of these algorithms. The five algorithms are Heuristic Algorithm (HA), General Genetic Algorithm (GGA), General Simulated Annealing (GSA), Heuristic Assisted Genetic Algorithm (HAGA), and Heuristic Assisted Simulated Annealing (HASA). The two criteria are the percentage deviation from the lower bound (DLB) and the frequency of resulting in the best solution (FBS). It is noted that the four heuristics presented by He and Babayan (2002) are counted as only one algorithm, calling it ‘Heuristic Algorithm,’ and this is achieved by running the three heuristics and selecting the best solution obtained by them to be the HA solution.

As previously stated, three factors are selected to investigate their effects on the responses mentioned above, and these factors are the number of part nodes in the digraph (factor A), the digraph complexity (factor B), and the ratio of average machining time per part to average subassembly time per subassembly operation (factor C). The rest of this chapter describes the verification procedure, provides a summary of experimental results, and then presents the detailed results for the ten experimental responses.
4.2 Verification

Experimentation started after verifying that the developed codes are yielding correct results. Methods of verification used with HA codes include comparing the code results to those of problems with known solutions and comparing intermediate results of the codes to hand calculations. With GGA, GSA, HAGA, and HASA all associated codes utilized the same objective function subroutine as the one for HA. Consequently, this subroutine, which converts a chromosome or solution to a schedule with a known makespan was already verified to be correct when the HA codes were verified.

Other subroutines for those four algorithms were verified to be correct in various ways. For instance, the subroutines that implement crossovers, mutations, and perturbation schemes were all verified to be correct by comparing their outputs to the results of hand calculations of sample chromosomes or solutions. In addition, if anyone of them yielded an illegal chromosome or solution, that would be revealed because the objective function subroutine would experience an error developing a schedule and computing its makespan for an illegal problem representation, which never happened during experimentation. For HAGA and HASA, it was verified that all makespans they obtained were not worse than the corresponding solutions obtained by HA.

4.3 Summary of Results

Table 4.1 lists the average DLB and the FBS for the ten responses in the eight experimental runs. The entries in bold are the minima (best) for DLB or the maxima (best) for FBS, with their cells shaded, in a certain run. If a solution is equal to the lower bound (DLB = 0), then this solution is optimal. It is noticed that all of the
proposed algorithms – GGA, GSA, HAGA, and HASA – outperform HA when $C = 1/2$. For example, in Run 3 DLB falls from 6.01% obtained by HA to 1.33% obtained by HASA, and FBS increases from 4 obtained by HA to 92 obtained by HASA.

**TABLE 4.1** Experimental results
(a) Results in averages and frequencies for 100 instances per run

<table>
<thead>
<tr>
<th>Run #</th>
<th>FACTORS Label</th>
<th>RESPONSES dHA</th>
<th>fHA</th>
<th>dGGA</th>
<th>fGGA</th>
<th>dGSA</th>
<th>fGSA</th>
<th>dHAGA</th>
<th>fHAGA</th>
<th>dHASA</th>
<th>fHASA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16 L 1/2 (1)</td>
<td>6.11</td>
<td>67</td>
<td>1.72</td>
<td>1.66</td>
<td>1.72</td>
<td>1.67</td>
<td>12</td>
<td>87</td>
<td>93</td>
<td>86</td>
</tr>
<tr>
<td>2</td>
<td>32 L 1/2 a</td>
<td>2.63</td>
<td>74</td>
<td>0.73</td>
<td>0.36</td>
<td>0.44</td>
<td>0.31</td>
<td>7</td>
<td>45</td>
<td>87</td>
<td>74</td>
</tr>
<tr>
<td>3</td>
<td>16 H 1/2 b</td>
<td>6.01</td>
<td>97</td>
<td>1.40</td>
<td>1.35</td>
<td>1.43</td>
<td>1.33</td>
<td>4</td>
<td>80</td>
<td>87</td>
<td>78</td>
</tr>
<tr>
<td>4</td>
<td>32 H 1/2 ab</td>
<td>3.76</td>
<td>86</td>
<td>0.85</td>
<td>0.38</td>
<td>0.83</td>
<td>0.45</td>
<td>0</td>
<td>42</td>
<td>89</td>
<td>47</td>
</tr>
<tr>
<td>5</td>
<td>16 L 2 c</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>32 L 2 ac</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>16 H 2 bc</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>32 H 2 abc</td>
<td>0</td>
<td>99</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>99</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE 4.1** (b) FACTORS Key

- **A** Number of digraph part nodes
- **B** Digraph complexity
- **C** Average machining time to average subassembly time ratio

**TABLE 4.1** (c) RESPONSES Key

- **dHA** Heuristic Algorithm average % deviation from lower bound
- **dGGA** General Genetic Algorithm average % deviation from lower bound
- **dGSA** General Simulated Annealing average % deviation from lower bound
- **dHAGA** Heuristic Assisted Genetic Algorithm average % deviation from lower bound
- **dHASA** Heuristic Assisted Simulated Annealing average % deviation from lower bound
- **fHA** Heuristic Algorithm frequency of resulting in best solution
- **fGGA** General Genetic Algorithm frequency of resulting in best solution
- **fGSA** General Simulated Annealing frequency of resulting in best solution
- **fHAGA** Heuristic Assisted Genetic Algorithm frequency of resulting in best solution
- **fHASA** Heuristic Assisted Simulated Annealing frequency of resulting in best solution
Furthermore, all available algorithms almost exhibit the same performance when \( C = 2 \), resulting in optimal solutions. This may be attributed to the fact that when \( C = 2 \), there is much more load on the single machine (stage one) than that on the assembly stations (stage two), leading to the reduction of the scheduling problem to that of a manufacturing system with a single machine in stage one and a single assembly station in stage two, in most cases. In other words, as the load on stage two is relatively very low, there is no need to have more than one assembly station in that stage. A similar scheduling problem of a manufacturing system with a single machine and a single assembly station was determined to be solvable with optimal solutions by Kusiak (1989), so this problem is easy to solve.

The experimental results of this research are summarized in the next three sections. First, the numerical results of DLB are presented for all eight experimental runs. Second, the numerical results of FBS are provided. Third, the factors that significantly affect the performance of the algorithms experimented with are highlighted.

### 4.4 Percentage Deviation from the Lower Bound

Tables 4.2 and 4.3 in addition to Figure 4.1 provide descriptive statistics of DLB for all five algorithms over the 800 instances solved in the computational experiment. In Table 4.2, which lists the minima, maxima, means, and standard deviations, the lowest values among the five heuristics are shown in bold. It can be deduced that HA is outperformed by the other four heuristics. Although all five algorithms result in optimal solutions for at least one instance, as a consequence of the fact that all have minimum values of zero, the maximum values decrease from above 20 % by HA to below 10 % by the other four heuristics. The averages show that GSA and HASA do
better than the rest. On the whole, GSA seems to be the best heuristic since it has the minimum values among all four statistics.

**TABLE 4.2 Statistics for percentage deviation from the lower bound**

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>0</td>
<td>21.21</td>
<td>2.31</td>
<td>3.54</td>
</tr>
<tr>
<td>GGA</td>
<td>0</td>
<td>9.26</td>
<td>0.59</td>
<td>1.19</td>
</tr>
<tr>
<td>GSA</td>
<td>0</td>
<td>9.26</td>
<td>0.47</td>
<td>1.11</td>
</tr>
<tr>
<td>HAGA</td>
<td>0</td>
<td>9.26</td>
<td>0.55</td>
<td>1.19</td>
</tr>
<tr>
<td>HASA</td>
<td>0</td>
<td>9.26</td>
<td>0.47</td>
<td>1.13</td>
</tr>
</tbody>
</table>

Figure 4.1 shows box plots of average DLB over 100 replicates for the eight experimental runs. It is noticed that the four proposed heuristics outperform HA and that GSA and HASA have the best results. Better performance is illustrated by shorter box length (lower inter-quartile range) and lower median (horizontal line inside the box). Heuristic-assisted algorithms have lower medians than those of the general ones. The reasons for these results include the nature of the problem, the structure of heuristics, and the choice of factor levels. This implies that those results are only valid for the conditions studied in this research, which is elaborated on in Section 5.2: Validation.

Table 4.3 presents the number of instances that belong to one of four ranges. For GGA, GSA, HAGA, and HASA the trend is a diminishing number of instances as DLB grows, until it reaches zero in R₄. However, for HA, as DLB associated with R₂ increases to that associated with R₃, the number of instances boosts.
69

Figure 4.1 Box plots of average DLB over 100 replicates for the eight experimental runs.

<table>
<thead>
<tr>
<th>Number of instances</th>
<th>Number of instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic</td>
<td>R₁ R₂ R₃ R₄</td>
</tr>
<tr>
<td>HA</td>
<td>452 111 205 32</td>
</tr>
<tr>
<td>GGA</td>
<td>645 115 40 0</td>
</tr>
<tr>
<td>GSA</td>
<td>684 82 34 0</td>
</tr>
<tr>
<td>HAGA</td>
<td>659 100 41 0</td>
</tr>
<tr>
<td>HASA</td>
<td>686 79 35 0</td>
</tr>
</tbody>
</table>

4.5 Frequency of Resulting in the Best Solution

FBS is the number of instances for which an algorithm obtains the best solution among the solutions obtained by the five heuristics studied in this research. Figure 4.2 shows FBS for the five algorithms studied over the 800 instances solved. It is noted that HA is outperformed by the four proposed heuristics and that HASA and GSA
surpass the rest. The same data along with the percentage frequency of resulting in
the best solution are shown in Table 4.4 for further depiction.

![Figure 4.2 FBS for the five heuristics](image)

**TABLE 4.4 FBS and %FBS for the five heuristics**

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Frequency of resulting in best solution</th>
<th>% frequency of resulting in best solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>423</td>
<td>52.9</td>
</tr>
<tr>
<td>GGA</td>
<td>653</td>
<td>81.6</td>
</tr>
<tr>
<td>GSA</td>
<td>756</td>
<td>94.5</td>
</tr>
<tr>
<td>HAGA</td>
<td>685</td>
<td>85.7</td>
</tr>
<tr>
<td>HASA</td>
<td>764</td>
<td>95.5</td>
</tr>
</tbody>
</table>

**4.6 Significant Factors and Interactions**

The significant factors are the factors that significantly affect the performance of the
heuristics, their abilities to arrive at optimal, near optimal, or best solutions. When a
change in one factor causes a different change in the response at one level of another
factor, there is an interaction between the two factors (Hicks 1993). In order to determine the significant factors and interactions, the 100 instances per run were divided by two to yield two 50-instance replicates per run, as shown in Table 4.5. Dividing the results into two replicates ensures that the mean square error, calculated in the analysis of variance (ANOVA), is not too low, as the mean square error is inversely proportional to the number of replicates. This is desirable so that the significant factors and interactions are pointed out.

**TABLE 4.5 Experimental results in two replicates**

<table>
<thead>
<tr>
<th>Run #</th>
<th>FACTORS</th>
<th>RESPONSES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A  B  C</td>
<td>dHA  dGGA dGSA dHAGA dHASA fHA fGGA fGSA fHAGA fHASA</td>
</tr>
<tr>
<td>1</td>
<td>16 L 1/2</td>
<td>6.16  1.70 1.66 1.71  1.69  4  43  46  42  44</td>
</tr>
<tr>
<td>2</td>
<td>16 L 1/2</td>
<td>6.05  1.73 1.66 1.72  1.65  8  44  47  44  48</td>
</tr>
<tr>
<td>3</td>
<td>32 L 1/2</td>
<td>2.90  0.64 0.38 0.43  0.32  4  27  44  38  47</td>
</tr>
<tr>
<td>4</td>
<td>32 L 1/2</td>
<td>2.36  0.83 0.35 0.44  0.30  3  18  43  36  47</td>
</tr>
<tr>
<td>5</td>
<td>16 H 1/2</td>
<td>6.08  1.29 1.22 1.29  1.16  1  40  44  40  49</td>
</tr>
<tr>
<td>6</td>
<td>16 H 1/2</td>
<td>5.94  1.50 1.48 1.58  1.50  3  40  43  38  43</td>
</tr>
<tr>
<td>7</td>
<td>32 H 1/2</td>
<td>3.66  0.70 0.32 0.74  0.43  0  24  46  28  45</td>
</tr>
<tr>
<td>8</td>
<td>32 H 1/2</td>
<td>3.86  1.00 0.44 0.92  0.46  0  18  43  19  41</td>
</tr>
<tr>
<td>9</td>
<td>16 L 2</td>
<td>0  0  0  0  0  0  50  50  50  50</td>
</tr>
<tr>
<td>10</td>
<td>16 L 2</td>
<td>0  0  0  0  0  0  50  50  50  50</td>
</tr>
<tr>
<td>11</td>
<td>32 L 2</td>
<td>0  0  0  0  0  0  50  50  50  50</td>
</tr>
<tr>
<td>12</td>
<td>32 L 2</td>
<td>0  0  0  0  0  0  50  50  50  50</td>
</tr>
<tr>
<td>13</td>
<td>16 H 2</td>
<td>0  0  0  0  0  0  50  50  50  50</td>
</tr>
<tr>
<td>14</td>
<td>16 H 2</td>
<td>0  0  0  0  0  0  50  50  50  50</td>
</tr>
<tr>
<td>15</td>
<td>32 H 2</td>
<td>0  0  0  0  0  0  50  50  50  50</td>
</tr>
<tr>
<td>16</td>
<td>32 H 2</td>
<td>0  0  0  0  0  0  50  49  50  50  50</td>
</tr>
</tbody>
</table>

Table 4.6 shows the significant factors and interactions associated with the ten responses in the experiment. It is noted that factor C is significant for all of the ten responses of the computational experiment and that factor A and interaction A-C are significant for most of them. This may be related to the nature of the problem; the choice of factor ranges; and the different heuristic structures. Further, the nature of the problem is a complex one involving the product structure, manufacturing system structure, and machining and assembly time set distributions.
### TABLE 4.6  Significant factors and interactions

**a) Significant factors and interactions**

<table>
<thead>
<tr>
<th>Factor / Interaction</th>
<th>RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dHA</td>
</tr>
<tr>
<td>A</td>
<td>√</td>
</tr>
<tr>
<td>B</td>
<td>√</td>
</tr>
<tr>
<td>C</td>
<td>√</td>
</tr>
<tr>
<td>AB</td>
<td>√</td>
</tr>
<tr>
<td>AC</td>
<td>√</td>
</tr>
<tr>
<td>BC</td>
<td>√</td>
</tr>
<tr>
<td>ABC</td>
<td>√</td>
</tr>
</tbody>
</table>

**b) Symbol key**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>SIGNIFICANT FACTOR / INTERACTION</td>
</tr>
<tr>
<td>*</td>
<td>FACTOR MASKED BY INTERACTION</td>
</tr>
</tbody>
</table>

**c) Factor Key**

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Number of digraph part nodes</td>
</tr>
<tr>
<td>B</td>
<td>Digraph complexity</td>
</tr>
<tr>
<td>C</td>
<td>Average machining time to average subassembly time ratio</td>
</tr>
</tbody>
</table>

**d) Response Key**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dHA</td>
<td>Heuristic Algorithm average % deviation from lower bound</td>
</tr>
<tr>
<td>dGGA</td>
<td>General Genetic Algorithm average % deviation from lower bound</td>
</tr>
<tr>
<td>dGSA</td>
<td>General Simulated Annealing average % deviation from lower bound</td>
</tr>
<tr>
<td>dHAGA</td>
<td>Heuristic Assisted Genetic Algorithm average % deviation from lower bound</td>
</tr>
<tr>
<td>dHASA</td>
<td>Heuristic Assisted Simulated Annealing average % deviation from lower bound</td>
</tr>
<tr>
<td>fHA</td>
<td>Heuristic Algorithm frequency of resulting in best solution</td>
</tr>
<tr>
<td>fGGA</td>
<td>General Genetic Algorithm frequency of resulting in best solution</td>
</tr>
<tr>
<td>fGSA</td>
<td>General Simulated Annealing frequency of resulting in best solution</td>
</tr>
<tr>
<td>fHAGA</td>
<td>Heuristic Assisted Genetic Algorithm frequency of resulting in best solution</td>
</tr>
<tr>
<td>fHASA</td>
<td>Heuristic Assisted Simulated Annealing frequency of resulting in best solution</td>
</tr>
</tbody>
</table>
Moreover, it is noticed from Table 4.6 that \( f_{GSA} \) and \( f_{HASA} \) are only affected by factor C and not affected by factors A or B, or any interaction. This implies that GSA and HASA, which are SA-based heuristics, are more robust than the other heuristics, including the GA-based ones. Adding to this that \( f_{GSA} \) and \( f_{HASA} \) are highest (Table 4.1), it follows that GSA and HASA are expected to perform best, i.e. obtain schedules with least makespans, in the most difficult instances of the problem under investigation. Design-Expert® 6 (Stat-Ease 2001) was used to generate the required statistics for analyzing the results, such as the Analysis of Variance (ANOVA).

For the purpose of demonstration, what presented next is the detailed analysis of the results for the response GGA frequency of resulting in the best solution (\( f_{GGA} \)); the results of the remaining nine responses are presented in Appendices B through J. The significance level for all analyses is 5%.

### 4.7 \( f_{GGA} \) Results

Figure 4.3 shows the half normal plot of effects for GGA frequency of resulting in the best solution (\( f_{GGA} \)) where it can be seen that factors C (average machining time to subassembly time ratio) and A (number of part nodes) along with interaction AC are the significant terms. Those three terms were picked to be the model terms since they are far from the rest of points that almost make a straight line. Selecting model terms is done from right to left for the points that significantly deviate from zero as described in Design-Expert® 6 User’s Guide (2003) and Montgomery (2001). Table 4.7 is the ANOVA table for \( f_{GGA} \) as generated by Design-Expert® 6 (Stat-Ease 2001).
Table 4.7 fGGA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2114.19</td>
<td>3</td>
<td>704.73</td>
<td>113.9</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>A</td>
<td>410.063</td>
<td>1</td>
<td>410.06</td>
<td>66.273</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>1314.06</td>
<td>1</td>
<td>1314.1</td>
<td>212.37</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>AC</td>
<td>390.063</td>
<td>1</td>
<td>390.06</td>
<td>63.04</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>Residual</td>
<td>74.25</td>
<td>12</td>
<td>6.1875</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>14.75</td>
<td>4</td>
<td>3.6875</td>
<td>0.4958</td>
<td>0.7400</td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>59.5</td>
<td>8</td>
<td>7.4375</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>2188.44</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

The Model F-value of 113.90 implies the model is significant. There is only a 0.01% chance that a Model F-Value this large could occur due to noise. Values of "Prob>F" less than 0.0500 indicate model terms are significant. In this case A, C, AC are significant model terms. In other words, the frequency of GGA obtaining the best
solution is affected by factors A and C, in addition to their interaction. The interaction means that the way A affects C depends on the level of C and vice versa.

Figure 4.4 shows the normal plot of residuals for fGGA. The normal probability plot indicates whether the residuals follow a normal distribution. In this case, the plot looks acceptable because there are no trends observed in the data and no points detected to significantly depart from the rest of points. Figure 4.5 shows the plot of residuals versus predicted values. Although the variance of residuals is not constant, this should not affect the results. This is due to the fact that if the assumption of homogeneity of variance is violated, the F test is only slightly affected in the balanced (equal sample sizes in all combinations) fixed effects model (Montgomery 2001). This slight effect should not alter the ANOVA results here because the three terms have ‘Prob > F’ values that are below 0.01%; in other words, the three terms are significant even at a below-one-percent significance level.

Figure 4.4 Normal plot of residuals for fGGA
Figure 4.5 shows the outlier t plot, which allows the visualization of any outliers – unusual response data points. Here, no outliers are found since all points are within 3.5 standard deviations from zero. Figures 4.7 and 4.8 show the interaction graphs for the interaction A-C, as factors A and C were found out to be significant. It is noted that as factor C increases from 1/2 to 2, fGGA increases for all combinations of other factors, A and B. The performance of GGA, as measured by fGGA, is better at A =16 than at A =32. This is expected since as the number of part nodes (A) increases, the size of search space (A!) expands. Moreover, the power of factor C is detected to be very high because no matter what A equals, when C = 2, fGGA is at its top value.
Figure 4.6 Plot of Outlier T for fGGA

Figure 4.7 A - C interaction graph at B = low for transformed fGGA
4.8 Computational Time

In this research, the codes were run on an Intel® Pentium® III, 350 MHZ computer. The computational time spent to solve an instance by GGA, GSA, HAGA, or HASA depends on the computer speed and the difficulty of that specific instance, but the average, of solving an instance by a single heuristic, in this research was 1.7 minutes. On other hand, the computational time spent to solve an instance by HA was recorded to be below two seconds. This implies that the proposed heuristics provide better solutions at the expense of extra computational effort.
CHAPTER 5
CONCLUSION

5.1 Summary and Conclusion

In this research, the problem of scheduling an AMS, which consists of machining and assembly stages, with the objective of minimizing the makespan was studied. This AMS fulfills the goals of agile manufacturing, the latest manufacturing paradigm, by allowing the manufacturing of modular products in high varieties, in small batches, and at low costs. He and Babayan (2002) developed heuristics to solve the problem under investigation in the current research in addition to presenting a lower bound. However, the heuristics sometimes resulted in makespans with high deviations from the corresponding lower bounds. Consequently, GAs and SA were thought of as powerful methods that can produce better results.

Four GAs and SA-based heuristics were developed to solve the scheduling problem, and a computational experiment was designed to compare the performance of all available heuristics and point out the significant factors and interactions. By the means of the computational experiment, the proposed heuristics were shown to perform better than the existing heuristics, but at the expense of extra computational time. The results of the computational experiment also revealed that the most significant factor affecting the performance of all available heuristics is the ratio of average machining time per part to average subassembly time per subassembly in a digraph (C). The higher this ratio is, the better the chance that an optimum schedule is found. This may be attributed to the fact that the scheduling problem reduces to that of a system with a single machine and a single assembly station, which is an easy
problem to solve (Kusiak 1989). This is true since when $C = 2$, there is generally no need for more than one assembly station in the second stage. Other significant factors and interactions were identified for the percentage deviation from the lower bound and the frequency of resulting in the best solution associated with the five heuristics experimented with. The reasons of the significance of those factors are related to the nature of the problem; the choice of factor ranges; and the different heuristic structures. The nature of the problem is a complex one involving the product structure, manufacturing system structure, and machining and assembly time set distributions. Section 5.3: Future Research further builds on this fact to identify how the outcomes of the current research can be utilized practically.

It was noticed that for $f_{GSA}$ and $f_{HASA}$, the two SA-based heuristics, the only significant factor is the ratio of average machining time per part to average subassembly time per subassembly in a digraph. In other words, $f_{GSA}$ and $f_{HASA}$ are not affected by the rest of factors and interactions. This implies that GSA and HASA are robust and that they are expected to perform best in the most difficult problem instances.

Some practical suggestions about how to improve the likelihood of finding either the optimum or the best solution with least effort may be derived from the results of this research. Before starting to solve any problem, the lower bound and the ratio of average machining time per part to average subassembly time per subassembly in a digraph are computed. If the ratio mentioned above is high (far above one), it is most probable that the optimum solution is easy to find and that it equals the lower bound. HA is chosen since it does not need long computation time. However, if that ratio is low, it is most likely that the optimum is difficult to find and
the HA solution is not optimal. This implies that GGA, GSA, HAGA, and or HASA are to be run.

It is worth mentioning that the developed heuristics in this research are capable of handling similar problem types and not only the one studied here. This results from the fact that the developed SA and GA-based heuristics deal with the problem representation and not the problem itself. Therefore similar problems that can have the same problem representation utilized here can be solved by the four developed heuristics in the current research.

5.2 Validation

The conclusions of this research, based on the results of the computational experiment, were validated when validating the results. The results were validated by statistical means, such as normal probability plots and outlier t plots, in addition to statistical arguments from Montgomery (2001).

All the results of this research work are perceived to be valid only for the AMS with the problem description and assumptions given in Chapter One. It should be noted that altering any one of these assumptions would change the problem studied here and thus expel it from the domain of the results. Moreover, as the scheduling problem parameters, such as the number of parts, fall outside of the experimental conditions specified in this thesis, the results of the computational experiment are not necessarily valid with high confidence.

5.3 Future Research

The outcomes of the current research provide insights on how to schedule similar manufacturing systems that are either agile or even more traditional. Similar
manufacturing systems to the one studied here include plasma cutting, NC pressing, patron cutting in apparel industry, and steel rolling, cleaning, or pickling. More research is needed to accurately model the system chosen, after defining the problem with the correct assumptions. For instance, it could be necessary to change the system structure allowing the feeding of supporting parts, at the assembly stations in stage two, which were not machined by the single machine in stage one. Also, including significant handling times between assembly stations could be essential in a specific problem context. In that case, the number of assembly stations can be included as an experimental factor.

Other numerous points are possible for extending this research. One example is to study an AMS with the structure of a flowshop in the first stage and multiple assembly stations in the second stage. However, more research is needed to match this AMS structure to real-life applications. Also, it is needed to study the similarities and differences between this new AMS structure and the one studied in the current research, so that the results here can be fully exploited while solving the new scheduling problem. Another instance is to study the same AMS structure as in this thesis but with sequence-dependent set up times.
REFERENCES


@www.me.uic.edu/research/Labs/he/PDFs/ija_paper_revised_final_final.pdf


Kim, J-U; Kim, Y-D. 1996. “Simulated annealing and genetic algorithms for scheduling products with multi-level product structure.” *Computers and


APPENDIX A

ALGORITHM BY KUSIAK

1. Label all nodes of the digraph G representing the structure of the product considered. If it is a simple digraph, then use the maximum level of depth first (MLDF) rule to generate optimal schedule, stop; otherwise go to 2.

2. Remove the root node in G and decompose it into sub-digraphs $g_l$, $l = 1, \ldots, L$. If all $g_l$s are simple digraphs, set $k=0$, and go to 3; otherwise, decompose each $g_l$ which is not a simple digraph into simple digraphs by removing its root node. Let $v_i$ denote a root node which has been removed, $j = 1, \ldots, J$ (note that the removed nodes should be numbered in increasing order starting from the root node of G). Set $k = J$ and go to 3.

3. Let $g_k$ denote the simple digraph associated with $v_k$. Use the MLDF rule to generate the minimum makespan partial schedule $S^*(g_k)$ for each sub-digraph $g_k$, $i = 1, \ldots, N_k$, where $N_k$ is the number of sub-digraphs obtained after $v_k$ has been removed.

4. For each partial schedule $S^*(g_k)$ obtained in 3 determine
   i) in-process idle time $I_{ik}$
   ii) terminal time $T_{ik}$, $i = 1, \ldots, N_k$

5. Separate the $S^*(g_k)$ into two lists:
   list 1: schedules $S^*(g_k)$ such that $I_{ik} \leq T_{ik}$
   list 2: schedules $S^*(g_k)$ such that $I_{ik} > T_{ik}$

6. Use the longest idle time last (LITL) rule to generate
   $S_1(g_k) = [S^*(g_{[i]k}), S^*(g_{[j]k}), \ldots, S^*(g_{[r]k})]$, for $S^*(g_k)$ in list 1, $i = 1, \ldots, r$
   and use the longest terminal time first (LTTF) rule to generate
   $S_2(g_k) = [S^*(g_{[r+1]k}), S^*(g_{[r+2]k}), \ldots, S^*(g_{[t]k})]$, for $S^*(g_k)$ in list 2, $i = r+1, \ldots, t$, $t = N_k$
   Then generate the partial schedule
   $S^*(g_k) = [S_1(g_k), S_2(g_k), v_k]$.

7. If $v_k$ is the root node, then $S^*(g_k)$ is the optimal schedule, stop; otherwise, go to 8.

8. Consider $S^*(g_k)$ as a simple sub-digraph schedule and calculate $I_k$ and $T_k$. Set $k = k-1$, go to 3.

---

1 Source: Kusiak (1989)
APPENDIX B

dHA RESULTS

For HA percentage deviation from the lower bound (dHA), it was necessary to transform the data using the power transformation according to equation B.1.

\[ dHA' = (dHA + 1)^3 \]  \hspace{1cm} (B.1)

Figure B.1 shows the half normal plot for the transformed dHA where it can be seen that factors C (Average machining time to assembly time ratio) and A (number of part nodes) along with interaction AC are most significant. Table 4.6 is the ANOVA table as generated by Design Expert 6.0.10.

![Half Normal Plot](image)

**Figure B.1** Half normal plot of effects for transformed dHA
### TABLE B.1 dHA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob. &gt; F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>336275</td>
<td>7</td>
<td>48039</td>
<td>563.68</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>A</td>
<td>74710.3</td>
<td>1</td>
<td>74710</td>
<td>876.63</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>B</td>
<td>504.036</td>
<td>1</td>
<td>504.04</td>
<td>5.9142</td>
<td>0.0411</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>183120</td>
<td>1</td>
<td>183120</td>
<td>2148.7</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>AB</td>
<td>1363.35</td>
<td>1</td>
<td>1363.3</td>
<td>15.997</td>
<td>0.0040</td>
<td>*</td>
</tr>
<tr>
<td>AC</td>
<td>74710.3</td>
<td>1</td>
<td>74710</td>
<td>876.63</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>BC</td>
<td>504.036</td>
<td>1</td>
<td>504.04</td>
<td>5.9142</td>
<td>0.0411</td>
<td>*</td>
</tr>
<tr>
<td>ABC</td>
<td>1363.35</td>
<td>1</td>
<td>1363.3</td>
<td>15.997</td>
<td>0.0040</td>
<td>*</td>
</tr>
<tr>
<td>Pure Error</td>
<td>681.794</td>
<td>8</td>
<td>85.224</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>336957</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

The Model F-value of 563.68 implies the model is significant. There is only a 0.01% chance that a Model F-Value this large could occur due to noise. Values of "Prob > F" less than 0.0500 indicate model terms are significant. In this case A, B, C, AB, AC, BC, ABC are significant model terms.

Figure B.2 shows the normal plot of residuals, which indicates whether the residuals follow a normal distribution. In this case, although the plot does not resemble a sound straight line, this should be of no serious concern. The F test is only slightly affected by departures from normality, which cause the true significance level to differ slightly from the advertised values, for the fixed effects model (Montgomery 2001). Figure B.3 shows the plot of residuals versus predicted, which looks satisfactory since the residuals appear to have a nearly constant variance. Figure B.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response dat points. Here, no outliers are found since all points are within 3.5 standard deviations from zero. Figures B.5 and B.6 show the interaction graphs for the two-factor interaction A-C, which was found out to be most significant along with factors C and A.
Figure B.2 Normal plot of residuals for transformed dHA

Figure B.3 Plot of residuals vs. predicted for transformed dHA
Figure B.4 Plot of Outlier T for transformed dHA

Figure B.5 A - C interaction graph at B = low for transformed dHA
Figure B.6  A-C interaction graph at B = high for transformed dHA
APPENDIX C

dGGA RESULTS

For GGA percentage deviation from the lower bound (dGGA), the data did not go through any transformations. Figure C.1 shows the half normal plot for dGGA where it can be noticed that factors C and A along with interaction AC are most significant. Those three terms were picked to be the model terms since they are far from the rest of points that almost make a straight line. Table C.1 is the ANOVA table for HA percentage deviation from the lower bound as generated by Design Expert 6.0.10.

Figure C.1  Half normal plot of effects for dGGA
TABLE C.1  dGGA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>6.67136</td>
<td>3</td>
<td>2.2238</td>
<td>134.22</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>A</td>
<td>0.58439</td>
<td>1</td>
<td>0.5844</td>
<td>35.273</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>5.4931</td>
<td>1</td>
<td>5.4931</td>
<td>331.55</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>AC</td>
<td>0.59387</td>
<td>1</td>
<td>0.5939</td>
<td>35.845</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>Residual</td>
<td>0.19881</td>
<td>12</td>
<td>0.0166</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>0.11358</td>
<td>4</td>
<td>0.0284</td>
<td>2.6652</td>
<td>0.1110</td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>0.08523</td>
<td>8</td>
<td>0.0107</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>6.87017</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

The Model F-value of 134.22 implies the model is significant. There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise. Values of "Prob > F" less than 0.0500 indicate model terms are significant. In this case A, C, and AC are significant model terms.

Figure C.2 shows the normal plot of residuals for the GGA percentage deviation from the lower bound. Figure C.3 shows the plot of residuals versus predicted values. Figure C.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response data points. Here, no outliers are found since all points are within 3.5 standard deviations from zero. Figures C.5 and C.6 show the interaction graphs for the two factor interaction A-C, which was found out to be significant along with factors C and A.
Figure C.2 Normal plot of residuals for dGGA

Figure C.3 Plot of residuals vs. predicted values for dGGA
Figure C.4  Plot of Outlier T for dGGA

Figure C.5  A - C interaction graph at B = low for dGGA
Figure C.6 A-C interaction graph at B = high for dGGA
APPENDIX D

dGSA RESULTS

For GSA percentage deviation from the lower bound (dGSA), the data were transformed using the power transformation method and the constants were chosen according to equation D.1.

\[ dGSA' = (dGSA + 1)^2 \]  \hspace{1cm} (D.1)

Figure D.1 shows the half normal plot for dGSA where it can be seen that factors C) and A along with interaction AC are most significant. Those three terms were picked to be the model terms since they are far from the rest of points that almost make a straight line. Table 4.7 is the ANOVA table for dGSA.

Figure D.1  Half normal plot of effects for dGSA
### Table D.1 dGSA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>1.97742</td>
<td>3</td>
<td>0.6591</td>
<td>1065.6</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>A</td>
<td>0.13814</td>
<td>1</td>
<td>0.1381</td>
<td>223.32</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>1.70114</td>
<td>1</td>
<td>1.7011</td>
<td>2750.1</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>AC</td>
<td>0.13814</td>
<td>1</td>
<td>0.1381</td>
<td>223.32</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>Residual</td>
<td>0.00742</td>
<td>12</td>
<td>0.0006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>0.00187</td>
<td>4</td>
<td>0.0005</td>
<td>0.6758</td>
<td>0.6274</td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>0.00555</td>
<td>8</td>
<td>0.0007</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>1.98485</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

The Model F-value of 1065.57 implies the model is significant. There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise. Values of "Prob > F" less than 0.0500 indicate model terms are significant. In this case A, C, AC are significant model terms.

Figure D.2 shows the normal plot of residuals for dGSA. Figure D.3 shows the plot of residuals versus predicted values. Figure D.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response data points. Here, no outliers are found since all points are within 3.5 standard deviations from zero. Figures D.5 and D.6 show the interaction graphs for the two factor interaction A-C, which was found out to be significant along with factors C and A.
Figure D.2 Normal plot of residuals for dGSA

Figure D.3 Plot of residuals vs. predicted values for dGSA
Figure D.4 Plot of Outlier T for transformed dGSA

Figure D.5 A - C interaction graph at B = low for transformed dGSA
Figure D.6 A - C interaction graph at B = high for transformed dGSA
APPENDIX E

dHAGA RESULTS

For HAGA percentage deviation from the lower bound (dHAGA), the data were transformed using the natural log transformation method and the constants were chosen according to equation E.1.

\[ dHAGA' = \ln(dHAGA + 1) \quad (E.1) \]

Figure E.1 shows the half normal plot for dHAGA where it can be clearly seen that factors C and A along with interaction AC are most significant. Those three terms, in addition to the other terms shown on the graph, were picked to be the model terms since they are far from the rest of points that almost make a straight line. Table E.1 is the ANOVA table for dHAGA.

![Half Normal plot](image)

Figure E.1  Half normal plot of effects for dHAGA
### TABLE E.1 dHAGA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2.5279615</td>
<td>7</td>
<td>0.361137</td>
<td>241.4733</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>A</td>
<td>0.2109202</td>
<td>1</td>
<td>0.21092</td>
<td>141.0311</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>B</td>
<td>0.0040905</td>
<td>1</td>
<td>0.00409</td>
<td>2.73507</td>
<td>0.1368</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2.0361901</td>
<td>1</td>
<td>2.03619</td>
<td>1361.491</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>AB</td>
<td>0.030875</td>
<td>1</td>
<td>0.030875</td>
<td>20.64449</td>
<td>0.0019</td>
<td>*</td>
</tr>
<tr>
<td>AC</td>
<td>0.2109202</td>
<td>1</td>
<td>0.21092</td>
<td>141.0311</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>BC</td>
<td>0.0040905</td>
<td>1</td>
<td>0.00409</td>
<td>2.73507</td>
<td>0.1368</td>
<td></td>
</tr>
<tr>
<td>ABC</td>
<td>0.030875</td>
<td>1</td>
<td>0.030875</td>
<td>20.64449</td>
<td>0.0019</td>
<td>*</td>
</tr>
<tr>
<td>Pure Error</td>
<td>0.0119645</td>
<td>8</td>
<td>0.001496</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>2.539926</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

The Model F-value of 241.47 implies the model is significant. There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise. Values of “Prob > F” less than 0.0500 indicate model terms are significant. In this case A, C, AB, AC, ABC are significant model terms.

Figure E.2 shows the normal plot of residuals for dHAGA. Figure E.3 shows the plot of residuals versus predicted values. Figure E.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response dat points. Here, no outliers are found since all points are within 3.5 standard deviations from zero. Figures E.5 and E.6 show the interaction graphs for the two factor interaction A-C, which was found out to be most significant along with factors C and A.
Figure E.2 Normal plot of residuals for dHAGA

Figure E.3 Plot of residuals vs. predicted values for dHAGA
Figure E.4 Plot of Outlier T for transformed dHAGA

Figure E.5  A - C interaction graph at B = low for transformed dHAGA
Figure E.6  A-C interaction graph at B = high for transformed dHAGA
APPENDIX F

dHASA RESULTS

For HASA percentage deviation from the lower bound (dHASA), the data were transformed using the power transformation method and the constants were chosen according to equation F.1.

\[ dHASA' = (dHASA + 1)^{-3} \]  \hspace{1cm} (F.1)

Figure F.1 shows the half normal plot for dHASA where it can be noticed that factors C and A along with interaction AC are most significant. Those three terms, in addition to the other terms shown on the graph, were picked to be the model terms since they are far from the rest of points that almost make a straight line. Table F.1 is the ANOVA table for dHASA.

![Half Normal Plot for transformed dHASA](image)

Figure F.1 Half normal plot of effects for transformed dHASA
TABLE F.1  dHASA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2.6089845</td>
<td>7</td>
<td>0.372712</td>
<td>2590.699</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>A</td>
<td>0.1018745</td>
<td>1</td>
<td>0.101875</td>
<td>708.1236</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>B</td>
<td>0.0016548</td>
<td>1</td>
<td>0.001655</td>
<td>11.50265</td>
<td>0.0095</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>2.3921575</td>
<td>1</td>
<td>2.392158</td>
<td>16627.74</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>AB</td>
<td>0.0048841</td>
<td>1</td>
<td>0.004884</td>
<td>33.94927</td>
<td>0.0004</td>
<td>*</td>
</tr>
<tr>
<td>AC</td>
<td>0.1018745</td>
<td>1</td>
<td>0.101875</td>
<td>708.1236</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>BC</td>
<td>0.0016548</td>
<td>1</td>
<td>0.001655</td>
<td>11.50265</td>
<td>0.0095</td>
<td>*</td>
</tr>
<tr>
<td>ABC</td>
<td>0.0048841</td>
<td>1</td>
<td>0.004884</td>
<td>33.94927</td>
<td>0.0004</td>
<td>*</td>
</tr>
<tr>
<td>Pure Error</td>
<td>0.0011509</td>
<td>8</td>
<td>0.000144</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>2.6101354</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

The Model F-value of 2590.70 implies the model is significant. There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise. Values of "Prob>F" less than 0.0500 indicate model terms are significant. In this case A, B, C, AB, AC, BC, ABC are significant model terms.

Figure F.2 shows the normal plot of residulas for the dHASA percentage deviation from the lower bound. Figure F.3 shows the plot of residuals versus predicted values, which looks satisfactory since the residuals appear to have a nearly constant variance. Figure F.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response dat points. Here, no outliers are found since all points are within 3.5 standard deviations from zero. Figures F.5 and F.6 show the interaction graphs for the two-factor interaction A-C, which was found out to be most significant along with factors C and A.
Figure F.2 Normal plot of residuals for transformed dHASA

Figure F.3 Plot of residuals vs. predicted values for transformed dHASA
Figure F.4  Plot of Outlier T for transformed dHASA

Figure F.5  A - C interaction graph at B = low for transformed dHASA
Figure F.6  A -C interaction graph at B = high for transformed dHASA
APPENDIX G

fHA RESULTS

For HA frequency of resulting in the best solution (fHA), the data were transformed using the square root transformation method and the constant was chosen according to equation G.1.

\[ fHA' = \sqrt{(fHA + 2)} \]  \hspace{1cm} (G.1)

Figure G.1 shows the half normal plot for fHA where it can be seen that factor C is most significant. This term, in addition to the other terms shown on the graph, were picked to be the model terms since they are far from the rest of points that almost make a straight line. Table G.1 is the ANOVA table for fHA.

Figure G.1  Half normal plot of effects for transformed fHA
### TABLE G.1 fHA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>104.529</td>
<td>3</td>
<td>34.843</td>
<td>443.38</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>B</td>
<td>0.76597</td>
<td>1</td>
<td>0.766</td>
<td>9.747</td>
<td>0.0088</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>102.997</td>
<td>1</td>
<td>103</td>
<td>1310.7</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>BC</td>
<td>0.76597</td>
<td>1</td>
<td>0.766</td>
<td>9.747</td>
<td>0.0088</td>
<td>*</td>
</tr>
<tr>
<td>Residual</td>
<td>0.94301</td>
<td>12</td>
<td>0.0786</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>0.53919</td>
<td>4</td>
<td>0.1348</td>
<td>2.6704</td>
<td>0.1105</td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>0.40382</td>
<td>8</td>
<td>0.0505</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>105.472</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

The Model F-value of 443.38 implies the model is significant. There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise. Values of "Prob > F" less than 0.0500 indicate model terms are significant. In this case B, C, BC are significant model terms.

Figure G.2 shows the normal plot of residuals for fHA. Figure G.3 shows the plot of residuals versus predicted values. Figure G.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response data points. Here, no outliers are found since all points are within 3.5 standard deviations from zero.
Figure G.2 Normal plot of residuals for transformed fHA

Figure G.3 Plot of residuals vs. predicted values for transformed fHA
Figure G.4 Plot of Outlier T for transformed fHA
APPENDIX H

fGSA RESULTS

For GSA frequency of resulting in the best solution (fGSA), there was no need to apply any transformation to the data. Figure H.1 shows the half normal plot for fGSA were it is shown that factor C is most significant. This term was picked to be the model term since it is far from the rest of points that almost make a straight line. Table H.1 is the ANOVA table for fGSA. The Model F-value of 94.11 implies the model is significant. There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise. Values of "Prob > F" less than 0.0500 indicate model terms are significant. In this case C is the only significant model term.

![Half Normal plot](image)

Figure H.1 Half normal plot of effects for fGSA
TABLE H.1 fGSA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>121</td>
<td>1</td>
<td>121</td>
<td>94.111</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>121</td>
<td>1</td>
<td>121</td>
<td>94.111</td>
<td>&lt; 0.0001</td>
<td>*</td>
</tr>
<tr>
<td>Residual</td>
<td>18</td>
<td>14</td>
<td>1.2857</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>12</td>
<td>6</td>
<td>2</td>
<td>2.6667</td>
<td>0.1001</td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>6</td>
<td>8</td>
<td>0.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>139</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

Figure H.2 shows the normal plot of residuals for fGSA. Figure H.3 shows the plot of residuals versus predicted values. Figure H.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response data points. Here, no outliers are found since all points are within 3.5 standard deviations from zero.

![Figure H.2 Normal plot of residuals for transformed fGSA](image)

Figure H.2 Normal plot of residuals for transformed fGSA
Figure H.3  Plot of residuals vs. predicted values for fGSA

Figure H.4  Plot of Outlier T for fGSA
APPENDIX I
fHAGA RESULTS

For HAGA frequency of resulting in the best solution (fHAGA), the data were transformed using the square power transformation method and the constants were specified as shown in equation I.1.

\[ f_{\text{HAGA'}} = (f_{\text{HAGA}} + 1)^3 \]  

(I.1)

Figure I.1 is the half normal plot for fHAGA showing that factors C and A along with interaction AC are most significant. Those three terms, in addition to the other terms shown on the graph, were picked to be the model terms since they are far from the rest of points that almost make a straight line. Table I.1 is the ANOVA table for the transformed fHAGA.

Figure I.1  Half normal plot of effects for transformed fHAGA
The Model F-value of 116.27 implies the model is significant. There is only a 0.01% chance that a "Model F-Value" this large could occur due to noise. Values of "Prob > F" less than 0.0500 indicate model terms are significant. In this case A, B, C, AC, BC are significant model terms.

Figure I.2 shows the normal plot of residuals for fHAGA. Figure I.3 shows the plot of residuals versus predicted values. Figure I.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response data points. Here, no outliers are found since all points are within 3.5 standard deviations from zero. Figures I.5 and I.6 show the interaction graphs for the two-factor interaction A-C, which was found out to be most significant along with factors C and A.
Figure I.2 Normal plot of residuals for transformed fHAGA

Figure I.3 Plot of residuals vs. predicted values for transformed fHAGA
Figure I.4 Plot of Outlier T for transformed fHAGA

Figure I.5 A - C interaction graph at B = low for transformed fHAGA
Figure I.6 A - C interaction graph at B = high for transformed fHAGA
For HASA frequency of resulting in the best solution (fHASA), there was no need to apply any transformation method to the data. Figure J.1 shows the half normal plot for fHASA where it is noted that factor C is most significant. This term was picked to be the only model term since it is far from the rest of points that almost make a straight line. Table J.1 is the ANOVA table for fHASA. The Model F-value of 21.81 implies the model is significant. There is only a 0.04\% chance that a "Model F-Value" this large could occur due to noise. Values of "Prob > F" less than 0.0500 indicate model terms are significant. In this case C is the significant model term.

**Figure J.1  Half normal plot of effects for transformed fHASA**
TABLE J.1 fHASA ANOVA

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Prob &gt; F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>81</td>
<td>1</td>
<td>81</td>
<td>21.808</td>
<td>0.0004</td>
<td>*</td>
</tr>
<tr>
<td>C</td>
<td>81</td>
<td>1</td>
<td>81</td>
<td>21.808</td>
<td>0.0004</td>
<td>*</td>
</tr>
<tr>
<td>Residual</td>
<td>52</td>
<td>14</td>
<td>3.7143</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>18</td>
<td>6</td>
<td>3</td>
<td>0.7059</td>
<td>0.6548</td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>34</td>
<td>8</td>
<td>4.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cor Total</td>
<td>133</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* significant at 5%

Figure J.2 shows the normal plot of residuals for fHASA. Figure J.3 shows the plot of residuals versus predicted values. Figure J.4 shows the outlier t plot, which allows the visualization of any outliers – unusual response data points. Here, no outliers are found since all points are within 3.5 standard deviations from zero.

Figure J.2  Normal plot of residuals for transformed fHASA
Figure J.3  Plot of residuals vs. predicted values for transformed fHASA

Figure J.4  Plot of Outlier T for transformed fHASA