CORE
Metadata, citation and similar papers at core.ac.uk
Provided by ethesis@nitr

Generation of Optimized Robotic Assembly Sequence using Soft Computing Methods

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology (Research)

in

Mechanical Engineering

by

Surajit Sharma
(Roll: 60703004)

Department of Mechanical Engineering
National Institute of Technology
Rourkela-769 008, Orissa, India

August 2009

Generation of Optimized Robotic Assembly Sequence using Soft Computing Methods

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology (Research)

in

Mechanical Engineering

by

Surajit Sharma

Under the supervision of

Prof. B.B. Biswal

Department of Mechanical Engineering
National Institute of Technology
Rourkela-769 008, Orissa, India

August 2009

National Institute of Technology

Rourkela-769 008, Orissa, India

CERTIFICATE

This is to certify that the work in the thesis entitled, "**Generation of Optimized Robotic Assembly Sequence using Soft Computing Methods**" submitted by **Mr Surajit Sharma** in partial fulfillment of the requirements for the award of **Master of Technology (Research) Degree** in the Department of Mechanical Engineering, National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the work reported in this thesis is original and has not been submitted to any other Institution or University for the award of any degree or diploma.

He bears a good moral character to the best of my knowledge and belief.

Place: NIT Rourkela
Date:

Prof. (Dr.) Bibhuti Bhusan Biswal
Professor
Department of Mechanical Engineering
National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

For each and every new activity in the world, the human being needs to learn or observe from somewhere else. The capacity of learning is the gift of GOD. To increase the capacity of learning and gaining the knowledge is the gift of GURU or Mentor. That is why we chanted in Sanskrit "*Guru Brahma Guru Bishnu Guru Devo Maheswara, Guru Sakshat Param Brahma Tashmey Shree Guruve Namoh*". That means the Guru or Mentor is the path of your destination.

The author first expresses his heartiest gratitute to his guide and supervisor *Prof. (Dr.) Bibhuti Bhusan Biswal*, Professor of Mechanical Engineering for his valuable and enthusiastic guidance, help and encouragment in the course of the present research work. The successful and timely completion of the research work is due to his constant inspiration and extraordinary vision. The author fails to express his appreciation to him.

The author is very much obliged to *Prof. (Dr.) Ranjit Kumar Sahoo*, Professor and Head of the Department of Mechanical Engineering for his continous encouragement and support.

The author is thankful to *Prof. (Dr.) D. R. K. Padhi*, Professor of Mechanical Engineering and *Prof. (Dr.) A. K. Panda*, Professor of Electrical Engineering, NIT Rourkela, for their support during the research period.

The author would like to give a special thanks to *Prof. (Dr.) Siba Sankar Mahapatra*, for his friendly activities, the sharing of knowledges of soft computing technology and continous and enjoyable encouragement during the research period. The author cannot express his appreciation to him.

The help and cooperation received from the author's friend-circle, staff of Department of Training and Placement, staff of Department of Mechanical Engineering is thankfully acknowledgeed.

Last but not the least, the author is forever indebted to his parents, his wife, and his daughter for their prayer, understanding and moral support during the tenure of his research work.

Surajit Sharma

ABSTRACT

The assembly process is one of the most time consuming and expensive manufacturing activities. The cost of assembly on an average is 10-30% of the manufacturing cost of a commercial product. The ratio between cost and performance of assembly has gradually increased with respect to the other phases of the manufacturing process and in recent years, this fact has caused a growing interest by industry in this area. *Robotic assembly system which comes under the automated assembly system* incorporates the use of robots for performing the necessary assembly tasks. This is one of the most flexible assembly systems to assemble various parts into desired assembly (usable end-product). Robotic assembly systems are the programmable and have the flexibility to handle a wide range of styles and products, to assemble the same products in different ways, and to recover from errors. Robotic assembly has the advantage of greater process capability and scalability. It is faster, more efficient and precise than any conventional process. A variety of optimization tools are available for application to the problem. It is difficult to model the present problem as an *n-p* problem. Finding the best sequence generation involves the conventional or soft-computing methods by following the procedures of search algorithms.

Determination of a correct and stable assembly sequence is necessary for automated assembly system and especially for robotic assembly system. Assembly sequence affects flexible and advanced manufacturing systems in many aspects such as cost, factory layout, and use of tools and so on. This contains the information on directional connections for each pair of mating components and gives little idea about the fixture design, which affects the complexity and dexterity of robots. To cope with the problem of query or time consuming geometric reasoning in assembly sequence, this research proposes a specific approach to determine stable robotic assembly sequences, taking into consideration of the

instability of assembly motions in three mutually orthogonal directions. The objective of the present work is to generate feasible, stable and optimal robotic assembly sequence satisfying the assembly constraints with minimum assembly cost. The present research aims at evolving an approach for generating robotic assembly sequences using the evolutionary technique considering of the instability of assembly motions and/or directions. As a prerequisite to the assembly sequence generation, the assembled states of parts in a product must be adequately described. Such a description, called the product modeling, will be utilized to infer the assembly constraints, and to evaluate the assembly cost.

The thesis describing the present research work is divided into six chapters. The subject of the topic its contextual relevance and the related matters including the objectives of the work and the methodology to be adopted are presented in Chapter 1. The reviews on several diverse streams of literature on different issues of the topic such as assembly sequence generation, disassembly sequence generation, assembly sequence representation, optimization techniques etc. are presented in Chapter 2. In Chapter 3, selected methods are explained and applied on number of suitable products. Chapter 4 presents generation of stable assembly sequences using Ant Colony Optimization (ACO) method and Artificial Immune System (AIS) for the generation of robotic assembly sequence. In Chapter 5, the pros and cons of different methods have been discussed and their suitability has been checked in view of the robotic assembly process. Finally, Chapter 6 presents the conclusion and future scope of the research work.

A systematic method of generating correct assembly sequence for robotic assembly has been proposed. The method is based upon the evaluation of base assembly motion instability along with part contact level graphs that infers the precedence constraints. The modeling procedure takes into consideration topological relationship between parts constituting the product assembly. The assembly sequence thus obtained is a stable sequence. To formulate a scheme for generating such a stable sequence, instability of base assembly movement has been defined utilizing the instability of each individual part motion. This facilitates the determination of the stable sequences. Among these assembly sequences, one can find the most desirable sequence which yields the minimum number of direction changes in base assembly movement. This is due to the fact that this sequence requires a simple design of fixtures and the minimal dexterity of robots. This procedure has been applied to a discriminator. It is claimed that the stable assembly sequences for robotic assembly can be efficiently inferred by utilizing the proposed inference method. The development of a procedure to cluster parts into subassemblies to obtain a hierarchical model of the assembly and the development of good heuristics to guide the generation of assembly sequences are issues for future research. While looking the assembly point of view the proposed method gives out the better results, but to quantify the obtained result is very difficult. To elaborate the effectiveness of the method, two soft-computing methods are applied to generate the optimized sequence(s).

A new modified optimization algorithm is presented to generate optimal stable assembly sequence. The present algorithm maintains population diversity by the immune operations of immune selection and immune metabolism, and employs a clonal selection operation to enhance the local search. Furthermore, the immune operation of inoculation is incorporated to improve the quality of solution candidates to speed up convergence. The approach is able to produce results similar or better than those generated by other algorithms that are representative of the state-of-the-art in evolutionary objective optimization. The present approach uses an affinity measure to control the amount of mutation to be applied to the antibodies. These affinity measures, combined with the population are used to distribute non-dominated solutions in a uniform way. The approach proposed uses a very simple mechanism to deal with assembly constrained functions, and the results indicate that such mechanism, despite its simplicity, is very effective in practice. The results show that the evolved method could be effectively utilized to generate stable assembly sequences.

CONTENTS

Certificate i

Acknowledgement ii

Abstract iv

Contents vii

List of Figures x

List of Tables xi

Nomenclature xii

Abbreviation xv

Chapter 1: Introduction

1.1 Overview 1

1.2 Assembly in design and manufacturing 2

1.3 Evolution of assembly 3

1.4 Stages in assembly 3

1.5 Classification of assembly 4

  1.5.1 Manual assembly 5

  1.5.2 Semi automatic assembly or hard automation 5

  1.5.3 Automated assembly or soft automation 5

1.6 Comparision of assembly methods 7

1.7 Benefits of robotic assembly 8

  1.7.1 Standardization of processes 9

  1.7.2 Rules to simplify the design and the assembly process 10

1.8 Importance of assembly sequence 10

1.9 Reasons for automated assembly 12

1.10 Optimization of assembly sequence 14

1.11 Objective of research 16

1.12 Methodology 17

1.13 Outline of thesis 17
1.14 Summary 18

Chapter 2: Literature Survey

2.1 Overview 19
2.2 Some important literatures related to the present work 20
2.3 Assembly sequence generation 22
2.4 Assembly sequence generation for robotic applications 25
2.5 Optimization of assembly sequence 27
2.6 Softcomputing techniques for optimization of assembly sequence 28
2.7 Summary 31

Chapter 3: Generation of Assembly Sequences

3.1 Overview 32
3.2 Product modeling for assembly sequence generation 34
3.3 Assembly constraints 38
3.4 Part contact level graph 39
3.5 Directional precedence constraint 41
3.6 Stable assembly sequences 47
    3.6.1 Part instability 47
3.7 Summary 51

Chapter 4: Assembly Sequence and Soft Computing Methods

4.1 Overview 52
4.2 Ant algorithm concept 54
4.3 Artificial immune concept 56
4.4 Approach for solution 59
4.5 Objective function of assembly sequence generation 61

4.5.1 Degree of motion instability and number of assembly direction changes 62
4.6 Applying ACO to ASG 65
4.6.1 Solution method 67
4.6.2 ACO algorithm 69
4.7 Applying AIS to ASG 71
4.7.1 AIS algorithm 73
4.8 Summary 75

Chapter 5: Results and Discussions

5.1 Overview 77
5.2 Conventional robotic assembly sequence generation 78
5.2.1 Discussion 81
5.3 Ant colony optimization algorithm applied to robotic assembly 82
5.3.1 Simulation for C_J, C_P, C_C 82
5.3.2 Results and discussions 84
5.4 Artificial immune system algorithm applied to robotic assembly 85
5.4.1 Discussion 89

Chapter 6: Conclusions and Future Scopes

6.1 Overview 90
6.2 Importance and usefulness 91
6.3 Scope for future work 93

References 94
Appendices 98
Curriculum Vita 111

List of Figures

Figure	Title	Page
Figure 1.1	The classification of assembly systems	4
Figure 1.2	Relative costs of different assembly methods by type and production volume	7
Figure 1.3	The appropriate ranges for each type of assembly method	7
Figure 3.1(a)	A simple example of a product (Grinder assembly)	36
Figure 3.1(b)	Directions for assembly or disassembly	36
Figure 3.1(c)	Liaison graph model of grinder.	36
Figure 3.2(a)	An exploded view of the discriminator	43
Figure 3.2(b)	The three mutually perpendicular axes for assembly direction	43
Figure 3.3	The liaison diagram of discriminator	44
Figure 3.4	Adjacent matrix AM_d for the Discriminator in the z-axis direction	45
Figure 3.5	Part contact level graph of discriminator	46
Figure 3.6	Motion of a part in free space	48
Figure 3.7	The directions of assembly by a robot	49
Figure 4.1	Disassembly parts of grinder	64
Figure 4.2(a)	Hierarchical tree structure of possible assembly directions	64
Figure 4.2(b)	Ordered list of possible assembly directions	64
Figure 4.3	Flow diagram of the ACO procedure	70
Figure 4.4	The flow diagram of the immune algorithm	74
Figure 5.1	The stable assembly sequence of the parts of discriminator after going through instability	79
Figure 5.2	The simulation result for C_P, C_J, C_C	83

List of Tables

Table	**Title**	**Page**
Table 2.1	Important literatures related to assembly sequence generation	20
Table 3.1	Part description of grinder assembly	36
Table 5.1	Feasible assembly sequence	80
Table 5.2	Stable assembly sequence	80
Table 5.3	The simulation condition for energy constant	83
Table 5.4	The simulation condition for ant parameters	83
Table 5.5	Affinity value of the stable sequences	85
Table 5.6	Result of cloning procedure	86
Table 5.7	Mutation (Inverse and/or Pairwise) result of AIS algorithm	86
Table 5.8	Receptor editing result of AIS algorithm	87
Table 5.9	Assembly sequences after one iteration	87
Table 5.10	Assembly sequence after 900 iterations	88

Nomenclature

Symbol	
L_i	Liaison of the i^{th} and $(i+1)^{th}$ components
n	Number of parts
$l_{\alpha\beta}$	Liaision between part α and β
p_{α}	α^{th} component of assembly product
p_{β}	β^{th} component of assembly product
$C_{\alpha\beta}$	Contact-type connection matrix
$f_{\alpha\beta}$	Fit-type connection matrix
d	Assembly direction
C_d	Directional contact connection
f_d	Fit type element
rc	Real contact in *d* direction between two parts
vc	Virtual contact in *d* direction between two parts
sw	Screwing in *d* direction
rf	Round peg-in-hole fit
mp	Multiple round peg-in-hole fit
pf	Polygon fit
tf	Tight fit
ca	Caulking
ri	Riveting
vf	Virtual fit
0	No fit
n_p	Set of parts
PC	Precedence Constraints
CL_d	Part contact level
G_d	Directional part contact level graph

P_d	Set of parts interconnected with contact type connections
L_d	Set of all the fit with the contact type connections in d directions
m_{ij}	Assigned value, when the part p_j makes a single contact with the part p_i in the direction d
AM_d	Adjacent matrix in d direction
$T(p_k)$	Translational instability matrix
$R(p_k)$	Rotational instability matrix
t_δ	Translation in δ direction
r_δ	Rotation in δ direction
$S(p_k)$	Motion instability matrx
$S\{p_k(l_{jk})\}$	Instability matrices of directional connections established by the liaison l_{jk}
c_δ	Directional contact connection
PF	Fixed base part
PU	Unstable base part
τ_{ij}	Pheromone matrix
E_{seq}	Energy function associated with ASG
E_{J}	Energy related with Assembly cost
E_{P}	Energy related with Precedence constraints
E_{C}	Energy related with Connectivity constraints
J	Assembly cost
C_P	Energy constant related with Precedence constraints
C_C	Energy constant related with Connectivity constraints
C_J	Assembly constant related with costing
C_{as}	Normalized degree of motion instability
C_{nt}	Normalized number of assembly direction changes
ρ_s	Cost constant related to normalized degree of motion instability

ρ_t	Cost constant related to normalized number of assembly direction changes
μ_i	Precedence index
λ_i	Connectivity index
DS	Possible assembly direction set
DL	Assembly order list
BA	Base assembly
DM	Disassembly interference matrix
I_{ijd}	Interference between parts i and j in d direction
m	Number of ants
H	Heuristic value
ρ	Pheromone evaporation rate
NC	Iteration counter
p	Antibody

Abbreviations

ACO	Ant Colony Optimization
AIS	Artificial Immune System
TSP	Traveling Salesman Problem
ASG	Assembly Sequence Generation
GA	Genetic Algorithm
IOA	Immune Optimization Approach
MMAS	Max-Min Ant System
PCLG	Part Contact Level Graph
AM	Adjacent Matrix
FBP	Feasible Base Part
CL	Contact Level
EA	Evolutionary Algorithm
DO	Disassembly Operation
U	Boolean operator OR
DM	Disassembly matrix

CHAPTER 1

Introduction

1.1 Overview

Progress in modern society depends on the engineering industrial growth. Engineering industries of today are bound to use advance manufacturing methods to achieve high quality products, high volume of production and competitive cost of manufacturing. Manufacturing and engineering design in industry require specialized knowledge and problem solving techniques. Emerging soft-computing technique can facilitate part design, process planning, scheduling, understanding and diagnosis and effective sequence determination to assemble a product. Products from every manufacturing industry cover a very broad range - starting from toys to aerospace. Every one of these products has some specific characteristic in common with every other product. The common characteristic is that, the product itself consists of number of parts that must be joined together to form a single entity/finished product. In other words, the parts must be assembled to form the desired product. Without the ability to assemble products, manufacturing companies could not manufacture, and hence their existence in world would really be difficult. Assembly is the process of joining separate components together to form a single final assembled unit (e.g. mechanism, device, building etc.). A single assembly task involves joining two or more components or subassemblies together. In many cases, the order in which these tasks are performed is an important consideration. Firstly, many such orders may not be feasible because of physical constraints such as accessibility and stability of assembly. It is also possible that many feasible sequences exist, but some are more desirable than others according to criteria such as the need for jigs or fixtures, the number of tasks that can be performed simultaneously and so forth. Assembly planning can be defined as the process of identifying an assembly plan, which defines either a complete or partial order in which the assembly tasks can be performed.

1.2 Assembly in design and manufacturing

An assembly operation is a type of manufacturing operation, in which two or more separate parts are joined to form a new subassembly or a new entity. In other words, a product is assembled by repeatedly joining one or several parts/subassemblies to form larger subassemblies until a target product is obtained. Assembly plays an important role in the design and manufacturing stages or integration of both. A product designer must consider a variety of factors such as functional goals, fabrication, and manufacturing, ease of use and future repair of products. A determining factor in manufacturing system is the sequence of assembly, itself a function of product design. Selecting a good assembly sequence has many significant advantages, including easier assembly moves, reduced need for fixtures, less rework, better testing opportunities, less risk of part mating, and lower unit assembly cost. A wrong selection of sequence may lead to either a wrong product or halting of the process from moving towards next stage. Sometimes there may be possibility of parallel operation of assembly for the same end product thereby speeding up the assembly process, compared to sequential step-by-step assembly processes where parts are put together only one at a time. The assembly professionals need to be interested, concerned and responsible for the successful and profitable application of assembly technology in manufacturing plants. In order to meet these expanding needs and to survive and grow in the global competitive market, modern factories use a high level of automation to build products. Automation provides the opportunity to make the products faster and better with lower cost and higher quality. Sequence of assembly of a set of parts plays a key role in determining important characteristics of the tasks of assembly and of the finished assembly. Matters, such as the difficulty of assembly steps, the needs for fixturing, the potential for parts damage during assembly, are all affected by assembly sequence choice.

1.3 Evolution of assembly

Research in assembly has gained momentum mainly in the last two decades. The reason why it emerged so late as a separate research topic in manufacturing and why it is so active today could be explained considering the degree of automation and it is directly associated with economic considerations and the complexity of optimization in assembly. Prior to that, most effort in manufacturing research was directed at primary manufacturing processes. It became clear that the potential for savings resulting from improvements in this sector was not covering the cost of the effort (an asymptotic evolution). It was, also, recognized that one virtually untapped source to reduce costs was assembly which, with limited exceptions, was still being carried out in the same way as almost a century ago. The degree of automation in assembly is considerably lower than that in manufacturing of parts; hence the costs with labour are considerably higher in assembly.

1.4 Stages in assembly

In production system a major role is performed by assembly process. Assembly involves several stages, such as, finding the relative position of components, part handling and accessing, selection of fixtures and part placement devices, and assembly sequence determination. Sequence of assembly of a set of parts plays a key role in determining important characteristics of the tasks of assembly and of the finished assembly. It involves the identification, section and sequencing of assembly operations. The sequencing of assembly operations usually leads to the set of all feasible assembly processes. Assembly sequence gives a feasible order in which parts are to be assembled together. The consideration that need to be made in a feasible assembly process are: the difficulty of assembly steps, the needs for holding parts/subassemblies, the potential for parts damage during assembly, the ability to do in-process testing, the occurrence of need for rework, the unit cost of assembly etc. These considerations influence the productivity of the process, product quality, and the cost of production. The assembly process is one of the most time consuming and expensive manufacturing activities. In today's world the cost of assembly on an average is 10-30% of the manufacturing cost of a commercial product [4]. The ratio between cost and performance of assembly has gradually increased with respect to the other phases of the manufacturing process and in recent years, this fact has caused a growing interest by industry in this area.

1.5 Classification of assembly

The assembly system can be broadly divided into the following three categories:

1) Manual Assembly
2) Semi automatic assembly (hard automation)
3) Automated assembly (soft automation).

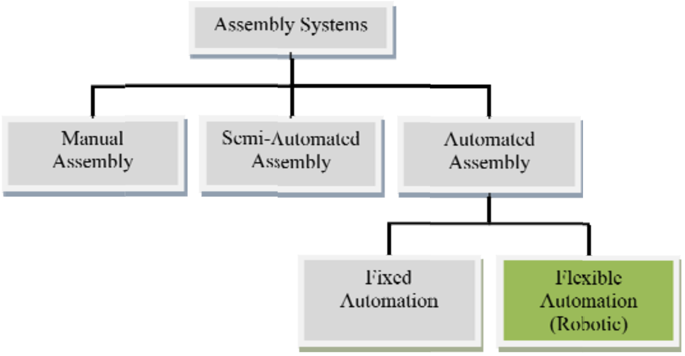


Figure 1.1: The classification of assembly systems

The classification of assembly system is presented in Figure 1.1.

1.5.1 Manual assembly

In manual assembly, parts are transferred to workbenches where workers manually assemble the product or components of a product. Hand tools are generally used to aid the workers. Although this is the most flexible and adaptable of assembly methods, there is usually an upper limit to the production volume, and labour costs (including benefits, cases of workers compensation due to injury, overhead for maintaining a clean, healthy environment, etc.) are higher. With manual assembly there is always the possibility of human errors. One way to make sure the mistakes won't happen, or that they are noticed in the earliest possible phase, is to use Poka-Yoke. Poka-Yoke is Japanese for mistake proofing. This means that the product's design, and its manufacturing processes and tools are as simple and appropriate for the purpose as possible. In addition, the benefits of mistake proofing come up to correct user operation and maintenance of the product, and service of the product.

1.5.2 Semi automatic assembly or hard automation

Fixed or hard automation is characterized by custom-built machinery that assembles one and only one specific product. Obviously, this type of machinery requires a large capital investment. As production volume increases, the fraction of the capital investment compared to the total manufacturing cost decreases. Indexing tables, part feeders, and automatic controls typify this inherently rigid assembly method. Sometimes, this kind of assembly is called "Detroit-type" assembly.

1.5.3 Automated assembly or soft automation.

With increasing demand for product variety, high rate of product-renewal in the context of present day’s global competition and growing need for faster rate of production, programmable automation is poised to be an inseparable part of individual establishments. Robotic assembly system which comes under the automated assembly system incorporates the use of robots for performing the necessary assembly tasks. This can take the form of a single robot, or a multistation robotic assembly cell with all activities simultaneously controlled and coordinated by a central controller. Although this type of assembly method can also have large capital costs, its flexibility often helps offset the expense across many different products. Robotic assembly systems are the programmable and have the flexibility to handle a wide range of styles and products, to assemble the same products in different ways, and to recover from errors or other unexpected events that cause the execution of the assembly to deviate from the preplanned course of action. However, it is critical that the entire robotic assembly work cell be precisely calibrated for most assemblies to be successfully accomplished. Precisely calibrating the entire work cell is tedious and expensive. An alternate to precise calibration is to use sensor feedback is sometimes incorporated in to assembly systems; however, force feedback is often difficult to use to instabilities that arise when contact is made and due to the poor signal-to-noise ratio that force/torque sensors provide. Partly because of this reasons, the use of force feedback in industrial applications has been limited. Other types of sensors may be used in conjunction with force sensors for assembly, so the problems encountered with force feedback can be addressed. Vision sensors technologies but have only been used within a limited framework. In a robotic assembly work cell, however, a single camera position will almost surely be inadequate for providing sufficient visual feedback throughout the assembly process. A camera mounted at the end-affecter of a manipulator could be servoed based on the assembly actions occurring in the work cell. Even with flexibility of the mechanical hardware, current robotic assembly systems are limited due to the inadequate data structure for representation of task plans.

1.6 Comparison of assembly methods

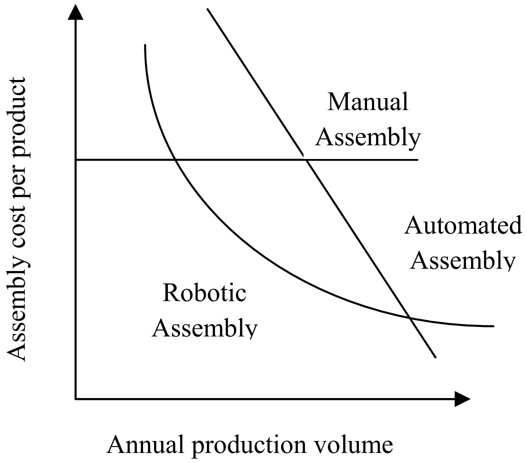


Figure 1.2: Relative costs of different assembly methods by type and volume

Graphically, the cost of different assembly methods can be displayed as in Figure 1.2. The non-linear cost for robotic assembly reflects the non-linear costs of robots (even small ones cost a lot).

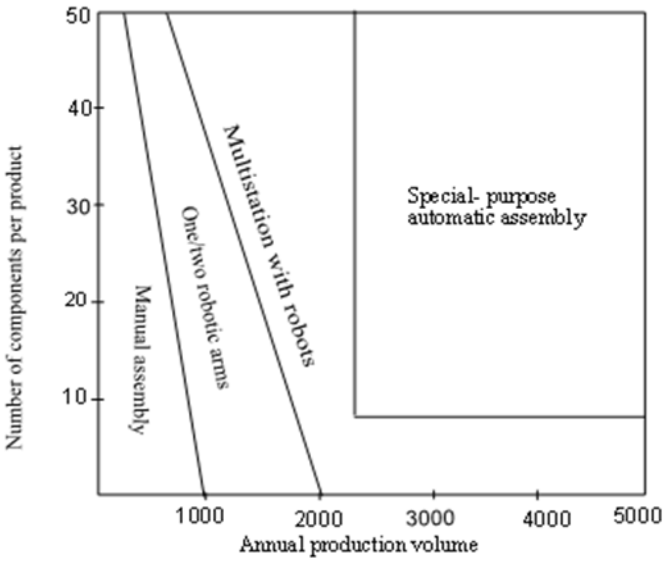


Figure1.3: The appropriate ranges for each type of assembly method

The appropriate ranges for each type of assembly method are shown (approximately) in Figure 1.3 assembly methods should be chosen to prevent bottlenecks in the process, as well as lower costs.

1.7 Benefits of robotic assembly

The manual assembly approach is likely to take much time and less qualitative assembly as compared to the automated one. In automated assembly, robotic assembly forms a major share. Robotic assembly is one of the most flexible assembly systems which utilize robots as its operators to assemble various parts into desired assembly (usable end-product). Robotic assembly has the advantage of greater process capability and scalability. It is faster, more efficient and precise than any conventional process. Robots save workers from tedious and monotonous assembly line jobs, and bring in increased production and savings in the process. Robotic assembly system is characterized by no fatigue, more output, better performance, standardization, and flexibility. An increasing requirement for flexibility of production is motivated by changes in technology and by customers demand for greater product variety. The main method of providing the desired flexibility is the development of flexible assembly systems (FAS), equipped with assembly robots. Robots play an important role in flexible assembly systems.

The following specific points are discussed on the benefits of robotic assembly;

i. Non-stable product design: If there is any change in product, the robot can be reprogrammed accordingly. However, this does not usually apply to the peripheral items in the system that contact the parts such as feeders, grippers etc.
ii. Production volume: A robot system can operate economically at much longer station cycle times than '5S' system of standardize cleanup. (Seiri: tidiness, Seiton: orderliness, Seiso: cleanliness, Seiketsu: standards, Shisuke: sustaining discipline).

iii. Style variations: A robot can more readily be arranged to accommodate styles of the same product.
iv. Fluctuation in demand: A truly programmable assembly system could be switched to different products according to demand.
v. Part defects: Firstly, it is interesting to note that a feeder jam caused by a faulty part causes much greater loss in production on a high-speed transfer assembly machine than on a robot system with a relatively long cycle time. In addition, the robot can be programmed to sense problems that may occur and re-attempt the insertion procedure.
vi. Part size: The part can be presented in patterns or arrays on pallets or part trays. In this case the severe restrictions on part size and high-speed automation do not apply.

1.7.1 Standardization of processes

In manufacturing of parts, highly standardized processes are available. Typically, machines for part manufacturing are process-oriented. Mechanization and automation in manufacturing of parts permit efficient production with a low proportion of labour costs. Assembly is product-oriented, low-or non-standardized due to high variability of tasks for handling, joining, adjusting and testing. Assembly automation for average or low volumes for a product has been difficult to justify economically, or infeasible due to the complexity of the problems. Assembly, as the final production stage, must cope with continuously shifting market requirements with regard to timing, batch sizes and product design or style, thus making it sensitive to any changes and requiring a flexibility that is not always possible.

1.7.2 Rules to simplify the design and the assembly process

a) Minimize number of parts: Reduced numbers of parts means a reduction in the number of assembly operations, hence, in most cases, the reduction in assembly cycle time. This is to be applied during conceptual and detail design stage.
b) Design for ease of handling: This is determined during the design phase and it depends on the specific part property(ies).
c) Design for ease of insertion: good insertion can be designed into components in the detail design stage by having suitable tolerances and chamfers on the mating parts.
d) Standardize parts, processes and methods: The overhead for a company for adding a part to their inventory, a process or a method is significant.

1.8 Importance of assembly sequence

An important aspect of this developing process is represented by the need to automatically plan the assembly of a product or, in other words, to identify the best sequence of operations and program the machines to perform the required tasks. That means there is a need for generation of an appropriate sequence of components of a product/ an assembly from the available choices of assembly sequences. Exploring the choices of assembly sequence is difficult for two reasons. First, the number of valid sequences can be large even at a small parts count and can rise staggeringly with increasing parts count. Second, seemingly minor design changes can drastically modify the available choices of assembly sequences. At the same time robotic assembly systems are more qualitative and cost effective. This directly influences the productivity of the process, product quality, and the cost of production. The product to be economically competitive, it is necessary to generate an appropriate sequence which minimizes the assembly cost as well as satisfies the assembly conditions. To mitigate the market value, it is necessary to generate a proper sequence of assembly which minimizes the assembly cost. The assembly conditions may involve the precedence constraints and the connectivity constraints.

The precedence constraint is a set of parts that must be connected before a pair of parts is mated. The connectivity constraints, is the connective relationships between the two parts. It states that, a part to be assembled onto an in-process subassembly must have at least one real connection with some part belonging to the in-process subassembly. Assembly sequences that satisfy the assembly constraints are called the feasible assembly sequences. The feasible assembly sequences do not always guarantee the parts to fix onto an in-process subassembly; parts may be loosely connected, and may come apart during the handling. Assembly sequences that maintain the stability of in-process subassembly movement are called the stable sequences, by means of which the parts can be successfully assembled to form an end-product.

An assembly is said to be potentially stable if the rigid bodies are in static equilibrium under the influence of external and internal forces. External forces arise from gravitation and internal forces arise from the mutual contact of the objects. Any reorientation of the assembly or the additional usage of fixtures reduces the flexibility and efficiency of the corresponding assembly process and thus results in additional costs. Since, in robotic assembly method, assembly is executed by robot(s), avoidance of unstable configuration shall result in reduction of sensing costs for the determination of the position and orientation of the objects in the workspace. Therefore, it is important to know, whether a configuration of the objects remains stable during or after the assembly process. This work presents an approach for generating robotic assembly sequences, considering the part motion instability of base assembly motions. In addition, the approach also uses the precedence constraints of the parts and three mutually orthogonal directional part contact level graphs to determine the correct assembly sequence for robotic assembly.

Assembly products with large number of parts have several alternative feasible sequences. Among these explosively increasing feasible sequences, finding the best assembly sequence is one of the most critical problems. If the problem is approached by traditional computational methods, it requires the generation of all the possible assembly sequences and their comparison by adopting proper evaluation criteria. The method can often produce combinatorial explosions of alternatives to be analyzed, with the consequence of unacceptable computational times. However, several alternative feasible sequences may be obtained while determining assembly sequences. Robotic assembly being a cost intensive process, it is necessary to find out the optimal sequence with the constraints of the process in mind.

1.9 Reasons for automated assembly

a) Cost reduction: Reduction of indirect labor, such as lead men and women, area supervisors, pay roll personnel clerks and inspectors, is a potential area of indirect labor savings. Additionally, well planned automated assembly should substantially reduce work-in-process inventories. Because of substantial productive capacity through assembly automation, manufacturers can keep completed parts inventory to the lowest reasonable levels.

b) Direct labor cost reduction: The cost of guaranteed annual wages is another indirect cost. Each person hired in the assembly area creates a potential liability for future operating costs. On the other hand, automatic assembly equipment can be turned off and incur no additional costs other than the depreciation expenses in the first years of production incurred during a period of idleness. Again, cyclical employment involves training of new workers and retraining of reassigned workers, with substantial cost penalties in learning curves and in assembly quality.

c) Marketing considerations: Knowledge that automated assembly systems are used to produce can give the marketing department significant psychological support and increased flexibility in the market place. Successful assembly systems provide psychological reassurance and competitive advantages in the market place.

d) Lower unit cost: Successful automated assembly has at least in theory, reduced assembly cost. Once this is achieved, low cost labor in other areas of the world no longer offer any labor cost advantage. Because of the low labor content and reduced factory size with automated assembly, it is often possible to place the assembly systems, reducing pipe line cost & delivery time.

e) Improved deliveries: Automated assembly systems are not only rapid in their assembly function, but they eliminate the delays and probable quality deterioration resulting from material handling between single station and manual assembly processes. Assembly lead times are dramatically shortened when production demands are seasonal. The productivity of most good assembly systems means that assembly and hence deliveries can be made as close to time of sale as possible. Cost of inventory is drastically reduced.

f) Uniform quality: Automated assembly with its capabilities of inspecting 100% of the assemblies produced and if necessary to document and even codify individual units of production offers the greatest assurance that recalls, field service problems, and warranty expenses with their negative impact on sales are held to an absolute minimum. In addition, the use of automated assembly forces uniform product quality, a consistency of quality that may be relied on both by sales people & customers.

g) Reduced warranty expenses: Automated assembly can often be fully justified by the reduction in warranty costs alone achieves through total functional inspection of the product during the assembly process.

h) Supply side economics: One of the fundamental assumptions of supply side economics is that inflation will be controlled and economic stability restored through adequate capital expenditure in productive facilities that supply can cope with demand. Automated assembly not only reduces direct impact & indirect costs but it tends to stabilize assembly costs when other costs are constantly rising.

1.10 Optimization of assembly sequence

The problem of sequencing has a primary role in the development of computer-based assembly systems. In these last two decades this stimulating subject has been debated very frequently in the scientific world, and this fact is demonstrated by the very wide literature concerning this topic; several kinds of algorithms have been developed and tested to generate feasible sequences and to select the most efficient ones. Nevertheless, some open issues still exist and currently prevent a fully automated solution to this problem. These issues mainly concern: integration with CAD, feasibility check of sequences, selection of the best assembly sequence. Different methods have been studied to approach this problem; the most efficient ones are based on the application of metaheuristic rules aimed to drastically reduce the number of sequences. These pruning methods are undoubtedly very effective and powerful, since they cut off many sequences that, according to the adopted heuristics, should not represent optimal solutions to the problem. However, the extension of this cut is often not predictable and not directly controllable by the user, increasing the risk of excluding valid sequences. For this reason, it is very important to develop new procedures which simultaneously satisfy the two contrasting objectives of maintaining all the valid sequences and reducing the computational time to acceptable values. For this purpose, a promising approach to this problem seems to be the ant colony optimization (ACO) and artificial immune system (AIS). ACO algorithm model derived from the observation of real ants' behavior, and uses these models as a source of inspiration for the design of novel algorithms for the solution of optimization and distributed control problems. The basic idea of ACO algorithms is to imitate the cooperative manner of an ant colony to solve combinatorial optimization problems within a reasonable amount of time. Artificial immune system (AIS) is a computational intelligence paradigm inspired by the biological immune system, which has found application in pattern recognition, scheduling, control, machine-learning and information systems security. AIS algorithm incorporates the features of clonal selection and immune memory to improve evolutionary search by identifying potential regions to explore while avoiding over emphasis in any region of the search space. Artificial immune system has emerged as a novel branch of computational intelligence. Inspired by the vertebrate immune system, AIS is a problem-solving paradigm. AIS has been successfully applied to the traveling salesman problem (TSP), the scheduling problem, capacitor placement, and assembly line balancing with encouraging results.

The present work focuses on the state of the art in assembly sequence generating (ASG). This critically examines various methods developed to solve and optimize the ASG problem, pointing out the limitations of each method. Other, related and subordinated, issues associated with assembly sequence generation such as; modeling, representation issues, optimization parameters, criteria and so on are addressed in subsequent, dedicated chapters. The research describes the approaches used to generate and search solutions for ASG problem. The searching methods are critically reviewed and their suitability and solving capacities are analyzed and justified based on parameters involved. Below mentioned section concentrates mainly on illustrating different approaches and techniques used in the attempt to optimize of the ASG problem. There is a huge difference between the two words; solving and optimizing. In ASG problem, after solving, the search result is more than one solution may be in terms of thousand based on number of individual parts in an assembly, the choice(s) are ‘better’ according to some criteria, is a rudimentary form of optimization. As a result, some approaches and techniques used for solving are common with S/O the ASP. Moreover, some approaches used for solving the ASG are embedded in the study. Optimizing the ASG is specially designed to determine the optimum or near optimum assembly sequence for assembling a product by searching most or all of the solution space. Thus, optimization considers a large number (and sometimes the totality) of assembly sequences of the product. Various aspects and issues involved are briefly analyzed and justified, and then the thesis structure is succinctly presented.

1.11 Objective of the research

Determination of a correct and stable assembly sequence is necessary for automated assembly system and especially for robotic assembly system. Assembly sequence affects flexible and advanced manufacturing systems in many aspects such as cost, factory layout, and use of tools and so on. This contains the information on directional connections for each pair of mating components and generates idea about the fixture design that decides the complexity and dexterity of robots. **T**his research proposes a specific approach to determine stable robotic assembly sequences, taking into consideration of the instability of assembly motions in three mutually orthogonal directions.

The study and analysis of some of the important literatures in the area of the automated generation of assembly sequences suggest that there is need to refine the methodologies with higher degree of assembly (having large and complicated parts) than that in the traditional methods to generate assembly sequences. Looking at the changing needs of automated industries and going through some of the relevant literatures by established authors. The objective of the present work is to generate feasible, stable and optimal robotic assembly sequence satisfying the assembly constraints with minimum assembly cost. The present research aims at evolving an approach for generating robotic assembly sequences using the evolutionary technique considering of the instability of assembly motions and/or directions. The broad objective of research work is outlined as follows.

i) To select some products of assembly that can be achieved with the help of automated assembly especially with robotic assembly.
ii) To develop new methodologies and modify some conventional methodologies for determining sequences of assembly for robotic assembly systems in a systematic manner.
iii) To develop and/or select suitable techniques for generation of optimized assembly sequence with the constraints of the products and process in mind.

1.12 Methodology

As a prerequisite to the subsequent assembly sequence generation, the assembled states of parts in a product must be adequately described. Such a description, called the product modeling, will be utilized to infer the assembly constraints, and to evaluate the assembly cost. This study adopts the modeling method previously proposed by Hong and Cho [20], in which products are modeled by a set of liaison data between parts. This section describes the modeling method of products, and the concept of assembly sequences. The objectives of the present research work are to be realized firstly by making an extensive study on the subject concerned and the research activities already carried out in the area and enumerating and analyzing the pros and cons of various methodologies. Secondly, considering the developments that have taken place and the needs of the process, a systematic way for generation of sequences of assembly for robotic assembly systems is proposed to be developed. This development of assembly sequences will look into the specific requirements of robotic assembly and that of the robot(s) under study besides the constraints posed by the components, process and the robotic system environment. A computer-based, generic, scalable and integrated optimization method for generation of assembly sequence is developed and tested for the benefit of the professionals. Finally, an appropriate method has been proposed with a view to achieve optimized robotic assembly sequence in relation with constraints, stability criteria and economic factor.

1.13 Outline of the thesis

The thesis describing the present research work is divided into six chapters. The subject of the topic its contextual relevance and the related matters including the objectives of the work and the methodology to be adopted are presented in Chapter 1. The reviews on several diverse streams of literature on different issues of the topic such as assembly sequence generation, disassembly sequence generation, assembly sequence representation, optimization techniques etc. are presented in

Chapter 2. In Chapter 3, selected methods are explained and applied on number of suitable products. Chapter 4 presents generation of stable assembly sequences using Ant Colony Optimization (ACO) method and Artificial Immune System (AIS) for the generation of robotic assembly sequence. In Chapter 5, the pros and cons of different methods have been discussed and their suitability has been checked in view of the robotic assembly process. Finally, Chapter 6 presents the conclusion and future scope of the research work.

1.14 Summary

The problem of sequence generation consists of a number of factors which cannot be modeled in mathematical terms. Only verbal reasoning and heuristic approaches are the only means of approaching the problem. Most of the time the solution to the problem is not unique. There will be multiple alternative sequences for the same product. As the number of parts increase the number of alternatives grows explosively. Hence it is not possible to examine each one of the alternatives to find out the optimum one by means of any traditional or mathematical means. This chapter presents the prevailing scenario in product assembly in today’s industry and the need for relook at the current practice. The objectives have been envisaged to help improve the process to a better standard and follow a systematic way for handling the problems.

CHAPTER 2

Literature Survey

2.1 Overview

Many studies in the last decade describe efforts to find more efficient algorithm for assembly sequence generation. This is due to the fact that, earlier described processes are time consuming and non-optimizing due to the problem of trapping in local optima. The generation of assembly sequences generally leads to ‘combinatorial explosion’ of the number of alternatives to analyze for checking and selecting the best assembly sequence and consequently, to unacceptable computational time. Different methods have been studied to approach this problem; the most efficient one are based on the application of meta-heuristic rules that aim to drastically reduce the number of sequences. The recent interest in robotic assembly and automatic generation of optimized robotic assembly plans has led to research on automatic generation of assembly sequences. For this reason, it is very important to develop new procedures which simultaneously satisfy the two contrasting objectives of maintaining all the valid sequences and reducing the computational time to acceptable values. There have been a lot of research work and experimental inferences for the generation of suitable and correct assembly sequences which is reflected through large number of literatures. The preliminary study of the subject necessitates a general review of the work carried out by various researchers. The relevant literatures are reviewed and discussed in relation to the methodologies and systems of implementing the above components or activities and towards an integrated environment for supporting the present goal set.

2.2 Some important literatures related to the present work

Table 2.1 presents some of the important work carried out on assembly sequence generation methods.

Table 2.1: Important literatures related to assembly sequence generation

Sl	Author(s)	Year	Topic
1	T. De Fazio and D. Whitney	1987	Develops a graphical method for generation of liaison sequences from the precedence relationship among the parts.
2	L. S. Homan de Mello and C. Sanderson	1989 1990 1991	Develops another graphical method to reduce the query and answer method for the generation of all mechanical assembly sequences.
3	Mascle and J. Figour	1990	Generates a constraint method which orderly disassembled the least constraint part at each step and obtains the assembly in its reverse.
4	Y. Huang and C. Lee	1989 1990 1991	Develops a classified method and gives the precedence knowledge in mating operation assembly planning.
5	S. H. Lee	1989 1990	In disassembly method, this reduces the $n!$ solutions to some extent.
6	Cho and H. Cho	1993	Works on Topological modeling and graphical inference which gives knowledge about part contact level graph

			and indication about the sequence generation.
7	D.Y.Cho, C.K.Shin and H.S.Cho	1993 1994	Develops a graph search or tree search method for automatic inference on stable robotic assembly sequences based upon motion instability.
8	D. S. Hong and H. S. Cho	1993 1995	A computational scheme has been developed for generation of optimized assembly sequence based on neural-network model.
9	D. S. Hong and H. S. Cho	1999	The same methodology has been tried by using simulating annealing.
10	M. Dorigo	1997	Use the ant algorithm in travelling salesman problem.
11	J. F. Wang, J. H. Liu and Y. F. Zhong	2005	The solution is based on assembly by disassembly philosophy by using ant colony algorithm.
12	L. N. De Castro and J. I. Timmis	2002	Develops a new computational intelligence approach by using artificial immune system.

2.3 Assembly sequence generation

De Fazio and Whitney [1] used a logical method through a set of questions that resulted in the desired precedence relationship among the parts. That means, the method generally uses two types of questions (‘Which connection must be established before connection L_i’ and ‘Which connections cannot be established before connection L’) to be asked. The number of questions to be asked is 2L, the method is far less time consuming but certain relations can be involuntarily omitted. The precedence relationships are used for the generation of assembly sequences. Also, the precedence relations don’t take alternative constraints into account, thus omitting a number of interesting assembly sequences. The techniques do not lead to failure if an obscure liaison is omitted or if conservatively too many are included. However, the method may be very reasonably applied to assemblies with parts counts in the teens or even tens.

Baldwin, Abell, MaxLui, De Fazio and Whitney [2] described an integrated computer aid that is useful for assembly line design and for concurrent design of mechanical products. The method built an integrated set of user-interactive computer programs that generates all feasible assembly sequences and then aids the user in judging their value based on various criteria. The programs use a disassembly analysis for generating sequences and provide on-line visual aids during generation and evaluation. The method generally is a cut-set method for the determination and representation of all geometric and mechanical assembly constraints as precedence relations.

Philip Chan [3] introduced the pattern matching system for the generation of automatic assembly sequence(s) considering that the problem of the part assembly relationship represented as a liaison can be solved in the same way as the traveling salesman problem, and developed a method of reducing the number of questions using the above mentioned method.

Sanderson and De Mello [4-7] used the information given by De Fazio and Whitney and process the connection diagram differently, posing far fewer questions for the same result. Their approach utilizes cut-sets of the connection diagram. The cut-set method developed an algorithm for correct and complete generation of mechanical assembly. In their approach the complete assembly is decomposed into distinct sub problems following a set of rules for validity and feasibility of the decomposed items, which are tested, on the relational model of the assembly. The methodology results in a valid and feasible disassembly sequence from which the target assembly sequence(s) is/are obtained. Since the process is tested for decomposition, the algorithm is claimed to be correct and complete. However, for completeness of methodology from robotic assembly viewpoint, the relational model should contain more relevant information and accordingly the decomposition algorithm will be influenced.

De Mello and Sanderson [5] analyzed high representation for assembly sequences: the directed graph of feasible assembly sequences, the AND/OR graph of feasible assembly sequences, the set of establishment conditions, and two types of sets of precedence relationships. The mappings of one representation into the others are established. The method describes that an assembly sequences is said to be feasible if all its assembly tasks are geometrically and mechanically feasible, and the input sub assemblies of all tasks are stable. Further, it is explained that an assemblies sequence could be represented in different ways: An ordered list of task representations, an ordered list of binary vectors, an ordered list of partitions of the set of parts, and an ordered list of subsets of connections. The authors claim that no previous precedence relationship representations of assemblies are correct and complete. However, the correctness and completeness of assembly sequences representation are established with mathematical proof.

Mascle and Figour [8] described a method which determines automatically the assembly/disassembly sequence of product units from the assembly drawing, the detail drawings and the nomenclature of parts. Assembly corresponds to a system in which flow parts and data are to be placed in specific order. The operative sequence is the basic program of this system. Generally, the approach generates a constraint method which orderly disassembled the least constraint part at each step and obtains the assembly in its reverse. The development is a methodological approach of sequences determination by using the disassembly analysis.

Huang, Lee and Shin [9-14] used a classified method to gain a framework of the precedence knowledge in mating operation during assembly of mechanical product. The method automatically generates an assembly procedure for products. The assembly generation procedure is based on two steps: i) each component in an assembly is located at a specific vertex of a hierarchical tree; and ii) then the assembly procedure is generated from the hierarchical tree with the help of interference checking. The suggested method can provide most of what the coding system expects from the user. However, in order to utilize the benefit of this method, the following areas need to be studied; i) a complete set of mating conditions must be identified, ii) more combinations of mating conditions for subassembly type virtual links must be identified, and iii) the program must be interfered with a geometric modeling systems for interference checking.

2.4 Assembly sequence generation for robotic applications

Cho and Cho [15] described a new approach to the automatic generation of assembly precedence constraints for robotic assembly, using a part level graph. The work utilizes three directional part contact level graphs which, in three orthogonal directions, contain the information on directional connections for each pair of mating parts. By using these graphs, an assembly precedence constraint is inferred in two steps: The first step infers a precedence constraint for each directional connection by applying the path-finding algorithm. Utilizing the precedence constraints thus obtained, the next step infers the precedence constraint for each part to be assembled with its base assembly.

Cho and Shin et.al. [16-18] presented an approach to the inference of robotic assembly sequences, taking into consideration the instability of base assembly motions. Based upon the evaluation of motion instability, the method generates stable assembly sequences by use of the precedence contact inference method.

Delchambre and Wafflard [19-21] presented a pragmatic approach to computer-aided assembly planning. This describes the formalism to analyze the product. The first step is one of the most important; the modernization of the product has to contain all the necessary information. With the help of the product analysis, the planner has to determine the feasible operational plans. The method describes the assembly planner itself and the formalism which they have adopted to describe the assembly plans. Several results are presented and discussed on the basis of a concrete case study.

Mosemann, Rohrdanz and Wahl [22] described that assembly planning and the subsequent execution of the generated plans by robots is one of the key technologies of modern and flexible manufacturing. The choice of an assembly sequence drastically affects the efficiency of the assembly process. They presented a high level assembly planning system (High LAP) covering all necessary aspects of high level assembly planning. High LAP takes the CAD descriptions of assembly components and high level assembly specifications with symbolic spatial relations as input. The assembly planning system generates and evaluates all assembly sequences of a mechanical assembly with minimal user interaction based on geometrical and physical reasoning. For the evaluation of all feasible assembly sequences several criteria are taken into account. For example stability, directionality, assembly pose, manipulability and parallelism are introduced and quantified. They claimed that the presented system is the first assembly planning system computing and taking into account the range of all stable orientations of an assembly considering friction for assembly plan evaluation.

Adachi et al [23-24] carried out the research on the representation of design information of assembly parts. The research describes the representation of an assembly structure, which includes the introduction of functional element. By using this functional element, constraint of the assembly structure can be naturally described. The functional element is used to express relations among the parts, and a relation can describe constraints among assembly parts which interfere without touching. The relation includes an assembly method and is capable of describing both constraint and interference. One demonstration has been carried out to reduce the search space of the assembly sequence.

2.5 Optimization of assembly sequence

Hong and Cho [25-26] proposed a neural-network-based computational scheme to generate optimized robotic assembly sequences for an assembly product. It is a hybrid intelligent system which uses both a neural network with functional link nets, and an expert system. Based on the assembly constraints inferred and the assembly costs obtained from the expert system, the evolution equations of the network are derived, and an optimal assembly sequence obtained from the evolution of the network.

Hong and Cho [27] stated that an assembly sequence was considered to be optimal when it minimizes assembly cost while satisfying assembly constraints. The assembly cost relates to assembly operations, assembly motions, and assembly direction changes. This study proposed a simulated-annealing method for the generation of such assembly sequences in robotic assembly. This method reflected the assembly cost to an energy function associated with the assembly sequence. The energy function is iteratively minimized and occasionally perturbed by a simulated annealing until no further change in the energy occurs. As a result, an assembly sequence with a low assembly cost was finally found. In order to show the effectiveness of the stated method, case study was presented for an electrical relay. The proposed method gives better performance in the generation of optimal sequence in comparison with the neural network-based approach.

2.6 Soft computing techniques for optimization of assembly sequence

Dorigo [28-30] introduced ant colony system which is a definition of a new computational paradigm. He proposes it as a viable new approach to stochastic combinatorial optimization. The main characteristics of this model are positive feedback, distributed computation and the use of a constructive greedy heuristic. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic helps find acceptable solutions in the early stages of the search process. This method is a distributed algorithm that is applied to the travelling salesman problem. In the ant colony system, a set of co-operating agents called ants cooperate to find good solutions to TSP's. Ants co-operate an indirect form of communication mediated by a pheromone they deposit on the edges of the TSP graph while building solutions.

Wahg, Liu and Zhong [31] used the ant colony algorithm-based approach to assembly sequence generation and optimize this of mechanical products. For diverse assemblies, the approach generates different amount of ants cooperating to find optimal solutions with the least reorientations during assembly processes. Based on assembly by disassembly philosophy, a candidate list composed by feasible and reasonable disassembly operations that are derived from disassembly matrix guides sequences construction in the solution space expressed implicitly, and so guarantees the geometric feasibility of sequences. The state-transition rule and local- and global-updating rules are defined to ensure acquiring of the optimal solutions.

Bonneville, Perrard and Henrioud [32] described a genetic algorithm that deals with the assembly planning problem. The method used a cut-set method in the form of genetic algorithm that generates and evaluates assembly plans. This algorithm starts from a set of valid assembly plans proposed by an expert of the product. This set is the initial population of potential solutions. Each assembly plan is encoded into a chromosome, to be manipulated by genetic operators. A reproduction process uses these operators to produce new assembly plans from parent assembly plans. An evolution function and a selection procedure retain the best plans that expand the population and serve for new generations.

Cao and Xiao [33] evolved a new branch of computational intelligence inspired by the vertebrate immune system called artificial immune system (AIS). This approach explored the application of AIS in the problem of assembly planning and proposed a novel approach, called the immune optimization approach (IOA), to generate the optimal assembly plan. Based on the bionic principles of AIS, IOA introduces manifold immune operations including immune selection, clonal selection, inoculation and immune metabolism to derive the optimal assembly sequence. Maintenance of population diversity, attention to the local as well as the global search and employment of heuristic knowledge to direct the search of optimized assembly sequences are the major concerns of IOA. The research is applied on two products to illustrate the validity of IOA in assembly planning, and encouraging solutions in quality and efficiency are achieved.

De Castro and Timmis [34] developed artificial immune system that can be defined as computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. Their development and application domains follow those of soft computing paradigms such as artificial neural networks, evolutionary algorithms and fuzzy systems. Despite some isolated efforts, the field of AIS still lacks an adequate framework for design, interpretation and application. This method proposes one such framework, discusses the suitability of AIS as a novel soft computing paradigm and reviews those works from the literature that integrate AIS with other approaches, focusing ANN, EA and FS.

Chandrasekaran, Asokan, Kumanan, Balamurugan and Nickolas [35] dealt with the criterion of makespan minimization for the job shop scheduling of different size problems. The computational method of artificial immune system algorithm they developed is used for finding optimal makespan values of different size problems. The artificial immune system algorithm is tested with number of benchmark problems. The results show that the AIS algorithm is an efficient and effective algorithm which gives better results than the Tabu search. In the research, the proposed AIS approach has been used for solving job shop scheduling problems with the objective of makespan minimization. The algorithm uses simple but effective techniques for calculating cloning process, applying mutations, and a receptor editing procedure.

Chandrasekharan and Ziegler [36-37] considered a problem of scheduling in permutation flowshops with the objective of minimizing the makespan, followed by the consideration of minimization of total flowtime of jobs. Two ant-colony optimization algorithms are proposed and analyzed for solving the permutation flowshop scheduling problem. The first algorithm called max–min ant system (MMAS) which is a newly proposed local search technique and the second ant-colony algorithm is newly developed and applied to number of benchmark problems. First, a comparison of the solutions yielded by the MMAS and the two ant-colony algorithms developed in the research, with the heuristic solutions is undertaken with respect to the minimization of makespan. The comparison shows that the two proposed ant-colony algorithms perform better, on an average, than the MMAS. Subsequently, by considering the objective of minimizing the total flowtime of jobs, a comparison of solutions yielded by the proposed ant-colony algorithms with the best heuristic solutions known for the benchmark problems.

2.7 Summary

Before finding the best possible assembly sequences, the generation of assembly sequences is important. The above mentioned literatures have been reviewed on generation of possible assembly sequences based on part design, assembly planning and sequence representation etc. Various old and recent methodologies have been studied to represent the sequences based on assembly structures in the form of nodes. Secondly, the numbers of constraints have been applied with the help of different literatures. The constraints are; precedence relationships, connectivity, cost of assembly and last but not the least stability criteria during assembly. The studied literatures give a comprehensive idea about the current trend of work in the subject. Lastly, the selection of paths for assembly is one of the critical factors, i.e, to select an optimal sequence by considering the mentioned constraints. To achieve that, the research takes the help of soft-computing techniques, i.e, ant colony optimization (ACO) and artificial immune system (AIS). The source of inspiration is taken from the metaheuristic methods to minimize search space explosion in the form of ant behavior and cloning processes, which is mentioned in the overview of literatures. The survey of literatures made in this chapter indicates that a lot of research remains to be done for improvement of the currently available techniques to combat the highly competitive future manufacturing systems.

CHAPTER 3

Generation of Assembly Sequences

3.1 Overview

The problem of sequencing has a primary role in the development of automated assembly and/or semi automated. In the last two decades this stimulating subject has been debated very frequently in the scientific world, and this fact is demonstrated by the very wide literature concerning to this topic. Several algorithms have been developed and tested to generate required sequences. When a product is assembled, an assembly agent will follow a prescribed order to put components into a fixture to complete the final assembly of the product. This order is known as assembly sequence of the product. In order to determine required sequences, many researchers used assembly constraints because the explicit acquisition of the assembly constraints has several merits. In some other methods, the users need to draw the graph of connections corresponding to the assembly and then answer a set of questions to each connection. The answers to these questions are expressed in the form of precedence relationships between logical combinations of connections. These precedence relationships between the connections are then used to generate the assembly sequences. Some authors have used matrices where as some other have used connectivity graph for modeling the product. The disassembly sequences are then generated following certain distinct rules. One common thread that appears in most of these works is the strategy of "assembly by disassembly" in which an assembly sequence is generated by starting with the complete product and working backwards through disassembly steps. The disassembly procedure is less complex than the assembly procedure. The backward assembly planning is based on one assumption that each part is rigid because the inverse of a disassembly sequence is equal to an assembly sequence only if each part is rigid.

This research work presents a methodology for the generation of robotic assembly sequences that is correct and complete. It is assumed that exactly two parts or subassemblies are joined at one time and after those parts is put together one by one to form a finish product. It is also assumed that whenever parts are joined and forming one subassembly, all contacts between the parts in that subassembly is established. Furthermore, it is assumed that the feasibility of joining two subassemblies is independent of how those subassemblies were built. These assumptions are consistent with the trend toward product designs that are suitable for automatic assembly. The correctness of the algorithm is based on the assumption that it is always possible to decide correctly whether two subassemblies can be joined, based on geometrical and physical criteria. This work presents an approach to compute this decision. The methodology described here operates on a relational model of an assembly. The relational model used in this work provides an efficient data structure that maintains contact geometry and connection information at the first level of representation and complete part geometry at a next level. This hierarchy permits many of the assembly decisions, such as local geometric feasibility, to be made by accessing only the highest level of the representation. In this sense, much of the assembly process is carried out in terms of an abstraction of the actual assembly parts description. Many different types of feasibility predicates could be incorporated into this overall structure. In this research, internal tasks are predicated which are related to constraints imposed by other factors, such as availability of resources, and stability predicated that assess the stability of the resulting subassemblies. No attempt is made to explore these criteria and constraints exhaustively here.

In this research, a product is considered to be suitable for robotic assembly when the following conditions are satisfied.

i) All the individual components are rigid;
ii) Assembly operation can be performed in all mutually perpendicular directions in space excepting +Z direction; and
iii) Each part can be assembled by simple insertion or screwing.

3.2 Product modeling for assembly sequence generation

The product modeling is a procedure to describe the assembled state in terms of connective relations between the component parts of a given product assembly. The connective relations are described in terms of the connective direction and the mating methods.

Considering a product consisting of *n* parts, the representation of the end product can be made in the following manner.

The product consisting *n* parts is represented in the format

$$A = (P, L) \tag{3.1}$$

where, A is a product having parts $P = \{p_\alpha \mid \alpha=1, 2 \, . \, . \, . \, . \, . \, n\}$, and interconnected by the liaisons $L = \{l_{\alpha\beta} \mid \alpha, \beta = 1, 2 \, . \, . \, . \, . \, . \, r. \; \alpha \neq \beta\}$.

Here *n* represents the number of parts of a product and *r* is the relationship between the connected parts and $(n-1) \leq r \leq n(n-1)/2$. The liaison $l_{\alpha\beta}$ represents the connective relationship between a pair of parts p_α and p_β. The connective relations can be either of a contact-type or of a fit-type. The representation of liaison $l_{\alpha\beta}$ is given by

$$l_{\alpha\beta} = \text{liaison} \quad (p_{\alpha,} C_{\alpha\beta}, f_{\alpha\beta}, p_\beta) \tag{3.2}$$

where, $C_{\alpha\beta}$ is the contact-type connection matrix and $f_{\alpha\beta}$ is fit-type connection matrix.

The dimension of each matrix is 2 × 3, and the matrices are represented by

$$C_{\alpha\beta} = \begin{pmatrix} Cx & Cy & Cz \\ C\bar{x} & C\bar{y} & C\bar{z} \end{pmatrix} \tag{3.3}$$

and

$$f_{\alpha\beta} = \begin{pmatrix} fx & fy & fz \\ f\bar{x} & f\bar{y} & f\bar{z} \end{pmatrix} \tag{3.4}$$

The assembly directions for robotic assembly are taken to be *d*, where, $d \in \{x, y, \bar{x}, \bar{y}, \bar{z}\}$. The representation of the elements of contact-type and fit-type are:

$$C_d = \begin{cases} 0{:}\, no\, contact\, in\, the\, d\, direction\, between\, p_\alpha \,\&\, p_\beta \\ rc{:}\, real\, contact\, in\, the\, d\, direction\, between\, p_\alpha \,\&\, p_\beta \\ vc{:}\, virtual\, contact\, in\, the\, d\, direction\, between\, p_\alpha \,\&\, p_\beta \end{cases} \tag{3.5}$$

and

$$f_d = \begin{cases} 0{:}\, no\ fit\ in\, the\, d\, direction\, between\ p_\alpha \,\&\, p_\beta \\ sw{:}\, screwing\ in\, the\, d\, direction\ between\ p_\alpha \,\&\, p_\beta \\ rf{:}\, round\ peg\ in\, hole\ fit\ in\, the\, d\, direction\, between\ p_\alpha \,\&\, p_\beta \\ mp{:}\, multiple\, round\ peg\ in\, hole\ fit\ in\, the\, d\, direction\, between\ p_\alpha\ and\ p_\beta \end{cases} \tag{3.6}$$

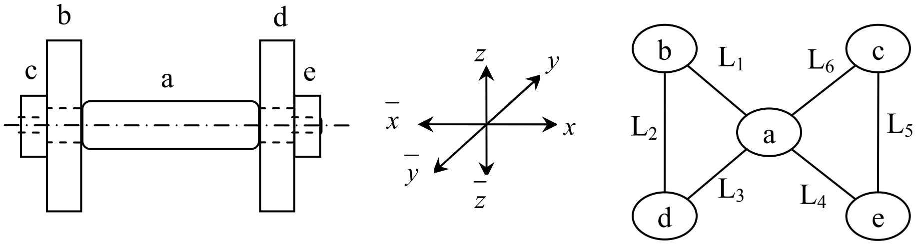
Each element of f_d can also be represented as round peg fit (*rf*), a polygon fit (*pf*), a tight fit (*tf*), a caulking (ca), a riveting (ri), a multi-peg-fit (*mp*), a virtual fit (*vf*) or no fit (0) .

The connection matrix in equation 3.2 is then represented in the following format.

$$l_{\alpha\beta} = liaison\left(p_{\alpha}, \begin{pmatrix} 0 & rc & rc \\ rc & rc & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & rf \\ 0 & 0 & 0 \end{pmatrix}, p_{\beta} \right) \tag{3.7}$$

Example Problem-1:

A grinder assembly as shown in Figure 3.1(a) is considered as an example problem for determination of the assembly sequence and validating the proposed method. Figure 3.1(b) shows the directions for assembly or disassembly operations whereas Figure 3.1(c) represents the liaison diagram of the individual component of the product. The table 3.1 shows the part description of the assembly product.



(a) (b) (c)

Figure 3.1(*a*): A simple example of a product (Grinder assembly); 3.1(*b*): Directions for assembly or disassembly; 3.1(*c)*: Liaison graph model of grinder.

Table 3.1: Part description of grinder assembly

Part Symbol	Part Name
a	Shaft
b	Blade
c	Nut
d	Blade
e	Nut

As per the codes of the model, the liaisons of the assembly components are represented as follows:

$$l_{ab} = liaison\left(a, \begin{pmatrix} o & rc & rc \\ rc & rc & rc \end{pmatrix}, \begin{pmatrix} o & o & o \\ rf & o & o \end{pmatrix}, b \right);$$

$$l_{ac} = liaison\left(a, \begin{pmatrix} o & o & o \\ vc & o & o \end{pmatrix}, \begin{pmatrix} o & o & o \\ sw & o & o \end{pmatrix}, c \right);$$

$$l_{ad} = liaison\left(a, \begin{pmatrix} rc & rc & rc \\ o & rc & rc \end{pmatrix}, \begin{pmatrix} rf & o & o \\ o & o & o \end{pmatrix}, d \right);$$

$$l_{ae} = liaison\left(a, \begin{pmatrix} vc & o & o \\ o & o & o \end{pmatrix}, \begin{pmatrix} sw & o & o \\ o & o & o \end{pmatrix}, e \right);$$

$$l_{bc} = liaison\left(b, \begin{pmatrix} o & o & o \\ rc & o & o \end{pmatrix}, \begin{pmatrix} o & o & o \\ o & o & o \end{pmatrix}, c \right);$$

$$l_{de} = liaison\left(d, \begin{pmatrix} rc & o & o \\ o & o & o \end{pmatrix}, \begin{pmatrix} o & o & o \\ o & o & o \end{pmatrix}, e \right)$$

3.3 Assembly constraints

The assembly constraints are divided into two categories: precedence constraints and connectivity constraints. A precedence constraint of a liaison $l_{\alpha\beta}$ is represented by a set of n_p parts that must be connected before two parts p_α and p_β are interconnected. The precedence constraint $PC(l_{\alpha\beta})$ of a liaison $l_{\alpha\beta}$ is expressed by

$$PC(l_{\alpha\beta}) = \{p_\gamma \mid \gamma = \gamma_1, \gamma_2, \ldots\ldots, \gamma_{n_p}\}, \tag{3.8}$$

and the precedence constraint $PC(p_f)$ of the part p_f is expressed by,

$$PC(p_f) = \bigcup_{l=l_1}^{l_q} P(l_{\alpha\beta}), \tag{3.9}$$

where $P(l_{\alpha\beta})$ is a precedence constraint of a liaison $l_{\alpha\beta}$ and is directly inferred from the part contact level graph.

While taking into consideration the overall structure of a product assembly, an assembly task can hardly be performed successfully because a previously assembled part p_j often prevents a part p_j from being assembled. In such a case p_k should be assembled before p_j. Such a p_k is called a precedence constraint to the assembly of p_j.

Carrying forward the same case study, the precedence constraints of the liaisons are:

$PC(l_{ab})=\{\Phi\}$; $PC(l_{ac})=\{b\}$; $PC(l_{ad})=\{\Phi\}$; $PC(l_{ae})=\{d\}$; $PC(l_{bc})=\{\Phi\}$ and
$PC(l_{de})=\{\Phi\}$ (3.10)

Accordingly, the precedence constraints of the parts can be listed as;

$PC(P_a)=\{b, d\}$; $PC(P_b)=\{\Phi\}$; $PC(P_c)=\{b\}$; $PC(P_d)=\{\Phi\}$ and
$PC(P_e)=\{d\}$ (3.11)

3.4 Part contact level graph

The part contact level graph represents the overall structure of a product assembly. In the graph, each node is denoted by a part, while each line represents directional connection between two parts. The assembly directions can be represented as $d \in \{x, y, z\}$ frame or $\overline{d} \in \{\overline{x}, \overline{y}, \overline{z}\}$ frame. The part contact level represents a precedence level with which a part can be assembled along a given direction when taking into consideration the contact type connections in that direction only. The level is denoted by $CL_d(p_i)$, where the subscript d indicates each direction $d \in \{x, y, z\}$ in the $\{x-y-z\}$ frame.

In the directional part contact level graph, $G_d = (P_d, L_d), d \in \{x, y, z\}$ is defined as follows:

The nodes P_d represent a set of all parts interconnected with contact type connections in a given direction d or $\overline{d}$, while the edge L_d denotes a set of all the fit and the contact type connections in those directions. Each node is arranged in the order of its contact level. Adjacent matrix (AM) includes the information on all the contact connections between parts interconnected in a particular direction.

The directional contact level graph can be obtained using the following steps:

Step 1: Generate an adjacent matrix AM_d for each direction $d \in \{x, y, z\}$;

Step 2: Calculate the contact levels from the adjacent matrix;

Step 3: Arrange nodes at their contact levels and connect them with edges.

The adjacent matrix represents adjacent relations between all possible pair p_f parts. If P_d is a set of g parts consisting of $p_1, p_2, p_3, \ldots.., p_n$ interconnected by liaisons, it is expressed as $g \times g$ matrix in the following form:

$$AM_d = \begin{array}{c|cccccc|} & p_1 & p_2 & \cdots & p_j & \cdots & p_g \\ \hline p_1 & m_{11} & m_{12} & \cdots & m_{1j} & \cdots & m_{1g} \\ p_2 & m_{21} & m_{22} & \cdots & m_{2j} & \cdots & m_{2g} \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots & \vdots \\ p_i & m_{i1} & m_{i1} & \cdots & m_{ij} & \cdots & m_{ig} \\ \vdots & \vdots & \vdots & \vdots\vdots\vdots & \vdots & \vdots\vdots\vdots & \vdots \\ p_g & m_{g1} & m_{g2} & \cdots & m_{gj} & \cdots & m_{gg} \\ \hline \end{array} \tag{3.12}$$

where

$$m_{ij} = \begin{cases} 1 \;\; if\; c_d \in \{rc, vc\} in\, C_{ij}\; or\, c_{\bar{d}} \in \{rc, vc\} in\, C_{ji} \\ 0 \;\; if\; c_d \vee c_{\bar{d}} = 0\; in\; C_{ij}\; or\, C_{ji} \\ -1 \;\; if\; C_d \in \{rc, vc\} in\, C_{ij}\; or\, c_{\bar{d}} \in \{rc, vc\} in\, C_{ji} \end{cases} \tag{3.13}$$

In the above c_d is an element of C_{ij} or C_{ji} in $d \in \{x, y, z\}$ and $c_{\bar{d}}$ is an element in $\bar{d} \in \{\bar{x}, \bar{y}, \bar{z}\}$. Each element m_{ij} is defined as follows; the value $m_{ij} = 1$ (assigned) when the part p_j in the column part makes a single contact with the part p_i in the row part from the above in the direction *d*. The value $m_{ij} = -1$ (assigned) when the column part p_j makes a single contact with the row part p_i from the below in the direction *d*.

Now the part contact level can be calculated by the following equation:

$$CL_d(p_j) = \begin{cases} \max\{m_{ij} + CL_d(p_i)\} & if\ m_{ij} = 1 \\ 1 & if\ m_{ij} \leq 0\ for\ \forall i \\ irrelevent & if\ m_{ij} = 0 \\ CL_d(p_i) & if\ i = j \end{cases} \tag{3.14}$$

where, $CL_d(p_i)$ and $CL_d(p_j)$ denote the contact level of the row part p_i and that of the column part p_j along a given direction *d*, respectively. The first equation determines the contact level of the column part p_j by taking the maximum value among the contact levels of the row parts p_i and adding their elements' value $m_{ij}=$ 1. The second one indicates that the column part p_j has the first contact level when the element $m_{ij} \leq 0$. The third one indicates that the contact level of the part p_j cannot be directly determined by that of the part p_i alone when the element $m_{ij}= 0$. The fourth one guarantees that any part has one contact level in a given direction regardless of a column or row part.

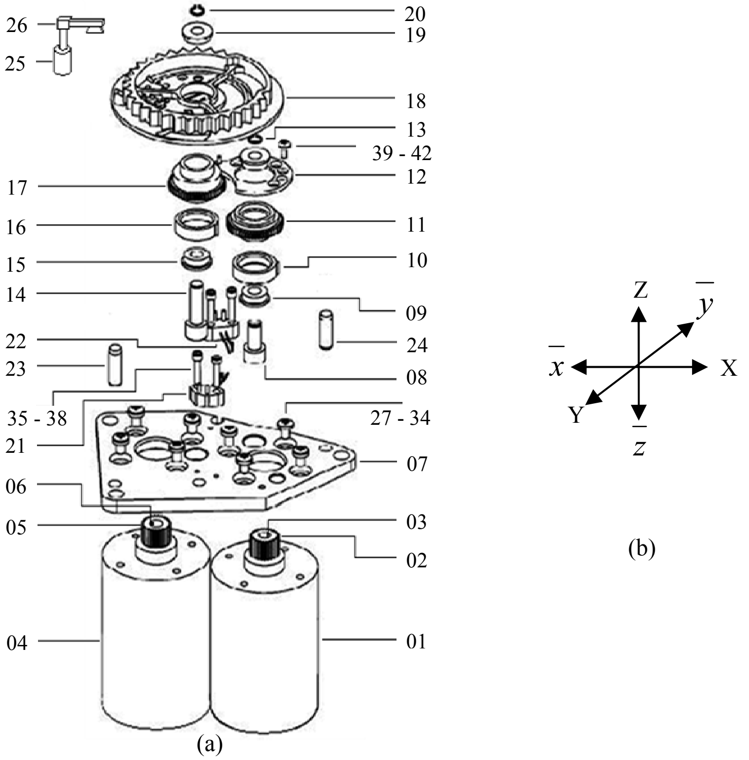
3.5 Directional precedence constraint

By utilizing the Part Contact Level Graphs (PCLG), the assembly precedence constraint for each directional connection is inferred. The directional connection between two adjacent parts within one contact level difference can be assembled freely at any time, whereas the connection between two parts with more than one contact level difference should be established after the parts existing at the intermediate contact levels are assembled. Therefore, the precedence constraint for each directional connection can be defined as parts and the level graphs. The precedence constraint for each directional connection corresponds to the nodes existing at the intermediate contact levels. The intermediate nodes of the edge can be inferred by finding the nodes or the paths with monotonically increasing contact level difference.

The assembly precedence constraint(s) for each directional connection is inferred using the part contact level graphs. The directional connection between two adjacent parts within one contact level difference can be assembled freely at any time, whereas the connection between two parts with more than one contact level difference should be established after the parts existing at the intermediate contact levels are assembled. Therefore, the precedence constraint for each directional connection can be defined as parts and the level graphs. The precedence constraint for each directional connection corresponds to the nodes existing at the intermediate contact levels. The intermediate nodes of the edge can be inferred by finding the nodes or the paths with monotonically increasing contact level difference.

Example Problem-2:

In order to work on the proposed method, another product (Discriminator) is considered as the next case study. The discriminator is a 42-part clockwork-like mechanism used as a safety device. Several parts overlapped in the CAD data, including 12 screws which were modeled larger than their corresponding holes, resulting in 18 model overrides. It contains 67 numbers of liaisons. However, only the feasible base parts (FBP) are considered. The exploded drawing of the discriminator and its part liaison diagram are presented in Figure 3.2(a) and Figure 3.3 respectively.



(a)

(b)

Figure 3.2(a): An exploded view of the discriminator; 3.2(b): The three mutually perpendicular axes for assembly direction

There are only six FBPs ($p_1, p_4, p_7, p_8, p_{14}$ and p_{14}). The liaisons of these FBPs are as follows:

$$l_{1-7} = \left(p_1, \begin{pmatrix} 0 & 0 & rc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & rf \\ 0 & 0 & 0 \end{pmatrix}, p_7 \right); \; l_{4-7} = \left(p_4, \begin{pmatrix} 0 & 0 & rc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & rf \\ 0 & 0 & 0 \end{pmatrix}, p_7 \right);$$

$$l_{7-8} = \left(p_7, \begin{pmatrix} 0 & 0 & rc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & rf \\ 0 & 0 & 0 \end{pmatrix}, p_8 \right); l_{7-14} = \left(p_7, \begin{pmatrix} 0 & 0 & rc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & rf \\ 0 & 0 & 0 \end{pmatrix}, p_{14} \right);$$

$$l_{14-18} = \left(p_{14}, \begin{pmatrix} 0 & 0 & rc \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & rf \\ 0 & 0 & 0 \end{pmatrix}, p_{18} \right)$$

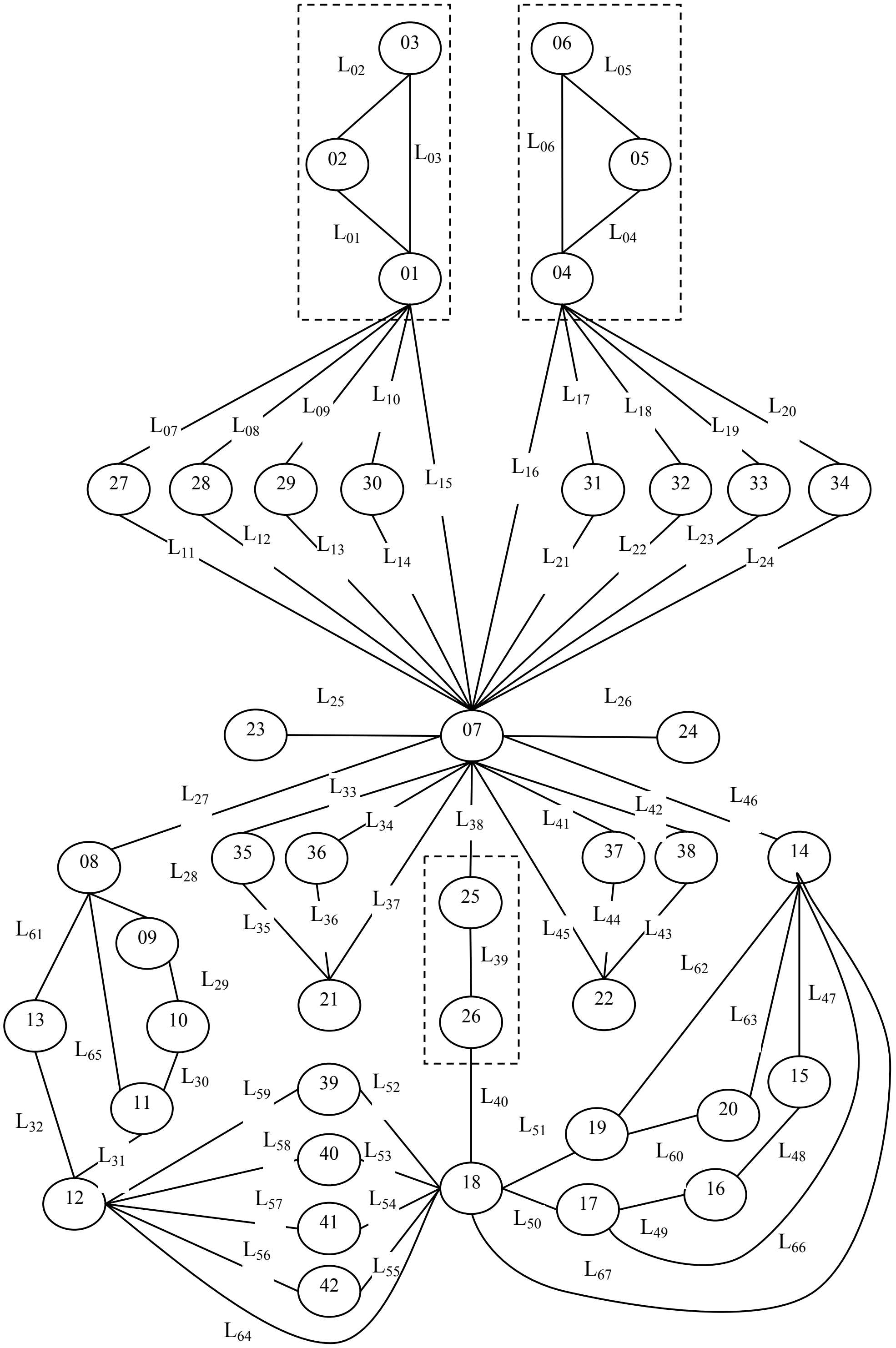


Figure 3.3: The liaison diagram of discriminator.

Considering the motion of the feasible base parts, it is found that none of the parts has motions in either x-direction or y-direction. Hence,

$$\mathrm{P}_x = \{\phi\}, \mathrm{P}_y = \{\phi\}, \mathrm{P}_z = \{p_1, p_4, p_7, p_8, p_{14}, p_{18}\}$$

The adjacent matrix AM_d for the product (Discriminator as shown in Figure 3.2(a)) is determined and is presented in Figure 3.4 as:

	P_1	P_4	P_7	P_8	P_{14}	P_{18}
P_1	0	0	1	0	0	0
P_4	0	0	1	0	0	0
P_7	-1	-1	0	1	1	0
P_8	0	0	-1	0	0	0
P_{14}	0	0	-1	0	0	0
P_{18}	0	0	0	0	-1	0

AM_z

Figure 3.4: Adjacent matrix AM_d for the Discriminator in the z-axis direction.

The contact level in z-direction is:

$$\left.\begin{aligned} &CL_z(p_1) = 1, CL_z(p_4) = 1 \\ &CL_z(p_7) = CL_z(p_4) + 1 = 2 \\ &CL_z(p_8) = CL_z(p_7) + 1 = 3 \\ &CL_z(p_{14}) = CL_z(p_7) + 1 = 3 \\ &CL_z(p_{18}) = CL_z(p_{14}) + 1 = 4 \end{aligned}\right\} \tag{3.15}$$

Figure 3.5 illustrates the directional part contact level graph for the base product of discriminator as shown in Figure 3.2(a). In this study the CL_x, CL_y, CL_z denotes the directional contact levels designated for three orthogonal directions of x, y, z respectively. In this particular case the CL_x and CL_y doesn't exist. Hence, the graph shows only in z-direction. The figure indicates the fact that p_1 has a compound connection consisting of a rc and a rf with p_7 in z-direction. The graph visualizes the overall structure of the product discriminator in level by level form, i.e. it gives the little indication towards the assembly sequence that which part is connected first or second or at any level.

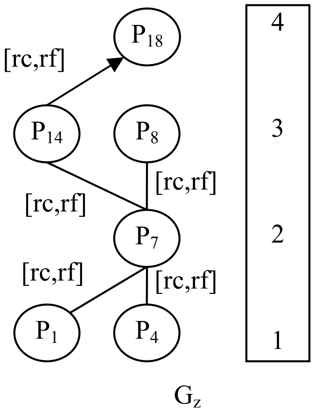


Figure 3.5: Part contact level graph of discriminator.

By utilizing the part assembly precedence constraints thus obtained, the feasible assembly sequence of the base parts is obtained as

$$(p_1, p_4) \rightarrow p_7 \rightarrow (p_8, p_{14}) \rightarrow p_{18} \tag{3.16}$$

The part assembly precedence constraint, therefore, is a key factor for inferring the assembly sequence. This inference procedure becomes more systematic and simple, when compared to that of conventional geometrical reasoning or query/answer methods.

3.6 Stable assembly sequences

Once assembly constraints have been inferred, assembly sequences satisfying the assembly constraints can be generated. Such assembly sequences are called the feasible assembly sequences. The feasible assembly sequences, however, do not always guarantee the parts to fix onto an in-process subassembly, parts may be loosely connected, and come apart when the subassembly is turned or moved. Such assembly sequences that keep the stability of in-process subassembly movement are called the stable sequences, by which the parts can be successfully assembled to form an end product.

3.6.1 Part instability

In this study the part stability is considered by evaluating the instability conditions. More is the instability, more is the weighted factor or vice versa. Here the part instability conditions can be inferred as 2×6 matrixes. That is, 2×3 each for translational motion and rotational motion. Figure 3.6 shows the direction of possible motions of the parts in 12 half degrees of freedoms.

$$S(p_k) = [T(p_k) : R(p_k)],$$

where, $T(p_k) = \begin{bmatrix} t_x & t_y & t_z \\ t_{\bar{x}} & t_{\bar{y}} & t_{\bar{z}} \end{bmatrix}$ is a translational instability matrix

and

$R(p_k) = \begin{bmatrix} r_x & r_y & r_z \\ r_{\bar{x}} & r_{\bar{y}} & r_{\bar{z}} \end{bmatrix}$ is a rotational instability matrix.

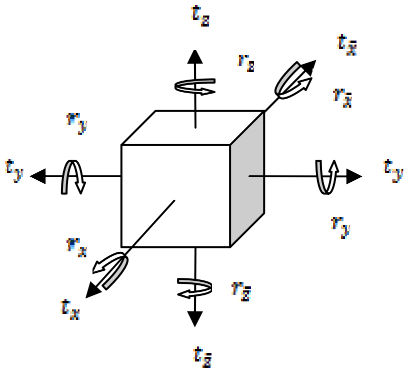


Figure 3.6: Motion of a part in free space

The element of $T(p_k), t_\delta, \delta \in \{x, y, z, \bar{x}, \bar{y}, \bar{z}\}$ is defined by:

$$t_\delta = \begin{cases} 1 \ \textit{if translated freely in the direction} \ \delta \\ 0 \ \textit{if blocked in the direction} \ \delta \end{cases}$$

while the element of $R(p_k), r_\delta, \delta \in \{x, y, z, \bar{x}, \bar{y}, \bar{z}\}$ is defined by:

$$r_\delta = \begin{cases} 1 \ \textit{if rotated freely in the direction} \ \delta \\ 0 \ \textit{if blocked around the direction} \ \delta \end{cases}$$

$$S(p_k) = \begin{bmatrix} 1 & 1 & 1 & \vdots & 1 & 1 & 1 \\ 1 & 1 & 1 & & 1 & 1 & 1 \end{bmatrix}$$

This means that the part is free to move in any direction in space. It has twelve half degrees of freedom. When the part is assembled with another part by a liaison, some of the motions in *d* directions of that part have been restricted. So the degree of motion instability has been reduced.

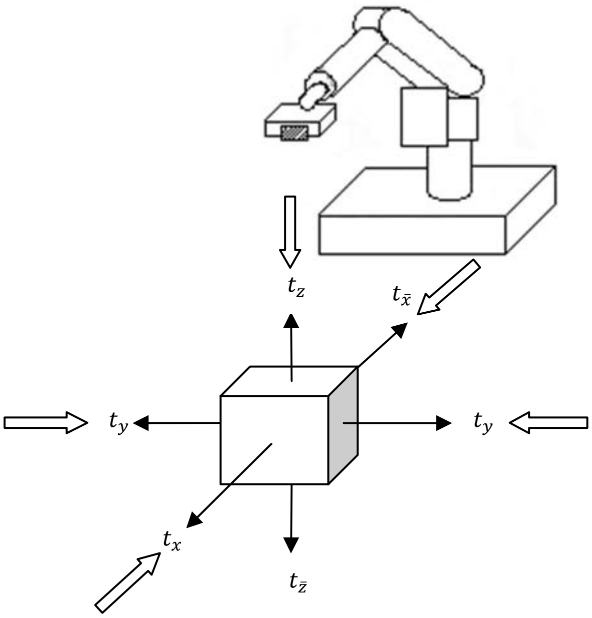


Figure 3.7: The directions of assembly by a robot

When a part is assembled with another part by a liaison which may have some of six directional contact or fit-type connections, the liaison instability can be obtained by following the procedure:

Step 1: Allocate a specific instability matrix for each directional connection established by the liaison.

Step 2: Obtained a liaison instability matrix by AND operating instability matrices of directional connections of the liaison.

Step 3: Modify the liaison instability matrix by OR operating with the part instability matrix of its base part if it is unstable.

Rule 1: When p_k is assembled to p_j by l_{jk}, a liaison instability of p_k with respect to the fixed part $p_j, S\{p_k(l_{jk})\}$ is obtained by AND operating all the instability matrices of directional connections established by the liaison

$$S\{p_k(l_{jk})\} = \left[\bigwedge_{\delta} S\{p_k(l_{jk}(c_\delta))\}\right] \wedge \left[\bigwedge_{\delta} S\{p_k(l_{jk}(f_\delta))\}\right] \qquad (3.17)$$

for all $\delta \in \{x, y, z, \bar{x}, \bar{y}, \bar{z}\}$, where, $S\{p_k(l_{jk}(c_\delta))\}$ is an instability matrix for p_k connected with a directional contact connection c_δ, and $S\{p_k(l_{jk}(f_\delta))\}$ indicates that for a directional fit connection f_δ.

Rule 2: When a part is simultaneously assembled with more than one part of a base part, the part instability is obtained depending upon stability of the base part movement.

Case 1: fixed base parts. If p_k is assembled with a set of fixed base parts, $PF = \{p_f \mid f = 1, 2, \cdots, m\}$, then $S\{p_k\}$ can be obtained by AND operating all the liaison instabilities, $S\{p_k(l_{fk})\}$, $(f = 1, \ldots, m)$

$$S\{p_k\} = \bigwedge_f S\{p_k(l_{fk})\}, \ (f = 1, 2, \ldots, m) \qquad (3.18)$$

Case 2: unstable base parts. If p_k is assembled with a set of unstable base parts, $PU = \{p_u \mid u = m+1, \cdots, q\}$, then $S\{p_k\}$ can be obtained by OR operating.

$$S\{p_k\} = \bigwedge_u [S\{p_k(l_{uk})\} \vee S(p_u)], \ (u = m+1, \ldots, q). \qquad (3.19)$$

Case 3: combination of fixed and unstable base parts. If p_k is assembled with both $PF = \{p_f \mid f = 1, 2, \cdots, m\}$ and $PU = \{p_u \mid u = m+1, \cdots, q\}$, then $S\{p_k\}$ can be obtained by AND operating.

$$S\{p_k\} = \bigwedge_f S\{p_k(l_{fk}\} \wedge \left[\bigwedge_u [S\{p_k(l_{uk}\} \vee S(p_u)] \right] ; \qquad (3.20)$$

$(f = 1, 2, \cdots, m), (u = m+1, \cdots, q)$.

3.7 Summary

A systematic method of generating correct assembly sequence for robotic assembly has been proposed. The method is based upon the evaluation of base assembly motion instability along with part contact level graphs that infers the precedence constraints. The modeling procedure takes in to consideration topological relationship between parts constituting the product assembly. The assembly sequence thus obtained is called a stable sequence. The basic concept is based upon the observation that, when a part should be assembled in such direction as a robot hand cannot access, base assembly should be rotated in a new posture, thus enabling the part to be assembled. In this situation, the base assembly should be kept firmly to prevent any of its parts from being disassembled.

To formulate a scheme for generating such a stable sequence, instability of base assembly movement has been defined utilizing the instability of each individual part motion. This facilitates the determination of the stable sequences. Among these assembly sequences, one can find the most desirable sequence which yields the minimum number of direction changes in base assembly movement. This is due to the fact that this sequence requires a simple design of fixtures and the minimal dexterity of robots. This procedure has been applied to a discriminator. It is claimed that the stable assembly sequences for robotic assembly can be efficiently inferred by utilizing the proposed inference method. The development of a procedure to cluster parts into subassemblies to obtain a hierarchical model of the assembly and the development of good heuristics to guide the generation of assembly sequences are issues for future research.

CHAPTER 4

Assembly Sequence and Soft Computing Methods

4.1 Overview

A variety of optimization tools are available for application to the problem, but their suitability and/or effectiveness are also under scanner. It is difficult to model the present problem as an *n-p* problem. Finding the best sequence generation involves the conventional or soft-computing methods by following the procedures of search algorithms. The problem of any search algorithm in general is guiding through the search process such that intensification and diversification are achieved to meet the global optima. Intensification is an expression commonly used for the concentration of search process on areas in the search space with good quality solutions. Diversification denotes the action of leaving already exploded areas and moving the search process to unexplored areas. Metaheuristic is a set of algorithms concepts that can be used to define heuristic methods applicable to a wide set of different problems. That means metaheuristic touches to the search space regions containing high quality solutions. A combinatorial optimization problem is either maximization or a minimization problem which has associated with a set of problem instances. The term instances refer to a problem with specified values for all the parameters. Examples of such techniques are: Simulated Annealing, Tabu Search, Evolutionary Computation, Ant Colony Optimization, and Artificial Immune System.

Metaheuristics are very useful for large search space problems. Study of various optimization methods reveals that Ant Colony Optimization (ACO) and Artificial Immune System (AIS) technique can be advantageously used to solve such problems. Previously numbers of methods have been proposed on robotic assembly sequence generation, but all of them have the problem of search space explosions. Sometimes it may happen that local optimum is selected. To overcome such situation a method has been modified to generate optimal assembly sequence using ACO and AIS. Ant Colony Optimization (ACO) is a model-based metaheuristic approach for solving hard combinatorial optimization problems. The inspiring source of ACO is the foraging behaviors of the real ants which enables them to find shortest path between a food source and their nest. It is best suitable for combinatorial optimization problems. Artificial immune systems (AIS) can be defined as computational systems inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. AIS is expected to give rise to powerful and robust information processing capabilities for solving complex problem, as it diversifies the solutions keeping the assembly constraints in mind.

4.2 Ant algorithm concept

The basic idea of ant algorithm is to imitate the cooperative manner of an ant colony to solve combinatorial optimization problems within a reasonable amount of time. While building their path from nest to food source, ants can deposit and sniff a chemical substance called pheromone, which provides them with the ability to communicate with each other. An ant lays some pheromone on the ground to mark the path it follows by a trail of this substance. Ants essentially move at random, but when they encounter a pheromone trail, they decide whether or not to follow it. If they do so, they deposit their own pheromone on the path, which reinforces the trail. The probability that an ant chooses one path over others is determined by the amount of pheromone on the potential path of interest. With the continuous action of the colony, the shorter paths are more frequently visited and become more attractive for the subsequent ants. By contrast, the longer paths are less attractive because the pheromone trail will evaporate with the passing of time. Finally, the shortest way from the nest to the source of foods is found.

The main characteristics of ant algorithms are positive feedback, distributed computation, and the use of a constructive greedy heuristic search. Positive feedback accounts for rapid discovery of good solutions, distributed computation avoids premature convergence, and the greedy heuristic search helps find acceptable solutions in the early stages of the search process. The generic ant algorithms have four main steps as follows:

I. Initialization: Set initial population of the colony and the pheromone trail. Place starting nodes for all ants randomly.
II. Solution construction: Taking into account the problem-dependent heuristic information and the trail intensity of the path, each ant chooses the next node that has not been visited to move by probability. Repeat the step until a completed solution is constructed.

III. Trail update: Evaluate the solutions and deposit pheromone on the solution paths according to the quality of solutions. The better the solution, the greater amount of pheromone deposited.
IV. Pheromone evaporation: The pheromone trail of all paths is decreased by some constant factor at the end of an iteration of building completed solutions.

The ACO algorithms have been applied successfully in a variety of optimization problems that can be expressed as searching for optimal paths on graphs, such as the traveling salesman problem (TSP), just-in-time (JIT) sequencing, and job-shop scheduling.

The present research is based on the following assumptions;

I. The possible ant trails joining the nest and food are represented by the possible disassembly sequences of components that, inversely, represent the assembly sequences;
II. The nest is represented by the first component of the sequence, and the food by the last components;
III. The concept of trail length (to be minimized) is substituted by the concept of sequence quality (to be maximized), evaluated according to the number of product orientation changes.

4.3 Artificial immune concept

Genetic Algorithm (GA) is successful application as an optimizer in non trivial combinatorial problems. Even if GAs are not Function Optimizers they nevertheless share with them many features which make it difficult to decide whether they are innovative optimization techniques or genuinely adaptive artificial systems. Their success over traditional optimizers in a wide range of applications is seen as an evidence of their superior adaptiveness. But their efficiency as optimizers ends up contradicting some of the principles of natural evolution. In particular, GAs converge and therefore lose progressively diversity as the optimum spread over their population. Such a behavior is fundamental for optimizers as they are valued according to the certitude and quickness of their convergence but for artificial adaptive systems, this uniforming is completely antagonistic with the polymorphism observed in populations undergoing natural evolution. Moreover, convergence inhibits progressively the effects of crossover and reduces the GA's exploration dynamics to mutation only.

Just as Genetic Algorithm (GA), Artificial Immune System (AIS) is also population based and the optimal solution is obtained by the evolution of the population. In AIS, the problems to be solved are regarded as antigens, while antibodies composed of genes just like chromosome in GA represent the solutions. In general, three types of measurements are used to evaluate the antibodies, namely, fitness for the quality, affinity for similarity between antibodies, and concentration for population diversity. This focuses on employing the bionic principles of AIS to achieve global optimization with desirable efficiency. First, immune regulation is applied to select the next generation antibodies with the purpose of maintaining population diversity and increasing the opportunity of global optimization. Secondly, the clonal selection principle is employed to enhance the local search, which intensifies exploitation of known space. Thirdly, vaccines are developed based on experiential knowledge of the problem to be solved and inoculation is employed to improve the quality of the solution candidates to speed up convergence.

Artificial immune system (AIS) is a computational intelligence paradigm inspired by the biological immune system, which has found application in pattern recognition, scheduling, control, machine-learning and information systems security. AIS has been applied successfully to a variety of optimization problems and studies have shown that it possesses several attractive immune properties that allow evolutionary algorithms (EAs) to avoid premature convergence and improve local search. The research considers the development of an evolutionary immune algorithm to exploit the complementary features of EA and AIS. The algorithm incorporates the features of clonal selection and immune memory to improve evolutionary search by identifying potential regions to explore while avoiding over emphasis in any region of the search space.

The study, borrow the concepts of vaccine and inoculation. Because the vaccines are developed according to specific problems, they can give guidance to solutions. Essentially, inoculation is to partially adjust solution candidates with the vaccines to make the candidates approach the optimal solution. To implement the optimization ideas addressed previously, immune operations including immune selection, clonal selection and inoculation are introduced. AIS is realized by the following steps: (1) recognition of antigens; (2) generation of initial antibodies; (3) evaluation of antibodies, i.e. calculations of the fitness, the affinity and the concentration of the antibodies; (4) proliferation and suppression of antibodies, i.e. conducting the immune selection operation to proliferate high fitness level antibodies and suppress high concentration level ones; (5) generation of new antibodies, i.e. conducting the crossover and the clonal selection operation to generate the next generation antibodies; and (6) improvement of antibodies, i.e. partially adjusting solution candidates with vaccines to make the candidates approach the optimal solution. Steps 3–6 will be iterated until convergence criteria are satisfied. Basically, AIS is a kind of general optimization approach that can be applied to solve many problems.

The research work utilizes the AIS which introduce an immune selection operation to take into account the fitness/energy function, the concentration and the affinity of the antibody/stable sequences when choosing the individuals of the next generation. Simultaneously, maintenance of population diversity can be achieved, which helps to avoid premature convergence and increase the opportunity of global optimization. By introducing immune operation of inoculation in the form of stability conditions of the robotic assembly, the validity of solution candidates/sequences is improved. Therefore, the search for the optimal assembly sequence is accelerated and improvement in efficiency of the algorithm is achieved.

In AIS, the clonal selection operation is employed to enhance the local search by intensifying the exploitation of known space, which helps the algorithm converge rapidly. Although the use of genetic algorithm, ant colony optimization, simulated annealing and neural network techniques used by previous researchers to solve such problems are quite encouraging, AIS is found to be interesting and suitable for such kind of formulations and hence it has been chosen for being applied to obtain optimized assembly sequence with additional constraints such as precedence and connectivity constraints. The solution is obtained with a high speed through this optimization procedure, because the antibody that met with a specific antigen in the past is produced faster.

4.4 Approach for solution

The present work aims at determining an optimal sequence for robotic assembly by using the evolutionary technique of ant colony optimization. A correct sequence for assembly can be generated from the disassembly sequence. Disassembly is defined as the methodical extraction of valuable parts/subassemblies from a product through a series of operations. A disassembly sequence is represented as an ordered list of disassembly operations (DOs), DO = (n, d), where, n is the number of parts and d is the assembly directions. For six half degrees of freedom (hdof), each component has six disassembly nodes, viz. (n, x), (n, y), (n, z), $(n, \bar{x})$, $(n, \bar{y})$, and $(n, \bar{z})$. By observing the value of disassembly operations, the first node to be disassembled is selected. Starting from the first node, ants search the feasible disassembly sequences by travelling all the nodes. All these sequences form the solution space of the problem. Sequences which satisfy precedence constraints, connectivity constraints and energy equations are the solutions to the problem. In the beginning, the tours constructed by the ants are the initial feasible sequences for the problem. The problem is reduced to a great extent after passing through the condition of stability. Among them the least value of the sequence is selected. That means the solution that is generated to the problem is the stable and optimized one corresponding to the assembly constraints taken. The inverse of this one with directions is the optimal assembly sequence. Similar to the TSP in combinatorial optimization, this study generates an ant algorithm which considers the feasible and stable sequences and ultimately gives the optimal solution with respect to the objective function. The modified ant algorithm generates and optimizes assembly sequence corresponding to the energy matrix of the order of 5n × 5n. The search space is adequately low and drastically reduces of the order of 5 × 5 matrices in every next visit of the ants. The precedence constraints, between liaisons, and the corresponding energy matrix to connect two liaisons, were adopted to process the ant algorithm. In the ACO algorithms, a pheromone 'τ_{ij}' is used as the shared memory of all ants and simultaneously it considers the energy matrix. The pheromone 'τ_{ij}' is updated during the processing. Like the shortest path in TSP, this algorithm also gives the minimum energy path which has to be followed during disassembly.

The final goal of the search algorithm is therefore the detection of the assembly sequence having the maximum quality in terms of: minimum assembly costs, satisfying the assembly constraints, and minimum number of product reorientations. The fulfillment of these requirements leads toward an optimized sequence, allowing a reduction of the total assembly time and a reduction of the complexity and/or dexterity of robotic assembly line.

The objective of the present work is to generate feasible, stable and optimal robotic assembly sequence satisfying the assembly constraints with minimum assembly cost. The present research aims at evolving an approach for generating robotic assembly sequences using the evolutionary technique considering of the instability of assembly motions and/or directions.

4.5 Objective function of assembly sequence generation (ASG)

Energy function E_{seq} is associated with assembly sequence, and it represented as:

$$E_{\text{seq}} = E_{\text{J}} + E_{\text{P}} + E_{\text{C}} \tag{4.1}$$

where,

E_{seq} = Energy function associated with ASG
E_{J} = Energy related with Assembly cost
E_{P} = Energy related with Precedence constraints
E_{C} = Energy related with Connectivity constraints

$$J = \begin{cases} 1{:}\, \textit{if an assembly sequence violates} \\ \quad \textit{assembly constraints, or it is unstable} \\ \\ \rho_s C_{as} + \rho_t C_{nt} : \textit{otherwise} \end{cases} \tag{4.2}$$

where J is the assembly cost related to assembly sequences.

The energy associated with precedence constraints is $E_P = C_P \sum_{i=1}^{n} \mu_i$, where C_{P} is a positive constant and μ_i is the precedence index. The value is assigned to 0, if it satisfies the precedence constraints, otherwise it is 1.

Similarly, $E_C = C_C \sum_{i=1}^{n} \lambda_i$ is associated with connectivity constraints. In a similar manner connectivity index λ_i is inferred on the basis of liaison relationships.
The objective factor is

$$E_{seq} = C_J\, J + \sum_{i=1}^{n} (C_P \mu_i + C_C \lambda_i) \tag{4.3}$$

4.5.1 Degree of motion instability and number of assembly direction changes

The possible assembly directions can be inferred from the liaisons between the interrelated parts. The detailed procedure of inferring the assembly direction can be obtained [20]. The possible assembly direction sets $DS^{k}_{cdabe}s\,(k = c,b,a,d,e)$ for each part of a sequence are expressed by

$$\left.\begin{array}{l} p_c :\ DS^{c}_{cbade} = \{d^{c}_{j} \in D \mid j = 1,2,.....\} \\ p_b :\ DS^{b}_{cbade} = \{d^{b}_{j} \in D \mid j = 1,2,.....\} \\ p_a :\ DS^{a}_{cbade} = \{d^{a}_{j} \in D \mid j = 1,2,.....\} \\ p_d :\ DS^{d}_{cbade} = \{d^{d}_{j} \in D \mid j = 1,2,.....\} \\ p_e :\ DS^{e}_{cbade} = \{d^{e}_{j} \in D \mid j = 1,2,.....\} \end{array}\right\} \tag{4.4}$$

The ordered lists $DL^{cbade}_{i}\ (i = 1,2,.....,m)$, of possible assembly directions corresponding to the assembly sequence can be expressed by:

$$\left.\begin{array}{l} DL^{cbade}_{1} = \left\{d^{c}_{1}, d^{b}_{1}, d^{a}_{1}, d^{d}_{1}, d^{e}_{1}\right\} \\ DL^{cbade}_{2} = \left\{d^{c}_{2}, d^{b}_{2}, d^{a}_{2}, d^{d}_{2}, d^{e}_{2}\right\} \\ \qquad\qquad .... \\ \qquad\qquad .... \\ DL^{cbade}_{m} = \left\{d^{c}_{m}, d^{b}_{m}, d^{a}_{m}, d^{d}_{m}, d^{e}_{m}\right\} \end{array}\right\} \tag{4.5}$$

The hierarchical tree of the sequence assembly directions may be represented in the format given in the Figure 4.2 (a).

Based upon the above equation, the normalized degree of motion instability and the normalized number of assembly direction changes are evaluated. The formula for normalized motion instability C_{as} is:

$$C_{as} = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{12 \times i}\sum_{j=1}^{i}\left(S\{BA_j\}\right)_i\right\} \tag{4.6}$$

where, $BA_j\,(j=1,2,3,4,5)$ is the in-subassembly formed at the j^{th} assembly step, and $S\{BA_j\}$ means the degree of motion instability of the j^{th} subassembly. A zero degree of motion instability means the parts belonging to the subassembly are completely fixed to each other, whereas twelve degrees means the parts are free to move in any direction. Similarly, the normalized number of assembly direction changes C_{nt} can be expressed as:

$$C_{nt} = \frac{1}{m}\sum_{i=1}^{m}\left\{\frac{1}{i}\sum_{j=1}^{i}(NT_j)_i\right\} \tag{4.7}$$

where, (NT_j) is assigned to 1 if direction change of BA_j occurs for a DL_i^{cbade}, otherwise it is 0. If the sequence is unstable for all $(NT_j)_i s\ (j=1,2,3,4,5)$ are assigned to 1.So the number for all C_{as}s, and C_{nt}s lie between 0 and 1. The zero means, the sequence completely satisfies the constraints, and one means, unstable relationship. The disassembly of the part is shown in Figure 4.1. For a study, a possible sequence $seq = \{\text{c} - b - a - d - e\}$ is considered. The assembly directions of mating parts can be inferred from liaisons between the parts. In this study each set of possible assembly directions are represented in the format $DS_{cbade}^{k}\ \{k = a,b,c,d,e\}$ and these are:

$$DS_{cbade}^{c} = \{\phi\}, DS_{cbade}^{b} = \{\bar{x}\}, DS_{cbade}^{a} = \{\bar{x}, z, \bar{z}\}, DS_{cbade}^{d} = \{\bar{x}, z, \bar{z}\}, DS_{cbade}^{e} = \{\bar{x}, z, \bar{z}\}$$

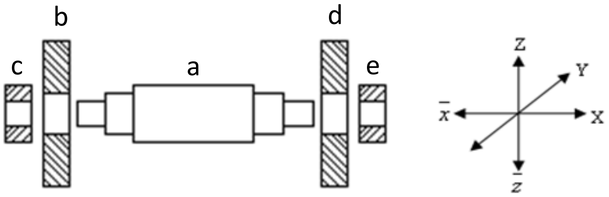
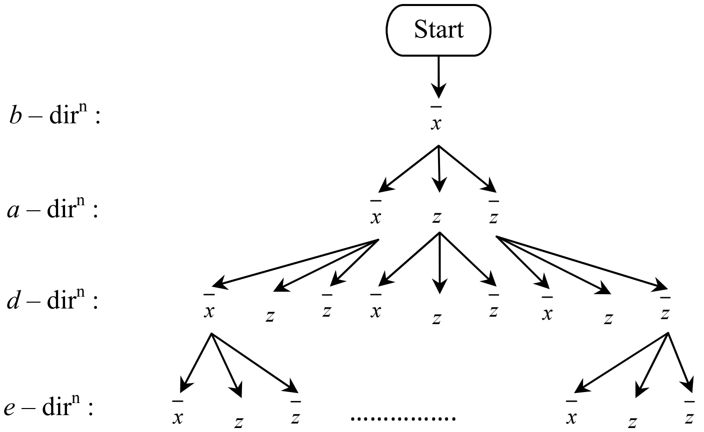


Figure 4.1: Parts of grinder

The order list of possible assembly directions corresponding to the assembly sequence "c – *b* – *a* – *d* – *e"* can be obtained.

The hierarchical structure of the assembly directions and the ordered list of possible assembly directions are shown in Figure 4.2(a) and (b) respectively.



(a)

$$\boxed{DL_1^{cbade}=\left\{\overline{x},\overline{x},\overline{x},\overline{x}\right\},DL_2^{cdade}=\left\{\overline{x},\overline{x},\overline{x},z\right\},\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots DL_m^{cbade}=\left\{\overline{x},\overline{z},\overline{z},\overline{z}\right\}}$$

(b)

Figure 4.2(a): Hierarchical tree structure of possible assembly directions. 4.2(*b*): Ordered list of possible assembly directions.

The ordered lists $DL_i^{cbade}\ (i=1,2,.....,m),$ given by the equation

$$\left.\begin{aligned} DL_1^{cbade} &= \left\{d_1^c, d_1^b, d_1^a, d_1^d, d_1^e\right\} \\ DL_2^{cbade} &= \left\{d_2^c, d_2^b, d_2^a, d_2^d, d_2^e\right\} \\ &\quad\ \ldots.. \\ &\quad\ \ldots.. \\ DL_m^{cbade} &= \left\{d_m^c, d_m^b, d_m^a, d_m^d, d_m^e\right\} \end{aligned}\right\} \tag{4.8}$$

The energy matrix is having the dimension of $5n \times 5n$ matrix. Each cell represent as the energy between two elements. In this study, each energy cell has been calculated and applied in ACO method.

4.6 Applying ACO to assembly sequence generation (ASG)

According to Marco Dorigo [24, 27 & 28] the basic concept of an ant colony algorithm is to solve combinatorial problems within a reasonable amount of time. Artificial ants iteratively tours through a loop that includes a tour construction biased by the artificial pheromone trails and the heuristic information. The main idea in modified algorithm is that the good tours are the positive feedback given through the pheromone update by the ants. The shorter is the tour the more amounts of pheromones deposits on the selected path. This means that the path have higher probability of being selected in the subsequent iterations of the algorithm. In this study, disassembly sequence is represented as disassembly operations (DO). The sequence considered the number of parts presented and the direction in which it is to be disassembled i.e. DO = (n, d), where ‘n’ is the number of components and ‘d’ is the direction of disassembly. In this paper, each component is having five possible DOs, i.e. (n, +x), (n, +y), (n, +z), (n, -x) and (n, -y). If the assembly consists of ‘n’ number of parts, then the disassembly operation is having ‘5n’ number of nodes. The disassembly operation is assigned to ‘1’ if there is interference in that direction, otherwise ‘0’. That means if DO=1, it cannot be disassembled from the product. In the modified ACO algorithms, a pheromone ‘ τ_{ij} ’ is used as the share memory of all ants and simultaneously it considers the energy matrix which is to be minimized. The pheromone ‘ τ_{ij} ’ is updated during the processing. Like the shortage path in TSP, this algorithm also gives the minimum energy path which is to be follow during disassembly. In this study the pheromone is expressed as 5n X 5n matrix as because one of the Z directions is restricted in study.

Interference matrix in (+)ve X, Y, Z directions:-

$$
DM = \begin{array}{c|cccc|}
 & e_1 & e_2 & & e_n \\
e_1 & I_{11x}I_{11y}I_{11z} & I_{12x}I_{12y}I_{12z} & \cdots & I_{1nx}I_{11y}I_{11z} \\
e_2 & I_{21x}I_{21y}I_{21z} & I_{22x}I_{22y}I_{22z} & \cdots & I_{2nx}I_{2ny}I_{2nz} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
e_n & I_{n1x}I_{n1y}I_{n1z} & I_{n2x}I_{n2y}I_{n2z} & \cdots & I_{nnx}I_{nny}I_{nnz}
\end{array} \tag{4.9}
$$

Where I_{ijd} is equal to 1 if component e_i interferes with the component e_j during the move along direction +*d*-axis; otherwise I_{ijd} is equal to 0. The initial disassembly matrix is calculated as:

$$
DM = \begin{array}{c|ccccc ccccc ccccc ccccc ccccc|}
 & \multicolumn{5}{c}{a} & \multicolumn{5}{c}{b} & \multicolumn{5}{c}{c} & \multicolumn{5}{c}{d} & \multicolumn{5}{c}{e} \\
 & x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} & x & y & z & \bar{x} & \bar{y} \\
a & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
b & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
c & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
d & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
e & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \tag{4.10}
$$

$$
DO_{i,(+d)} = \bigcup_{j=1}^{n} I_{ijd} \tag{4.11}
$$

$$
DO_{i,(-d)} = \bigcup_{j=1}^{n} I_{jid} \tag{4.12}
$$

$$
\begin{bmatrix}
DO_{a,x} = 1 & DO_{a,y} = 1 & DO_{a,z} = 1 & DO_{a,\bar{x}} = 1 & DO_{a,\bar{y}} = 1 \\
DO_{b,x} = 1 & DO_{b,y} = 1 & DO_{b,z} = 1 & DO_{b,\bar{x}} = 1 & DO_{b,\bar{y}} = 1 \\
DO_{c,x} = 1 & DO_{c,y} = 1 & DO_{c,z} = 1 & DO_{c,\bar{x}} = 0 & DO_{c,\bar{y}} = 1 \\
DO_{d,x} = 1 & DO_{d,y} = 1 & DO_{d,z} = 1 & DO_{d,\bar{x}} = 1 & DO_{d,\bar{y}} = 1 \\
DO_{e,x} = 0 & DO_{e,y} = 1 & DO_{e,z} = 1 & DO_{e,\bar{x}} = 1 & DO_{e,\bar{y}} = 1
\end{bmatrix} \tag{4.13}
$$

Here, U is the Boolean operator OR. The result will be equal to 0 if all the elements involving in the operation are 0. This means the element can be disassembled in that direction. If the DO is equal to 1, the element cannot be disassembled. In this study, the initial feasible disassembly operations are: (c, -x) and (e, +x).

4.6.1 Solution method

Robotic assembly is a case of combinatorial optimization problem. The problem is similar to Traveling salesman problem i.e. to give the shortest path with minimum cost. Combinatorial optimization problem is a triple (S, f, Ω), where S is the set of candidate solutions, f is the objective function which assigns an objective function value $f(s)$ to each candidate solution $s \in S$, and Ω is a set of constraints. The solutions belonging to the set of solutions S that satisfy the constraints Ω are called feasible solutions. In robotic assembly one term is added is called the stable solutions. The stable solutions $\tilde{\Omega} \subseteq \Omega$ belong to the feasible solutions. One of the major advantages is that, the optimal solution satisfies all the assembly constraints, objective function and also it is a part of stable solutions $\tilde{\Omega}$.

In ant system, m ants simultaneously build a solution of the ASG. Initially ants are put in first feasible DO. At each construction step, ant k applies a probabilistic state transition rule, called random proportional rule, to decide which node visit next.

$$P_d^{(i,j)} = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}{\sum\limits_{u \in C_d(i)} [\tau(i,j)]^{\alpha}[\eta(i,j)]^{\beta}}, \; if \; j \in C_d(i) \\ 0, \qquad otherwise \end{cases} \tag{4.14}$$

The heuristic value selected in this study is $\eta(i,j) = \dfrac{1}{E_{seq}}$.

After all the ants have constructed their tours, the pheromone trails are updated. The pheromone evaporation is giving by $\tau(i,j) \leftarrow (1-\rho)\tau(i,j)$, where $0 \leq \rho \leq 1$ is the pheromone evaporation rate. After evaporation, all ants deposit pheromone on the arcs they have crossed in their tour:

$$\tau(i,j) \leftarrow (1-\rho)\tau(i,j) + \sum_{k=1}^{m} \Delta\tau_k(i,j) \tag{4.15}$$

Where, *m* is the number of ants that find the iteration-best sequences and $\Delta\tau^k(i,j))$ is the amount of pheromone ant *k* deposits on the arcs it has visited. It is given an equation:

$$\Delta\tau^k(i,j) = \begin{cases} \dfrac{1}{E_{seq}^k(i,j)}, if\ (i,j) \in sequence\, of\ ant\, k \\ 0, \qquad otherwise \end{cases} \tag{4.16}$$

Where, $E_{seq}^k(i,j)$ is the tour energy the *k*th ant belonging to that tour. During the construction of sequences, local pheromone updating encourages exploration of alternative solutions, while global pheromone updating encourages exploitation of the most promising solutions.

4.6.2 ACO Algorithm

```
1. Generate the initial feasible Dos and compute their quantity
2. Set the cycle counter NC = 1
3. While NC < $NC_{max}$
    a. Place ants on the initial feasible disassembly node
    b. While each ant has not completed its tour
        i. Put current DO into sequence of the ant
        ii. Generate candidate list of the ant and calculate the energy
        iii. Calculate $p_k(i, j)$ of each candidate
        iv. Choose next DO $j$ based on energy matrix
        v. Move the ant to DO $j$
        vi. Add the component number of DO $j$ to the tabu list of the ant
        vii. Locally update PM
    c. Evaluate all solutions taking into account their reorientations
    d. Globally update PM using iteration-best solutions
    e. Update the best sequence of each ant if its iteration sequence is the best one found so far
    f. Empty the sequence, candidate list, and tabu list of each ant
    g. Set $NC = NC + 1$
4. Output the reversed best sequence of each ant
```

The reverse of the output is the optimal assembly sequence generation with inverse directions. The solution may be optimal or near optimal.

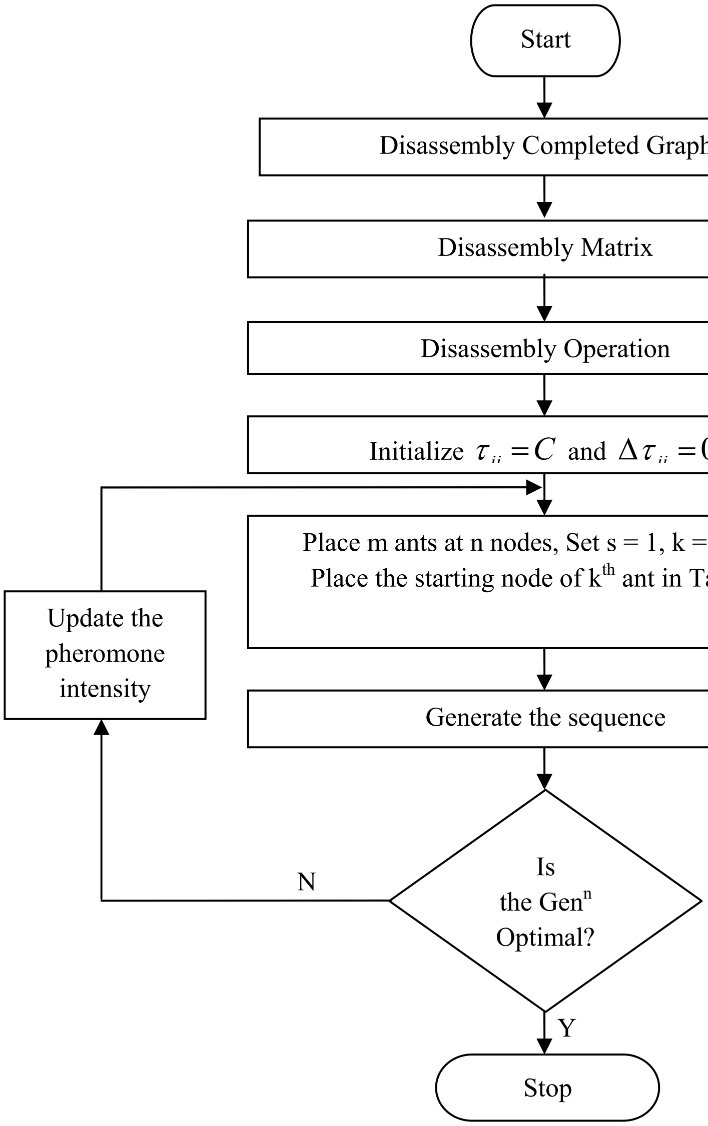


Figure 4.3: Flow diagram of the ACO procedure

4.7 Applying AIS to ASG

Initial antibodies i.e. the stable assembly sequences represent initial solutions and influence the quality of the final solution as well as the convergence speed to a certain extent. The modified algorithm generates the initial antibodies in the form of stable assembly sequences. For the case of generating optimal stable assembly sequences, the artificial immune system was built on the two principles.

a) Clonal selection principle
b) Affinity maturation principle

Clonal principle is the procedure in which affinities of antibodies become mature. Each stable assembly sequence has an energy value that refers to the affinity value of that antibody. Affinity and concentration are the measurements of similarity between antibodies and population diversity, respectively. The affinity is calculated through informative entropy and concentration can be obtained through affinity. The affinity of each stable sequence is calculated from affinity function and is defined by the equation

$$Affinity\left(p\right)= 1/E_{Seq} \tag{4.17}$$

where, E_{Seq} is the energy value of an individual assembly and p is the antibody.

From the equation, the lower is the energy value gives out the higher affinity value. The cloning of the antibodies is dependent on the affinity values. For a stable sequence, the energy related to assembly sequence generation is lower as compare to the other sequences. This is due to the fact that, the cost will rise and stability does matter for unnecessary handling of the parts of the product. The numbers of handling directions of the robots are directly linked to the energy function. The affinity function is the inverse of the energy sequence value. The more is the affinity value the less cost of the assembly sequence and also more stable it is. The cloning of antibodies is done directly proportional to the affinity function. More clones are generated on higher affinity values or lower energy values.

The affinity maturation principle is defined by two methods.

a) Mutation
b) Receptor editing

For the sake of diversification and getting the lower energy value of the sequence, a two phase mutation has been taken and they are:

a) Inverse mutation
b) Pairwise interchange mutation

In the first one, randomly two positions were selected and then inversed. After inversing if the mutated sequence affinity is greater than the original one, then the mutated one is stored. Otherwise, it will go to the pairwise mutation. Here randomly two positions were interchanged. If the affinity value is more than the original one, then it stores the new one. Otherwise, it stores the original one.

In receptor editing phase, worst percentage of the antibodies were eliminated and randomly created antibodies were replaced with them which satisfies the constraints criteria. This mechanism corresponds to new search regions in the total search space.

4.7.1 AIS Algorithm

```
1.  Generate a population of P antibodies (assembly sequences).
For each iteration:
    a.  Select the sequence in the antibody population
    b.  Find out the affinity of each antibody based on energy factor
    c.  Cloning process (generate copies of the antibodies)
2.  Steps in mutation process
For each clone:
    a.  Find inverse mutation (generate a new sequence)
    b.  Select the new sequence obtained from inverse mutation
    c.  Find the energy factor of the new sequence
    d.  If energy factor (new sequence) < energy factor (clone) then
                Clone = new sequence
        Else, do pairwise interchange mutation (generate a new sequence)
          i.   Select the new sequence
          ii.  Find the energy factor of the new sequence
          iii. If energy (new sequence) < energy (clone) then
                Clone =new sequence
               Else, clone = clone, antibody = clone
3.  Eliminate worst number of antibodies in the population
    a.  Create new antibodies at the same number
    b.  Change the eliminated ones with the new created ones
                while stopping criteria = false
```

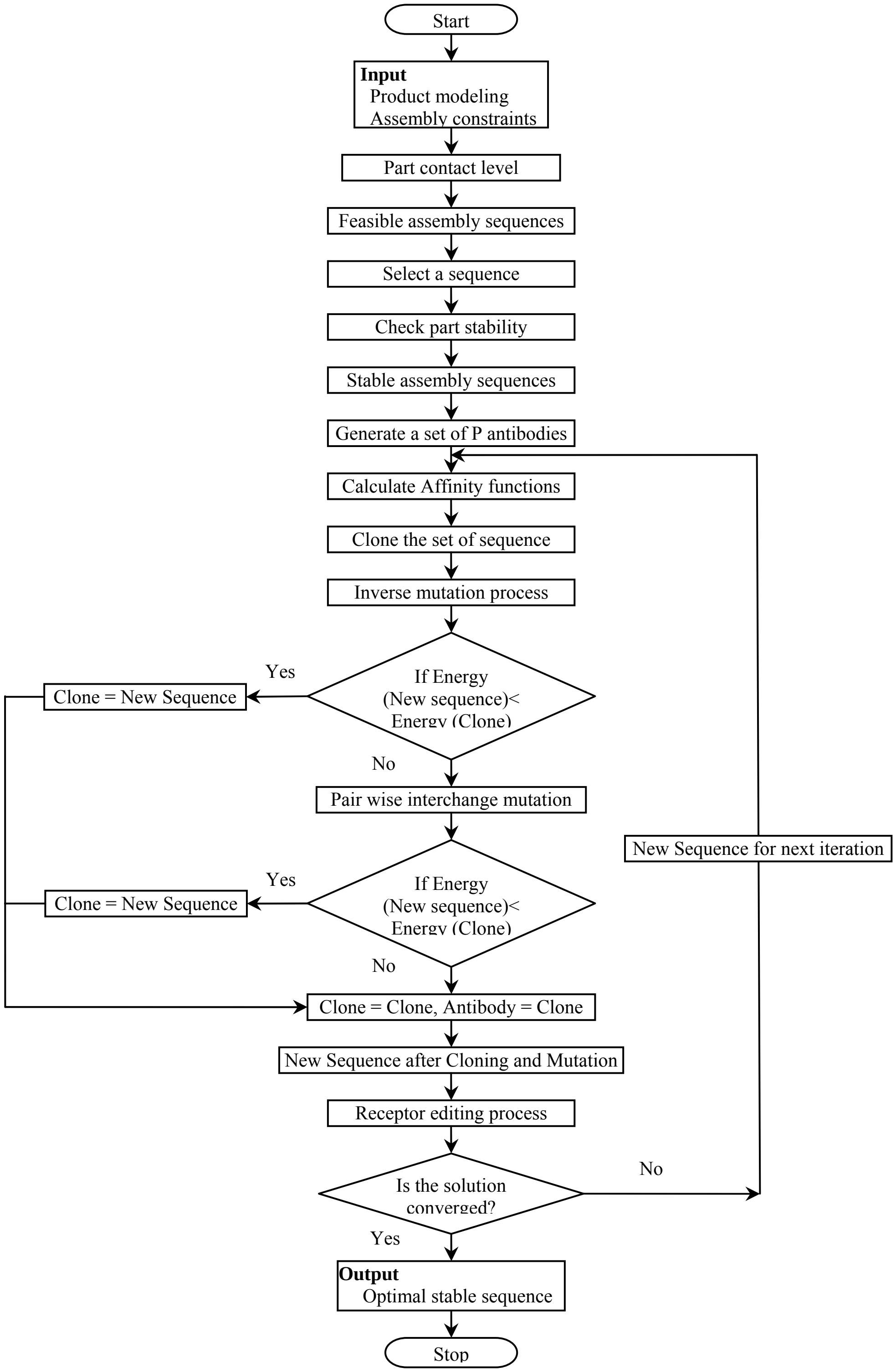


Figure 4.4: The flow diagram of the immune algorithm

4.8 Summary

The consideration of part motion instability considered in Hong and Cho [20] is crucial to the determination of assembly sequence since this affects the complexity and dexterity of robots. The method presented in the work generates stable assembly sequences by the use of the precedence constraints using the part contact level graph. The work carried out in this chapter is more transparent, i.e; the soft computing technique itself generate the stable assembly sequences in application with number of constraints and optimize the stables sequences to give out the best possible result as needed in robotic assembly sequence.

A robotic assembly sequence is considered to be optimal when it minimizes assembly cost while satisfying the process constraints. The assembly cost relates to assembly operations, assembly motions and assembly direction changes. The work utilizes an ant colony optimization (ACO) method for generation of robotic assembly sequences. The method relates the assembly cost to an energy function associated with the assembly sequence. The energy function is iteratively minimized by ACO. As a result, an assembly sequence with a minimum assembly cost is finally generated. Example problems are presented to show the effectiveness of the method. This modified method will be more preferable in robotic assembly sequence generation that considers the assembly constraints and the assembly costs. The proposed method gives indication about the interrelationship between the mating parts in every possible direction of assembly.

In the second section of this chapter, a new modified optimization algorithm is presented to generate optimal stable assembly sequence based on the clonal selection principle. The present algorithm maintains population diversity by the immune operations of immune selection and immune metabolism, and employs a clonal selection operation to enhance the local search. Furthermore, the immune operation of inoculation is incorporated to improve the quality of solution candidates to speed up convergence. The approach is able to produce results similar or better than those generated by other algorithms that are representative of the state-of-the-art in evolutionary objective optimization. This approach uses an affinity measure to control the amount of mutation to be applied to the antibodies. Affinity here is defined in terms of non-dominance and feasibility. These affinity measures, combined with the population are used to distribute non-dominated solutions in a uniform way.

CHAPTER 5

Results and Discussions

5.1 Overview

The results obtained by using various methods for the products under consideration are presented in the following sections. Essentially three different types of approaches are adopted in generating the optimize assembly sequences. In the first kind of approach conventional techniques using part contact level graph (PCLG) and part instability methods along with directional precedence constraints that are suitable and convenient for robotic assembly sequence generations have been considered. The second approach uses one of the effective metaheuristic optimization tool, viz. ant colony optimization (ACO), to solve the robotic assembly sequence generation. The third approach uses artificial immune systems (AIS) which is a computational systems inspired by the theoretical immunology. The immune functions, principles and mechanisms of the methodology are used to solve the problem of robotic assembly. The following sections present the results obtained through all these methods and the related discussions and comparisons.

5.2 Conventional robotic assembly sequence generation

The results of assembly sequence generation method as described in chapter 3 can be represented as an ordered list of assembly states or an ordered list of assembly tasks. The directed graph representation of assembly sequences represents all the feasible assembly sequences in the form of ordered lists of assembly states. Every state in the sequence is a feasible state. Every directed part in the directed sequence is an assembly task. An ordered list of assembly tasks is said to be a base assembly sequence of an assembly with respect to the representation of the assembly sequences if the sequence starts from the starting state and every assembly task in the list corresponds to one edge in the process. Since all the parts in the ordered sequence stand for feasible assembly states, no part would point to an infeasible state, i.e. no base assembly sequence with respect to the ordered sequence will lead to an infeasible state. Hence, the ordered sequence representation of feasible assembly sequences has the real time property. A set of precedence relationships that have the real time property may be generated from all the infeasible states by writing a precedence relation for each infeasible state. The algorithm to generate all the infeasible states is presented. The proposed methodology when applied to the discriminator shown in Figure 3.2(a) of the product assembly, gives out the results in the following manner.

The base assembly sequence, as mentioned in equation 3.16 is found out to be

$$(p_1, p_4) \rightarrow p_7 \rightarrow (p_8, p_{14}) \rightarrow p_{18}$$

This provides the backbone for further assembling the auxiliary and other parts to complete the product. As mentioned earlier the precedence constraints of the parts are derived from the part contact level graphs (PCLG) using their topological relationships. This is essentially obtained from the product and part data. The complete sequence of the assembly as obtained by the process is presented below in Figure 5.1.

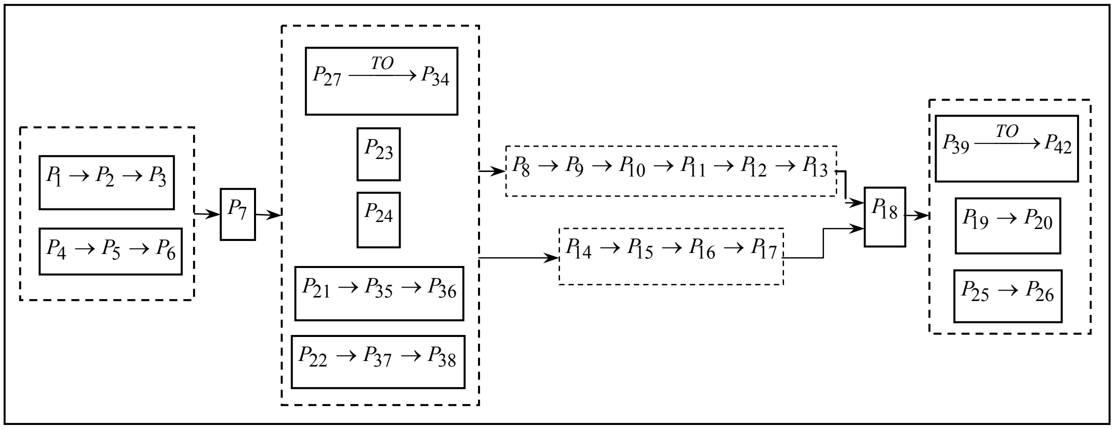


Figure 5.1: The stable assembly sequence of the parts of discriminator after going through instability

While taking the example of second product assembly shown in Figure 3.1(a), the grinder assembly is considered as a next case study for inference of the feasible assembly sequences by using the procedure of part contact level graph (PCLG). After applying the PCLG, the following results are found out. The results obtained through this method are presented in Table 5.1.

Table 5.1: Feasible assembly sequence

Seq No.	*Sequence Order*				
1	*a*	*b*	*c*	*d*	*e*
2	*b*	*a*	*c*	*d*	*e*
3	*d*	*a*	*e*	*b*	*c*
4	*a*	*b*	*d*	*c*	*e*
5	*a*	*d*	*b*	*e*	*c*
6	*d*	*a*	*b*	*c*	*e*
7	*e*	*d*	*a*	*b*	*c*
8	*b*	*a*	*d*	*c*	*e*
9	*a*	*b*	*d*	*e*	*c*
10	*d*	*a*	*b*	*e*	*c*
11	*c*	*b*	*a*	*d*	*e*
12	*a*	*d*	*b*	*e*	*c*
13	*b*	*a*	*d*	*e*	*c*

Using the part contact level graph (PCLG) and the parts stability criteria to the generated feasible assembly sequences from the motion instability, the number of sequences reduced to 3 as shown in Table 5.2.

Table 5.2: Stable assembly sequence

Seq No.	*Sequence Order*				
1	*a*	*b*	*c*	*d*	*e*
2	*b*	*a*	*c*	*d*	*e*
3	*d*	*a*	*e*	*b*	*c*

5.2.1 Discussion

A systematic method of generating correct assembly sequence for robotic assembly has been proposed. The method is based upon the evaluation of base assembly motion instability along with part contact level graphs that infers the precedence constraints. The modeling procedure takes in to consideration topological relationship between parts constituting the product assembly. The assembly sequence thus obtained is called a stable sequence. The basic concept is based upon the observation that, when a part should be assembled in such direction as a robot hand cannot access, base assembly should be rotated in a new posture, thus enabling the part to be assembled. In this situation, the base assembly should be kept firmly to prevent any of its parts from being disassembled.

To formulate a scheme for generating such a stable sequence, instability of base assembly movement has been defined utilizing the instability of each individual part motion. This facilitates the determination of the stable sequences. Among these assembly sequences, one can find the most desirable sequence which yields the minimum number of direction changes in base assembly movement. This is due to the fact that this sequence requires a simple design of fixtures and the minimal dexterity of robots. This procedure has been applied to a discriminator. It is claimed that the stable assembly sequences for robotic assembly can be efficiently inferred by utilizing the proposed inference method. The development of a procedure to cluster parts into subassemblies to obtain a hierarchical model of the assembly and the development of good heuristics to guide the generation of assembly sequences are issues for future research.

While looking the assembly point of view, no doubt the proposed method gives out the better results, but to quantify the obtained result is very difficult. To elaborate the effectiveness of the method we have applied the two soft-computing methods to this kind of problem to generate the optimized sequence(s).

5.3 Ant colony optimization algorithm applied to robotic assembly

As per the discussion in the above section, we take the example of grinder assembly to strengthen the methodology. The simulation results of selecting the parameters are shown in the Figure 5.2 and discussed below section.

5.3.1 Simulation for C_J, C_P, C_C

Let the product assembly is composed of *n* parts and is having 5*n* different nodes as disassembly considered in 5 directions except the restricted motion of robot in $\bar{z}$ direction. That means, for each part there are 5 different assembly directions. According that concept, the objective matrix now becomes a size of 5n X 5n nodes. Each node is having a weighted value in connection with other nodes. If the proceeding of assembly is approaching towards instability, the value of weighted factor is rising on and vice versa. This means that the lowest value selected through iteration means the more is the stable assembly. The assembly constraints, C_P, C_C and C_J mentioned in equation 4.3 are assigned random values from 5 to 75 with the increment of +5. These have been chosen in such a way that the convergence of the objective function is more as shown in Figure 5.2. It is assumed that the selected factors behave in a similar manner throughout the tour. The simulation results are listed in Table 5.3 and Table 5.4, and the simulation results are shown. The energy constraints given in Table 5.3 are determined during the simulation by observing the convergence tendency.

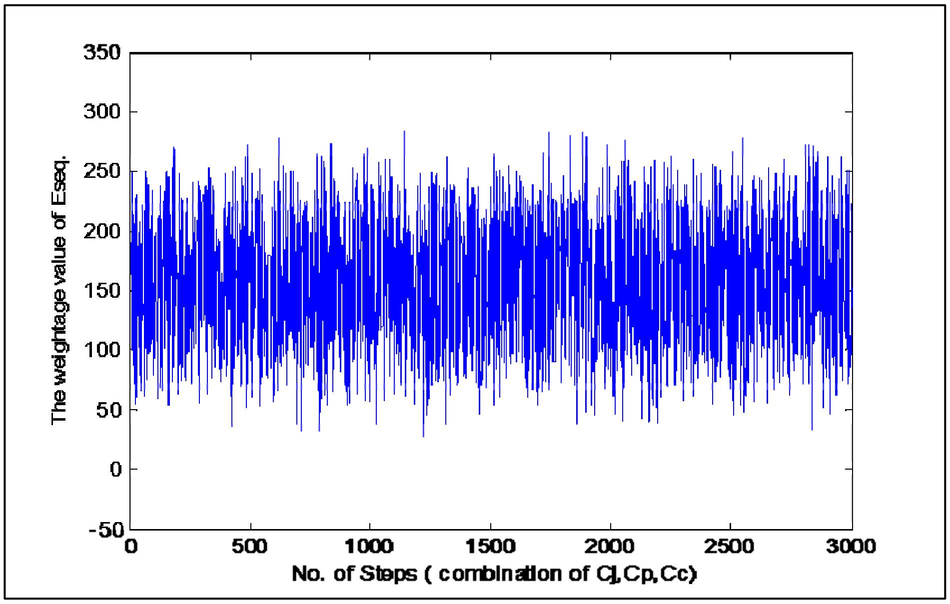


Figure 5.2: The simulation result for C_P, C_J, C_C

Based on the above graph, the following conditions are obtained;

Table 5.3: The simulation condition for energy constants

Energy Constants			Cost Constants	
C_J	C_P	C_C	ρ_t	ρ_t
45	35	45	0.5	0.5

Table 5.4: The simulation condition for ant parameters

Influencing Parameter of Pheromone Trail		Pheromone Evaporation Rate	Base Part	Assembly Directions
α	β	ρ	a	$x, y, \bar{x}, \bar{y}, \bar{z}$
1	2	0.25		

5.3.2 Results and discussions

An ant colony algorithm-based approach has been used to the generation of optimal stable assembly sequence. The influence of ACO parameters on the performance of the algorithm has been evaluated taking into an account of pheromone updating and the sequence quality in terms of energy matrixes to converge the solution. It has been observed that, lower value of pheromone allow a fast convergence toward the solution. The lower values accelerate the evaporation process of the pheromone in low quality trails, increasing more and more the relevance to get the best solutions. Extremely lower values of pheromone can cause the disappearance of all types of trail. The ant colony algorithm starts from searching the first disassembly node to the last one. In between the search process follows path based on the kind of parameters selected and to an extent pheromone used. Ultimately, the sequence generated in the algorithm is the optimal disassembly sequence to that product. The reverse of it, is optimal assembly sequence.

The work considered two example problems to measure the accuracy of algorithm and the following results were obtained; in grinder assembly the optimal sequence is:

$(a,\bar{x})-(d,\bar{x})-(e,\bar{x})-(b,x)-(c,x)$

There have been several methods for generating robotic assembly sequences. Some of these follow conventional techniques such as liaison method, disassembly method, matrix method, constraint method etc. However these conventional methods give rise to multiple solutions. With increasing number of parts for the assembly product, the problem becomes quite large in size. The problem as such is not acquiescent to any mathematical modeling. Comparable techniques such as GA can also be an alternative proposition for such type of problem. However the very philosophy of ACO makes it more suitable for its application. The results obtained from the case study amply suggest that ACO is one of the most suitable techniques for solving problem of this type.

5.4 Artificial immune system algorithm applied to robotic assembly

For the modified method of immune algorithms, the input is in the form of generation of p antibodies. The antibodies refer to the stable assembly sequences, which are generated from the part contact level graph and its stability criteria. The energy values are calculated from the objective function. The inverse of these are the affinity values. The evaluation result of the grinder assembly for affinity values of the stable assembly sequences are presented in Table 5.5.

Table 5.5: Affinity value of the stable sequences

Stable Sequences					*Energy*	*Affinity*
a	*b*	*c*	*d*	*e*	55.777	0.017928
b	*a*	*c*	*d*	*e*	163.228	0.006126
d	*a*	*e*	*b*	*c*	163.681	0.006109

The first step of the immune algorithm is the cloning principle. The main idea of the cloning principle is that generated antibodies that can recognize the antigens proliferate and mutate to produce specific antibodies. As indicated before, the present algorithm has taken ideas from the clonal selection principle, modeling the fact that only the highest affinity antibodies will proliferate. The result of the cloning procedure is presented in Table 5.6.

Table 5.6: Result of cloning procedure

Seq No	*No of clones*	*Sequence Order*					*Energy*	*Affinity*
1	3	*a*	*b*	*c*	*d*	*e*	55.777	0.17928
		a	*b*	*c*	*d*	*e*	55.777	0.17928
		a	*b*	*c*	*d*	*e*	55.777	0.17928
2	2	*b*	*a*	*c*	*d*	*e*	163.228	0.006126
		b	*a*	*c*	*d*	*e*	163.228	0.006126
3	2	*d*	*a*	*e*	*b*	*c*	163.681	0.006109
		d	*a*	*e*	*b*	*c*	163.681	0.006109

The next step of AIS algorithm is the mutation process. In the mutation process, if original sequence energy is less than the energy obtained after inverse mutation and pairwise interchange mutation, then the original sequence is retained. Otherwise, the sequence generated during inverse or pairwise interchange mutation is taken based on mutation condition.

During the processing after cloning, the following results are found out as shown in Table 5.7.

Table 5.7: Mutation (Inverse and/or Pairwise) result of AIS algorithm

Seq. No.	*No. of clones*	*Sequence Order*					*Energy*	*Affinity*
1	3	*a*	*b*	*c*	*d*	*e*	55.777	0.17928
		a	*b*	*c*	*d*	*e*	53.380	0.0187336
		a	*b*	*c*	*d*	*e*	53.380	0.0187336
2	2	*b*	*a*	*c*	*d*	*e*	163.228	0.006126
		b	*a*	*c*	*d*	*e*	90.473	0.01105302
3	2	*d*	*a*	*e*	*b*	*c*	163.681	0.006109
		d	*a*	*e*	*b*	*c*	90.149	0.0110927

After sorting the sequence and deleting repetition, the receptor editing process has to be done. In the previous process, there is no repetition of sequences occur. For the algorithm, the worst percentage of the assembly sequences is taken as 30%. The results obtained during the receptor editing process are presented in Table 5.8.

Table 5.8: Receptor editing result of AIS algorithm

Seq. No.	*Sequence Order*					*Energy*	*Affinity*
1	*a*	*b*	*d*	*e*	*c*	53.380	0.01873360
2	*a*	*d*	*b*	*c*	*e*	53.380	0.01873360
3	*a*	*b*	*c*	*d*	*e*	55.777	0.01792800
4	*d*	*a*	*b*	*c*	*e*	90.149	0.01109270
5	*b*	*a*	*d*	*e*	*c*	90.473	0.01105302

It may be clear that, the process eliminates the higher affinity values. That means the assembly having higher costs are eliminating.

Table 5.9: Assembly sequences after one iteration

Seq. No.	*Sequence Order*				
1	*a*	*b*	*d*	*e*	*c*
2	*a*	*d*	*b*	*c*	*e*
3	*a*	*b*	*c*	*d*	*e*
4	*d*	*a*	*b*	*c*	*e*
5	*b*	*a*	*d*	*e*	*c*

The results of the stable assembly sequences are obtained from first iteration of the AIS algorithm as shown in Table 5.9. When subsequent iterations are taken place, the results satisfying convergence tendency are obtained. Table 5.10 is the results of the grinder assembly after 900 iterations.

Table 5.10: Assembly sequence after 900 iterations

Seq. No.	*Sequence Order*					*Energy*	*Affinity*
1	*a*	*d*	*b*	*c*	*e*	53.380	0.01873360
2	*a*	*b*	*d*	*e*	*c*	53.380	0.01873360
3	*a*	*d*	*b*	*e*	*c*	54.017	0.01851269
4	*a*	*b*	*c*	*d*	*e*	55.777	0.01792800
5	*a*	*d*	*e*	*b*	*c*	55.777	0.01792800

The dotted line shows the optimal solutions to the problem. For this product, there are two optimal solutions. The affinity values of last two sequences are lower down. The cost of the assembly increases as the gripper changes occurs. From the result, it can be shown that, the variations of the affinity values of the final result are higher. This is due to the facts that, a lot of constraints are applying for the generating optimal stable assembly sequences. The choices of generating optimal solutions are drastically reduced.

5.4.1 Discussion

A new modified optimization algorithm is presented to generate optimal stable assembly sequence based on the clonal selection principle. The present algorithm maintains population diversity by the immune operations of immune selection and immune metabolism, and employs a clonal selection operation to enhance the local search. Furthermore, the immune operation of inoculation is incorporated to improve the quality of solution candidates to speed up convergence. The approach is able to produce results similar or better than those generated by other algorithms that are representative of the state-of-the-art in evolutionary objective optimization. The present approach uses an affinity measure to control the amount of mutation to be applied to the antibodies. Affinity here is defined in terms of non-dominance and feasibility. These affinity measures, combined with the population are used to distribute non-dominated solutions in a uniform way. The approach proposed also uses a very simple mechanism to deal with assembly constrained functions, and the results indicate that such mechanism, despite its simplicity, is very effective in practice.

The approach proposed also uses a very simple mechanism to deal with assembly constrained functions, and the results indicate that such mechanism, despite its simplicity, is very effective in practice. The results show that the evolved method could be effectively utilized to generate stable assembly sequences.

CHAPTER 6

Conclusions and Future Scopes

6.1 Overview

For a wide variety of manufacturing industries, assembly accounts for a significant percentage of the total manufacturing cost of a product. Until recently, much effort has been devoted to designing and producing individual parts, while very little effort has been directed towards optimizing product assembly. Hence, it is likely that an improvement in this area could yield a great reduction in both manufacturing cost and time. Following the present trend it can be forecast that in the future an increasing number of products will be designed for automatic robotic assembly. It is therefore, much more important to plan the assembly process and to generate the suitable sequence of commands for the assembling robots.

An assembly sequence plays a key role in determining many important things in product assembly. For instance, each of the following is closely dependent on the selected assembly sequence:

i. The characteristics of a product and an assembly system;
ii. The difficulty of product assembly;
iii. The unit cost of a product;
iv. The error rate.

Hence, the assembly sequence should be carefully determined in order to produce the best output. For optimal assembly sequencing, a planner should be aware of as many desirable assembly sequences as possible. Unfortunately, the number of desirable assembly sequences rises explosively as the number of parts as well as the complexity of the product increase. Without using efficient algorithms, generation of the correct assembly sequences is not possible. The assembly sequencing based only on a designer's knowledge and experience may miss many efficient assembly sequences. The continuous evolving of the design of a product greatly increases the flexible automation of the assembly sequencing since small changes in the design of a product may cause great changes in assembly sequences. Also, common use of computing systems to design products and individual parts encourages the industrial automation of the assembly sequencing. The flexible automation of assembly sequencing is useful in many industrial applications, such as; automobile industries, aerospace industries, marine technologies and so on.

The flexible automation of assembly sequencing may be very helpful in determining prototypes for new or continually evolving products with time and cost savings. The robotic assembly system is a programmable and flexible one. It is appropriate to develop method(s) which can be incorporated with the robot motion program thereby improving the efficiency and effectiveness of the system.

6.2 Importance and usefulness

The results obtained from this research are sure to help the system designers to apply the assembly/disassembly method for robotic assembly system for generating the cost effective assembly sequence. In this aspect, the designer requires to construct the directional interference and connectivity for the product. The computing methods developed and/or modified in this work make the generation of optimized sequences faster and more accurate.

The constraints used in methods help a designer to examine all valid assembly sequences and to reduce the large number of sequences to a size useful for further analysis. As a result, the designer can find a better robotic assembly sequence correctly without having problem of search space explosion and local minima.

Generation of stable sequences by PCLG, and reducing them to an optimal value by ACO and AIS will certainly benefit the automated industries to reduce the cost of the assembly.

The work presents a combined approach for the selection of appropriate method for the generation of assembly sequences in the contest of robotic assembly system, testing of the stability of the generated sequences, and finding the optimal sequence and hence is definitely a new dimension to this subject.

In summary, the present piece of work can be seen as a guide-line for many designers and planners of product assembly, especially those using robotic workcells. Some of the observations can be outlined as follows.

i. Previously GA technique has been used by researchers to solve such problems and the results are quite encouraging.
ii. The techniques such as ACO and AIS are also found to be interesting and suitable for such kind of formulations and hence they have been chosen for being applied to obtain optimized assembly sequence with additional constraints.
iii. The additional constraints in the present problem are: Precedence constraints and connectivity constraints
iv. Since the disassembly method gives rise to a correct assembly sequence, the same strategy has been used to find the assembly sequences in the present problem.
v. The proposal and the formulation of the problem for applying ACO and AIS has been well appreciated by peer researchers and the results are very much in line with the alternative and existing techniques used by previous researchers.

6.3 Scope for future work

Although extensive research works have been carried out, several avenues for future research still remain open. Several methods for the generation of assembly sequences are discussed in course of the present work. However, the core work concentrates on important and simple methods for selecting the appropriate method for robotic assembly system. Nevertheless, sufficient research outcome may be realized in some of the following areas.

a) Manual generation of precedence constraints is obviously a cumbersome work. Any attempt towards automating the precedence constraints will be more meaningful and beneficial.
b) Automating the sequence generation would be much useful in the field of product assembly.
c) The reduction in assembly cost may be achieved by designing products in such a way that they are easy to handle and assemble. In order to assemble the newly designing products, simpler and less expensive robots are required. Basically, these robots need to perform simple straight line movements, mainly along the principal axes, while functional operations like screwing are performed by the gripper. The required information must therefore be attached to the specific item. Thus, not only the robots are less expensive, but more importantly, they need fewer adjustments and yet can maintain the required accuracy even better due to their simple movements.
d) Incorporating the logic for assembly of parts, the task for robots with the logic of sequence generating process shall be a commanding work that will make the entire robotic assembly system more flexible.

References

1. De Fazio T. and Whitney D., “Simplified generation of all mechanical assembly sequences”, *IEEE J. Robotics and Automations* RA-3, No. 6, 1987, pp. 640-658.
2. Baldwin D., Abell T., MaxLui M., De Fazio T. and Whitney D., “An integrated computer aid for generating and evaluating assembly sequences for mechanical products”, *IEEE Trans. on Robotics and Automation* 7, No. 1, 1991, pp. 78-94.
3. Chan P., “Automatic assembly sequence generation by pattern matching”, *IEEE Transactions on Systems, Man and Cybanetics*, 1991, pp. 376–389.
4. Homan de Mello L. S. and Sanderson A. C., “Representations of mechanical assembly sequences”, *IEEE Transactions on Robotics and Automation*, 1991, pp. 211–227.
5. Homan de Mello L. S. and Sanderson A. C., “And/or graph representation of assembly plans” *IEEE Transactions on Robotics and Automation*, 1990, pp. 188–199.
6. Homan de Mello L. S. and Sanderson A. C., “A correct and complete algorithm for the generation of mechanical assembly sequences”, *IEEE Trans Robot Automation* 7(2), 1991, pp. 228–240.
7. Homan de Mello L. S. and Sanderson A. C., “Representations of Assembly Sequences”, *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*., 1989, pp.211-227.
8. Mascle and Figour J., “Methodological approach of sequences determination using the disassembly method” *Proc. Rensselaer’s Second International Conference on Computer Integrated Manufacturing, Troy*, 1990, pp.483-490.

9. Huang Y. and Lee C., “Precedence knowledge in feature mating operation assembly planning”, *Proc. 1989 IEEE International Conference on Robotics and Automation*, 1989, pp. 216-221.
10. Huang Y. and Lee C., “An automated assembly planning system” *Proc. 1990 IEEE International Conference on Robotics and Automation*, 1990, pp. 1594-1599.
11. Huang Y. and Lee C., “A framework of knowledge-based assembly planning” *Proc. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp.599-604.
12. Lee S., “Asssembly planning based on subassembly extraction” *Trans. Robotics and Automation*, Special Issue on Manufacturing Systems: Design, Planning, Scheduling and Real-Time Control, 1990, pp.1-32.
13. Lee S. and Shin Y. G., “Assembly planning based on subassembly extraction” *Proc. 1990 IEEE International Conference on Robotics and Automation*, 1990, pp.1606-1611.
14. Lee S. H, “Disassembly planning by subassembly extraction,” *Proc. 3rd ORSA/TIMS Conf. Flex. Manuf. Sys.*, 1989, pp.383-388.
15. Cho and Cho H., “Inference on robotic assembly precedence constraints using part contact level graph” *Robotica*, Vol. 11, 1993, pp. 173-183.
16. Cho D. Y., Shin C. K. and Cho H. S., “Automatic inference on stable robotic assembly sequences based upon the evaluation of base assembly motion instability” *Robotica*, Vol. 11, 1993, pp.351-362.
17. Shin C. K. and Cho H. S., “On the generation of robotic assembly sequences based upon separability and assembly motion instability”, *Robotica*, 12, 1994, pp.7-15.
18. Shin C. K., Hong D. S., and Cho H. S., “Disassembly analysis for generating robotic assembly sequences”, 1995, *IEEE Proc. Int. Conf. Rob. Auto.*, Vol. 2, pp.1284-89.

19. Delchambre, P. Gasspart and A. Wafflard, "An automated assembly planning approach" *Proc. 10th International Conference on assembly automation*, 1988, pp.319-326.
20. Delchambre, "A pragmatic approach to computer aided assembly planning" *Proc. 1990 IEEE International Conference of Robotics and Automation*, 1990, pp.1600-1605.
21. Delchambre and Wafflard A., "An automatic, systematic and user-friendly computer aided planner for robotized assembly" *Proc. 1991 IEEE International Conference of Robotics and Automation*, 1991, pp.592-598.
22. Mosemann H., Rohrdanz F. and Wahl F. M., "Stability Analysis of Assemblies Considering Friction" *IEEE Trans. of Robotics and Automation*, Vol.13, no.6, 1997, pp. 805-813.
23. Adachi Y., "Assembly sequence generation using constraints between parts", *In Proc. Ann. Conf. Robotics Soc., Japan*, 1992, pp. 751–752.
24. Adachi Y., "Representing of product assembly architecture for assembly sequence generation", *In Proc. 3th FAN Symp.*, 1993, pp.99–100.
25. Hong D. S. and Cho H.S, "A neural-network-based computational scheme for generating optimized robotic assembly sequences," *Elsevier, Journal of Engineering Applications of Artificial Intelligence*, Vol. 8(2), 1995, pp. 129-145.
26. Hong D. S. and Cho H. S, "Optimization of robotic assembly sequences using neural- network," *IEEE International Conference on Intelligent Robots and System,* 1993, pp. 232-239.
27. Hong D. S. and Cho H. S, "Generation of robotic assembly sequences using a simulated annealing," *IEEE International Conference on Robots and Systems*, 1999, pp. 1247-52.
28. Doigo, M., "Ant colony system: A cooperative learning approach to the travelling salesman problem", *IEEE Transactions on Evolutionary Computation. IEEE*, 1(1), 1997, pp. 53-66.

29. Dorigo M et al., “The ant system: optimization by a colony of cooperating agents”, *IEEE Trans Syst Man Cybernet*, 26(1), 1996, pp. 29–41.
30. Dorigo M, Gambardella LM “Ant colony system: a cooperative learning approach to the traveling salesman problem”, *IEEE Trans Evolu. Comput.* 1(1), 1997, pp.53–66.
31. Wang, J., F., Liu, J., H., Zhong, Y., F., “A novel ant colony algorithm for assembly sequence Planning”, *International Journal of Advanced Manufacturing Techology. Springer-Verlag*, 25, 2005, pp.1137-43.
32. Bonneville F., Perrard C., Henrioud J.M., “A genetic algorithm to generate and evaluate assembly plans”, *Proceedings of the IEEE Symposium on Emerging Technology and Factory Automation*, 1995, pp.231–239.
33. Cao P. B. and Xiao R. B, “Assembly planning using a novel immune approach”, *International Journal of Advance Manufacturing Technology*, Vol. 31, 2007, pp.770-82.
34. De Castro L. N., Timmis J. I., “Artificial immune system: a new computational intelligence approach”, *Springer, Berlin Heidelberg New York*, 2002.
35. Chandrasekaran M., Asokan P., Kumanan S., Balamurugan T. and Nickolas S., “Sloving job shop scheduling problems using artificial immune system”, *International Journal of Advanced Manufacturing Technology. Springer-Verlag.*, 31, 2006, pp. 580-593.
36. Chandrasekharan R. and Ziegler H., “Ant colony algorithms for permutation flowshop scheduling to minimize makesspan/total flowtime of jobs”, *European journal of operation research,* Elsevier, 2004, pp.426-438.
37. Rajasekaran S., Amalraj R., “Ants net genetic algorithm for the travelling salesman problem”, *Journal of the CSI,* 2003, pp.9-17.

Appendices

A-I: Computer program for generating E_C matrix

```
% PROGRAMMING FOR 'Ec' MATRIX
% INPUT BEGINS
clc
syms rc vc sw rf
cc=45;

g=[1 2 3 1 2 1 2 3 1 2 1 2 3 1 2 1 2 3 1 2 1 2 3 1 2];

for p=1:25
   for q=1:25
      ec(p,q)=90;
   end
end

cdab=[0 rc rc;rc rc rc];          fdab=[0 0 0;rf 0 0];
cdac=[0 0 0;vc 0 0];              fdac=[0 0 0;sw 0 0];
cdad=[rc rc rc;0 rc rc];          fdad=[rf 0 0;0 0 0];
cdae=[vc 0 0;0 0 0];              fdae=[sw 0 0;0 0 0];
cdba=[rc rc rc;0 rc rc];          fdba=[rf 0 0;0 0 0];
cdbc=[0 0 0;rc 0 0];              fdbc=[0 0 0;0 0 0];
cdca=[vc 0 0;0 0 0];              fdca=[sw 0 0;0 0 0];
cdcb=[rc 0 0;0 0 0];              fdcb=[0 0 0;0 0 0];
cdda=[0 rc rc;rc rc rc];          fdda=[0 0 0;rf 0 0];
cdde=[rc 0 0;0 0 0];              fdde=[0 0 0;0 0 0];
cdea=[0 0 0;vc 0 0];              fdea=[0 0 0;sw 0 0];
cded=[0 0 0;rc 0 0];              fded=[0 0 0;0 0 0];
```

```
for p=1:5
   for q=6:10
      if cdab(g(p))~=0 | cdab(g(q))~=0
         lm1(p,q)=0;
      else
         lm1(p,q)=0;
      end
      if fdab(g(p))~=0 | cdab(g(q))~=0
         lm2(p,q)=0;
      else
         lm2(p,q)=1;
      end
      ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
   end
   for q=11:15
      if cdac(g(p))~=0 | cdac(g(q))~=0
         lm1(p,q)=0;
      else
         lm1(p,q)=0;
      end
      if fdac(g(p))~=0 | cdac(g(q))~=0
         lm2(p,q)=0;
      else
         lm2(p,q)=1;
      end
      ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
   end
   for q=16:20
```

```
            if cdad(g(p))~=0 | cdad(g(q))~=0
                lm1(p,q)=0;
            else
                lm1(p,q)=0;
            end
            if fdad(g(p))~=0 | cdad(g(q))~=0
                lm2(p,q)=0;
            else
                lm2(p,q)=1;
            end
            ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
        end
        for q=21:25
            if cdae(g(p))~=0 | cdae(g(q))~=0
                lm1(p,q)=0;
            else
                lm1(p,q)=0;
            end
            if fdae(g(p))~=0 | cdae(g(q))~=0
                lm2(p,q)=0;
            else
                lm2(p,q)=1;
            end
            ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
        end
    end

    for p=6:10
        for q=1:5
```

```
      if cdba(g(p))~=0 | cdba(g(q))~=0
         lm1(p,q)=0;
      else
         lm1(p,q)=0;
      end
      if fdba(g(p))~=0 | cdba(g(q))~=0
         lm2(p,q)=0;
      else
         lm2(p,q)=1;
      end
      ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
   end
   for q=11:15
      if cdbc(g(p))~=0 | cdbc(g(q))~=0
         lm1(p,q)=0;
      else
         lm1(p,q)=0;
      end
      if fdbc(g(p))~=0 | cdbc(g(q))~=0
         lm2(p,q)=0;
      else
         lm2(p,q)=1;
      end
      ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
   end
end
for p=11:15
   for q=1:5
      if cdca(g(p))~=0 | cdca(g(q))~=0
```

```
            lm1(p,q)=0;
        else
            lm1(p,q)=0;
        end
        if fdca(g(p))~=0 | cdca(g(q))~=0
            lm2(p,q)=0;
        else
            lm2(p,q)=1;
        end
        ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
    end
    for q=6:10
        if cdcb(g(p))~=0 | cdcb(g(q))~=0
            lm1(p,q)=0;
        else
            lm1(p,q)=0;
        end
        if fdcb(g(p))~=0 | cdcb(g(q))~=0
            lm2(p,q)=0;
        else
            lm2(p,q)=1;
        end
        ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
    end
end
for p=16:20
    for q=1:5
        if cdda(g(p))~=0 | cdda(g(q))~=0
            lm1(p,q)=0;
```

```
        else
            lm1(p,q)=0;
        end
        if fdda(g(p))~=0 | cdda(g(q))~=0
            lm2(p,q)=0;
        else
            lm2(p,q)=1;
        end
        ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
    end
    for q=21:25
        if cdde(g(p))~=0 | cdde(g(q))~=0
            lm1(p,q)=0;
        else
            lm1(p,q)=0;
        end
        if fdde(g(p))~=0 | cdde(g(q))~=0
            lm2(p,q)=0;
        else
            lm2(p,q)=1;
        end
        ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
    end
end
for p=21:25
    for q=1:5
        if cdea(g(p))~=0 | cdea(g(q))~=0
            lm1(p,q)=0;
        else
```

```
            lm1(p,q)=0;
        end
        if fdea(g(p))~=0 | cdea(g(q))~=0
            lm2(p,q)=0;
        else
            lm2(p,q)=1;
        end
        ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
    end
    for q=16:20
        if cdea(g(p))~=0 | cdea(g(q))~=0
            lm1(p,q)=0;
        else
            lm1(p,q)=0;
        end
        if fdea(g(p))~=0 | cdea(g(q))~=0
            lm2(p,q)=0;
        else
            lm2(p,q)=1;
        end
        ec(p,q)=cc*(lm1(p,q)+lm2(p,q));
    end
end
```

A-II: Computer program for generating ‘Ep’

```
% PROGRAM TO FINDOUT THE 'Ep'
% these are the input of one problem
clc
syms a b c d e bd
% INPUT PRECEDENCE CONSTRAINTS
ax=[bd];ay=[bd];az=[bd];amx=[bd];amy=[bd];bx=0;by=0;bz=0;bmx=0;bmy=
0;
cx=[b];cy=[b];cz=[b];cmx=[b];cmy=[b];dx=0;dy=0;dz=0;dmx=0;dmy=0;
ex=[d];ey=[d];ez=[d];emx=[d];emy=[d];
i=[ax ay az amx amy bx by bz bmx bmy cx cy cz cmx cmy dx dy dz dmx
dmy ex ey ez emx emy];
j=[ax ay az amx amy bx by bz bmx bmy cx cy cz cmx cmy dx dy dz dmx
dmy ex ey ez emx emy];
cp=35;
% INPUT ENDS
k=[a a a a a b b b b b c c c c c d d d d d e e e e e];
for p=1:25
    for q=1:25
        if i(p)==0
            mu1(p)=0;
        else
            mu1(p)=1;
        end
        if j(q)==k(p) | j(q)==0
            mu2(q)=0;
        else
            mu2(q)=1;
        end
        ep(p,q)=cp*(mu1(p)+mu2(q));
    end
%    disp(k(p))
end
```

```
for p=1:5
    for q=1:5
        ep(p,q)=70;
    end
end
for p=6:10
    for q=6:10
        ep(p,q)=70;
    end
end
for p=11:15
    for q=11:15
        ep(p,q)=70;
    end
end
for p=16:20
    for q=16:20
        ep(p,q)=70;
    end
end
for p=21:25
    for q=21:25
        ep(p,q)=70;
    end
end
```

A-III: Computer program for generating cost matrices

```
% ---------------PROGRAM FOR COST MATRIX---------------------
% Input begins here
clc
cj=45;

for p=1:25
    for q=1:25
        ej(p,q)=45;
    end
end

m=1;
ros=0.5;
rot=0.5;

% Input ends

for p=1:5
    for q=6:10
        sba(p,q)=3;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
    for q=11:15
        sba(p,q)=3;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
    for q=16:20
        sba(p,q)=3;
```

```
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
    for q=21:25
        sba(p,q)=3;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
end
for p=6:10
    for q=1:5
        sba(p,q)=3;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
    for q=11:15
        sba(p,q)=7;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
end
for p=11:15
    for q=1:5
        sba(p,q)=3;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
```

```
        ej(p,q)=cj*j(p,q);
    end
    for q=6:10
        sba(p,q)=7;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
end
for p=16:20
    for q=1:5
        sba(p,q)=3;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
    for q=21:25
        sba(p,q)=7;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
end
for p=21:25
    for q=1:5
        sba(p,q)=3;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
    for q=16:20
```

```
        sba(p,q)=7;
        cas(p,q)=sba(p,q)/(12*m);
        cnt(p,q)=0;
        j(p,q)=(ros*cas(p,q))+(rot*cnt(p,q));
        ej(p,q)=cj*j(p,q);
    end
end
```

CURRICULUM VITA

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

PERSONAL PROFILE

Name : **SURAJIT SHARMA**

Father's Name : **MAHESH PRASAD SHARMA**

Date of Birth : 20th June 1978

Fields of Interest : Robotics, FMS, CAD/CAM/CIM & Soft Computing

Phone: +91 9967689750, E-mail: surajitnit@gmail.com

surajit.sharma@lechlerindia.com

Academic Qualification

2009	MTech(R) Thesis Submitted, NIT, Rourkela
2000	B.E (Manufacturing Sc & Engg), UCE, Burla

Work Experience

Employer	Date of Joining	Date of Leaving	Nature of Job
Lechler India Pvt. Ltd., Thane	01.07.08	Continuing	Planning and Manufacturing
NIT Rourkela (MTech (R))	15.01.07	Withdrawal 24.06.08	Research
Puroshottam Institute of Engineering and Technology, Rourkela	16.02.06	14.01.07	Teaching
UCE Burla	01.07.04	30.12.05	P T Teaching
UCE Burla	01.07.02	31.05.03	P T Teaching
IACM, New Delhi	01.04.01	30.06.02	Developer
UCE Burla	01.07.00	31.03.01	P T Teaching

PUBLICATIONS

International Journals

1. B. B. Biswal, S. Sharma, and P. Dash, "Correct assembly sequence for robotic assembly using motion instability and part contact level graphs" Int. J. Computer Applications in Technology, Vol. 36, No. 2, 2009,pp 149–158.

International Conferences

1. S. Sharma and B. B. Biswal, "Optimization of Robotic Assembly Sequence using Immune based Technique" 2nd International and 23rd All India Manufacturing Technology, Design and research Conference on *Manufacturing and Automation*, Dec 15-17, 2008, organized by Department of Mechanical Engineering, IIT Madras, Chennai.
2. S.Sharma, B.B.Biswal, P.Das and B.B.Choudhury "Generation of Optimized Robotic Assembly Sequence using Ant Colony Optimization", Proc. of IEEE CASE, Washington DC, USA, pp. 894-899, 2008.

Journals Communicated

1. B.B.Choudhury, B.B.Biswal and S.Sharma, "Optimal Task Allocation in a Multirobot Environment Using Cost Indices", Communicated to Journal of Mechanical Engineering, ICFAI Press.

2. B.B.Choudhury S.Sharma and B.B.Biswal "Integrated Multi-Robotic Assembly Planning for Improved Productivity" Communicated to International Journal of Computational Vision and Robotics (IJCVR).

3. B.B.Choudhury ,S.Sharma and B.B.Biswal "A Strategic approach to integrated Product Design for Small to Medium Sized Enterprises" Communicated to Journal of Interscience Management Review.

4. B.B.Choudhury S.Sharma and B.B.Biswal "An Integrated Approach to Optimized Assembly Planning and Task Assignment in Multi-Robotic Cell" Communicated to International Journal of Computer Science and Management System (IJCSMS).

5. B.B.Choudhury S.Sharma and B.B.Biswal "An Optimized Multirobotic Assembly System for Industrial Applications" Communicated to International Journal of Advaced Robotic Systems.