Computer Aided Optimal Robotic Assembly Sequence Generation

Dissertation submitted in partial fulfillment
of the requirements of the degree of
Doctor of Philosophy
in
Industrial Design
by
M V A Raju Bahubalendruni
(Roll Number: 512ID1006)

based on the research carried out
Under supervision of
Prof. Bibhuti Bhusan Biswal

December, 2016

Department of Industrial Design
National Institute of Technology Rourkela

Department of Industrial Design
National Institute of Technology Rourkela

December, 2016

Certificate of Examination

Roll Number: *512ID1006*
Name: *M V A Raju Bahubalendruni*
Title of Dissertation: *Computer Aided Optimal Robotic Assembly Sequence Generation*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Industrial Design* at National Institute of Technology Rourkela. We are satisfied with the volume, quality, correctness, and originality of the work.

(Prof. Bibhuti Bhusan Biswal)
Principal Supervisor

(Prof. Mohammed Rajik Khan)
Member, DSC

(Prof. Jonnalagadda Srinivas)
Member, DSC

(Prof. Chandan Kumar Biswas)
Member, DSC

(Prof. Bijan Sarkar)
External Examiner

(Prof. Dayal Ramakrushna Parhi)
Chairperson, DSC

(Prof. Mohammed Rajik Khan)
Head of the Department

Department of Industrial Design
National Institute of Technology Rourkela

Prof. Bibhuti Bhusan Biswal
Professor and Dean (Faculty Welfare)

December, 2016

Supervisors' Certificate

This is to certify that the work presented in this dissertation entitled *Computer Aided Optimal Robotic Assembly Sequence Generation* by *M V A Raju Bahubalendruni*, Roll Number 512ID1006, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Industrial Design*. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Bibhuti Bhusan Biswal
Principal Supervisor

Dedicated
to
My Loving
Parents

M V A Raju Bahubalendruni

Declaration of Originality

I, *M V A Raju Bahubalendruni*, Roll Number: *512ID1006* hereby declare that this dissertation entitled *Computer Aided Optimal Robotic Assembly Sequence Generation* presents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the section “References” or “Bibliography”. I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

M V A Raju Bahubalendruni

December, 2016
NIT Rourkela

Acknowledgement

This dissertation is a result of the research work that has been carried out at **National Institute of Technology, Rourkela**. During this period, the author came across with a great number of people whose contributions in various ways helped in the field of research and they deserve special thanks. It is a pleasure to convey the gratitude to all of them.

First of all the author expresses his heartiest gratitude to his supervisor and guide **Dr. B. B. Biswal,** Professor and Dean, Faculty Welfare, NIT, Rourkela for his valuable guidance, support and encouragement in the course of the present work. The successful and timely completion of the work is due to his constant inspiration and constructive criticisms. The author cannot adequately express his appreciation to him. The author records his gratefulness to Madam **Mrs. Meenati Biswal** for her constant support and inspiration during his work and stay at NIT, Rourkela.

The author take this opportunity to express his deepest gratitude to **Prof. M.R. Khan,** Head of the Department of the Department of Industrial Design, NIT Rourkela and all other **DSC Members** for their constant advice, useful discussions, encouragement and support in pursuing the research work.

The author is grateful to **Prof. S.K. Sarangi**, Director, NIT, Rourkela, for his kind support and concern regarding his academic requirements.

The author willing to extend his thanks to Dr. BBVL Deepak, Mr. Manish Kumar, Mrs. Radharani Nayak, Mr. Balamurali Mr. Panda BN Mr. Sahu OP, Mr Ray BB, Mrs. Mishra N, Mr Sampath and all his friends for their continuous support.

The completion of this work came at the expense of author's long hours of absence from home. Words fail to express his indebtedness to his loving wife **Smrutilekha** and loving son **Lakshmi Parardhi,** for their understanding, patience, active cooperation and after all giving their times throughout the course of the doctoral dissertation. The author thanks them for being supportive and caring. His **parents** and **family members** deserve special mention for their inseparable support and prayers.

Last, but not the least, the author thank the one above all, the omnipresent **Goddess**, for giving him the strength during the course of this research work.

M V A Raju Bahubalendruni
Roll Number: 512ID1006

December, 2016
NIT Rourkela

Abstract

Robots are widely used for assembly operations across manufacturing industries to attain high productivity through automation. An appropriate robotic assembly sequence further minimizes the total production lead time and overall cost by minimizing the number of assembly direction changes, assembly gripper changes and assembly energy thus selection of a valid optimal robotic assembly sequence is significantly essential to achieve economized manufacturing process. An optimal assembly sequence must comply with various assembly requirements in order to make sure that the sequence of assembly operations is functionally feasible in physical environment.

In order to test an assembly sequence for its practical possibility, necessary assembly information must be collected accurately from the product. Obtaining such assembly information from product drawings or Computer Aided Design (CAD) models in manual mode were involved in lots of complexity and needs high level skills to ensure correctness. Though retrieving such information from products with less number of parts is simple and less time consuming, for products composed of huge number parts it is very complicated and time consuming. Besides retrieving the assembly information, using it for validating an assembly sequence further raises the complexity of the Assembly Sequence Generation (ASG) problem. To perform optimal feasible assembly sequence generation efficiently, an effective computer aided automated method is developed and executed at two phases. The first phase of research is mainly focused on representing the assembly information in a streamlined manner by considering all possible states of assembly configurations for ease of computerization and developing efficient methods to extract the assembly information automatically from CAD environment though Computer Aided Automation (CAA). These methods basically use assembly contact analysis, part transformations and laws of equilibrium & balancing of rigid bodies. From the existing ASG methods, it is observed most of the researchers ignored/not-considered few of the assembly information such as assembly stability data and mechanical feasibility data due to higher complexity in retrieving it from CAD environment.

In the current work, the effect of considering and ignoring assembly stability data and mechanical feasibility data on the ASG outcomes has been explored for various instances of assemblies. The research study revealed that, ignoring an assembly predicate reduce the total computational time for few instances of assembly configurations without affecting the quality of outcomes. For certain types of assembly configurations ignoring an assembly predicate result an inappropriate assembly sequence.

The second phase of research is predominantly dedicated on developing a novel method named part concatenation to generate the geometrically feasible and stable assembly sequences by considering the automatically retrieved assembly information. Assembly tools/grippers are commonly used for holding and joining a part to the existent part/sub assembly. Parts with similar geometries or surface features can be operated with same tools/gripper. From the huge number of feasible assembly sequences, few may offer less gripper changes, whereas other may offer less assembly directional changes. A combined mode of objective function based on gripper changes, assembly direction changes and assembly energy is defined and is coupled with the computer programs to achieve optimal feasible assembly sequences.

Various assembled products have been created using Part design and Assembly Design modules of Computer Aided Three Dimensional Interactive Application (CATIA) V5 software. Application Program Interface (API) is used to extract assembly information extraction automatically through VB script. Part concatenation method is applied to generate set of all stable and feasible assembly sequences and optimal sequences directly. The proposed method is proven in solving products with large number of parts.

Keywords: Assembly Automation; Assembly Sequence Generation; Geometric Feasibility; Assembly Stability; Assembly Predicates.

Contents

List of Tables

List of Figures

List of Symbols

w_i	Weight factor
c_i	Connection
δ_i	Assembly Energy Factor

Abbreviations

3D	3 Dimensional
ACA	Ant Colony Algorithm
AI	Artificial Intelligence
AIS	Artificial Immune System
ANN	Artificial Neural Network
API	Application Programme Interface
ASG	Assembly Sequence Generation
ASP	Assembly Sequence Planning
BLS	Break out Local Search
B-rep	Boundary Representation
CAA	Computer Aided Automation
CAD	Computer Aided Design
CATIA	Computer Aided Three Dimensional Interactive Application
COG	Centre of Gravity
COM	Centre of Mass
CPU	Central Processing Unit
CSG	Constructive Solid Geometry
DFA	Design for Assembly
DFIG	Disassembly Feasibility Information Graph
DFMA	Design for Manufacturing and Assembly
DOF	Degree of Freedom
EIM	Extended Interference Matrices
FLA	Frog Leaping Algorithm
GA	Genetic Algorithm
GSA	Gravitational Search Algorithm
GOPS	Genetic Operator Probability Sequence
HA	Hybrid Algorithm
ICA	Imperialistic Competitive Algorithm
KBS	Knowledge Based System
LG	Liaison Graph
MA	Memetic Algorithm
NBDG	Non Directional Blocking Graph
NN	Neural Network
PDFMA	Product Design For Manufacturing and Assembly

PSO	Particle Swarm Optimisation
RAM	Random Access Memory
SA	Simulated Annealing Algorithm
STEP	Standard For the Exchange of Product model data
STL	STereo Lithography
VB	Visual Basic

List of Assumptions and Rules

In order to reduce the complexity of the problem and enhance the capabilities of methodologies while solving the assembly sequence planning problem, a list of valid assumptions are made based on the part configurations, their attributes and the limitations in CAD software interaction.

I Parts used to perform assembly operations are solid and rigid in nature by definition, i.e. there would be no change in shape during assembly/disassembly operations.

II Sub-assemblies created at each phase are stable and thus no change in the relative position of the sub-assembled parts at all further phases.

III If a part can be dis-assembled from the existing subassembly along a specified direction without any destructive operation, the part can be assembled in the opposite direction to the same subassembly.

IV Parts are assembled by only part movements under ideal conditions, i.e. no friction and gravity is considered.

V To generate the interference matrix/data, straight line movements along principal axes directions(x+, x-, y+, y-, z+ and z-) are only considered.

VI In the robotic assemblies, the robot end effector has all necessary flexible tooling capabilities.

VII In the robotic assemblies, the robot has necessary degree of freedom (DOF) to reach the object to pick and to perform assembly operation.

VIII The physical connectors are considered as secondary parts such as Nut & Bolt, Screws, Rivets and Connecting pins, which are numbered and placed next to complete set of primary parts. This mode of representation offers flexibility in retrieving mechanical feasibility testing data.

IX Grippers used to handle parts are appended to the primary part name (For example: Part1_G1_G2).

X Press fit/ Force fit, adhesive bonding and welding are not considered to test the stability of an assembly.

XI Friction between the mating faces is not considered in case of inclined surfaces for stability matrix extraction.

XII Parts are fabricated through uniform material and hence the center of mass (COM) coincides with Center of gravity (COG).

Beside these list assumptions, to make ASG process more efficient and effortless during feasibility testing and assembly cut-set generation superset and subset rules are most useful to save computational time (De mello and Sanderson, 1988 & 1990) proposed. These rules are initially suggested by Bourjault (Bourjault, 1984), which are widely accepted and used in other traditional methods.

a. **Superset rule:** If two subassemblies cannot be assembled together due to interference in the path, then adding any additional part to either of the subassembly cannot improve the situation.

b. **Subset rule:** If two subassemblies can join together, removing any part from either of the subassembly, which is not associated with mating liaison(s), cannot influence the situation.

Chapter 1

INTRODUCTION

1.1 Overview of Introduction

The introduction covers role of assembly in product manufacturing, types of assembly systems and evolution of assembly process through fixed and flexible automation. Significance of assembly sequence generation to economize the overall manufacturing process, history of assembly sequence generation methods is briefed along with organization of thesis.

1.2 Product Manufacturing and History of Assembly

Assembly process is not just joining parts together, it is the most prominent process of product development cycle to achieve a functional product. It fetches all the upstream engineering, design and manufacturing processes together to create an object that performs a desired function. Assemblies are the product of the assembly process often these are also the product of a complex engineering design process. This process involves defining the functions that the assembly must perform and then defining physical objects (parts and subassemblies) that will work together to perform those functions. The structure of the assembly must be defined, including all the interrelationships between the parts. Assembly permits parts to function by working together as a system.

The process of assembling mechanical products normally contains a long chain of activities of arranging parts in right quantity and proper sequence, transporting parts and subassemblies to the assembly workstations, joining parts or subassemblies to create assemblies, inspecting to confirm correct assembly and finally testing it to confirm correct function.

Arranging parts is a function which may be executed following any one of many strategies based on estimates of work schedules, the planned production, and list of the parts needed at each phase of assembly operation. Two types of strategies are most commonly used, the push type and the pull type. The push type operates on the basis of a planned production schedule of anticipated final need for finished assemblies. The pull method starts with anticipated demand or orders in hand for finished assemblies (De Fazio, 2004).

Transport of parts or assemblies is carrying them between different stations, where. Assembly gripper, tool, or robot are used to place and orients them so that assembly can be done with only minor operations.

Part mating is the actual process of connecting parts together through permanent or temporary joining process. Joining accompanies mating and usually involves fastening some way. Screws, rivets, adhesive bonding, welding, soldering, crimping, staking, and ultrasonic bonding are some examples. Inspecting usually contains testing and ensuring the correctness and completeness of an assembly operation. One may check the tightness of a screw or freedom of motion of a shaft in its bearings. This is different from functional testing, where the issue is to determine that a subassembly functions correctly.

Assembly is an ancient process, and until very recently it was accomplished exclusively by humans with specialized skills, possibly. The evolution of assembly process until 1940s was largely characterized based on the principle of division of labour to improve efficiency and speed through industrial organization and time studies.

In nineteenth century, industries found the need of skilled worker who can efficiently adjust the shapes of parts so that they would appropriately assemble in less time. Henry Ford recognized that mass production in large quantities can be achieved by eliminating time-consuming fitting operations. To increase the accuracy and repeatability of fabrication machinery, he organized his assembly workers in groups, each of which built large stable subassemblies. His efforts lead to enhanced production capacity and thus it leads to the mass production age.

Automatic assembly machines were developed in the beginning of twentieth century to perform assembly of simple items at faster rate. Although it was implemented on cigarettes in the initial days, but today many simple products such as pens, valves of spray bottles, small motors, razor blade cartridges, and other similar items are assembled by billions of quantity on automatic machines. Each machine performs a simple operation like part feeding/ part inserting/ part screwing. This technology is often referred to as "fixed automation" because most such machines are intended to make one product (often product family) and are difficult or uneconomical to convert for a different product.

In the 1970s, researchers paid their attention towards robot assembly process upon the successful usage of robot for spot welding of car bodies in the late 1960s. High hopes were placed on robots combined with sensor vision systems, force and torque, tactile sensors, powerful hardware and artificial intelligence.

1.3 Manual and Robotic Assembly Systems

People can always complete usual assembly tasks, within imposed limits due to their capacity against the weight of parts ability to do the task. Assembly machines were not really needed but preferable to speed up the process and to ensure the correctness.

Often assembly process is a complex task with more number of actions, most of workers use their skill by experience, observation, practice. Now the challenge for the designer of an assembly machine or robot is to accomplish all the things people do or to eliminate the need to do so. The need of flexibility and repeatability in factory automation, during 1970's robots were found to be reliable.

Industrial robots have to be designed explicitly to do each individual required action, which include sensing, moving, and judging in many dimensions at once at high speed. A highly ingenious robot is much necessary, when manual assembly is deemed inappropriate. Some prominent examples include Sony robots capable to assemble Walkmen, camcorders and cameras. Each of these assemblies required a variety of complicated tasks, including placing springs, meshing gears, and winding rubber belts around multiple pulleys. Compared to human performing such complicated tasks on high precision products leads to more errors, the robots commit fewer errors to perform the tasks more repeatedly, and thus deliver more stable product quality (De Fazio, 2004).

1.4 Significance of Assembly Sequence Generation

Recent advances in advanced manufacturing processes, machining equipment offer great flexibility to product design engineers for selecting complex geometries in order to economize the manufacturing process. The concept of Design for Manufacturing and Assembly (DFMA) utilize sophisticated equipment and advanced materials to simplify the part handling and to reduce the part count in an assembled product which further create need for assembly sequence planning (Kuo et al., 2002 and Boothroyd et al., 2010).

The rapid advancement in manufacturing technology improves the strategy for manufacturing process planning, and thereby also for assembly process planning which leads to reduction of assembly efforts by avoiding unnecessary expenses and related investments.

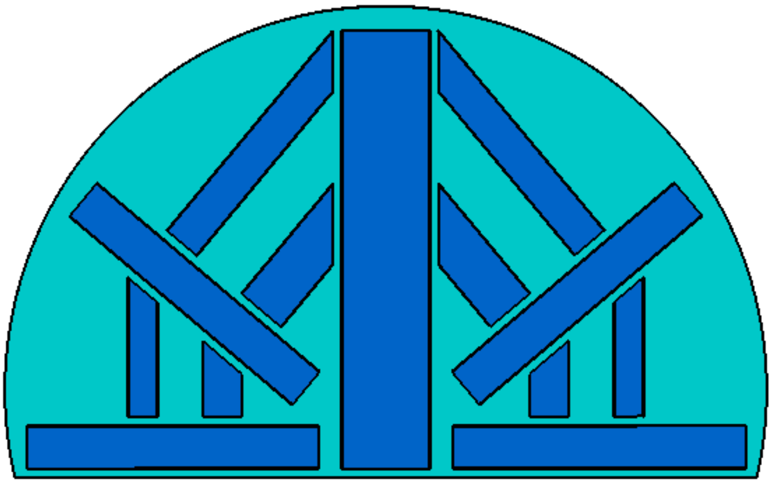


Figure 1.1: Upper forward Pressure bulkhead of an aircraft

Upper pressure bulkhead located at the forward region of aircraft is shown in Figure 1.1. Pressure bulkhead is a primary component in civil aircraft in order to maintain the pressurized cabin during the flight. The basic design of pressure bulkhead must be a stiffened panel to resist the pressure difference. In the early 70's of manufacturing most the aircraft design industries opted for a sheet metal panel attached with symmetrical sheet metal stiffeners with rivets.

The recent advances in manufacturing methods and availability of machines, allows the designer to reduce the part count by joining the rigid mating parts with similar material to reduce the part count. The modified bulkhead does not need any connectors to join the stiffened geometry. Machined upper pressure bulkhead is shown in Figure 1.2.

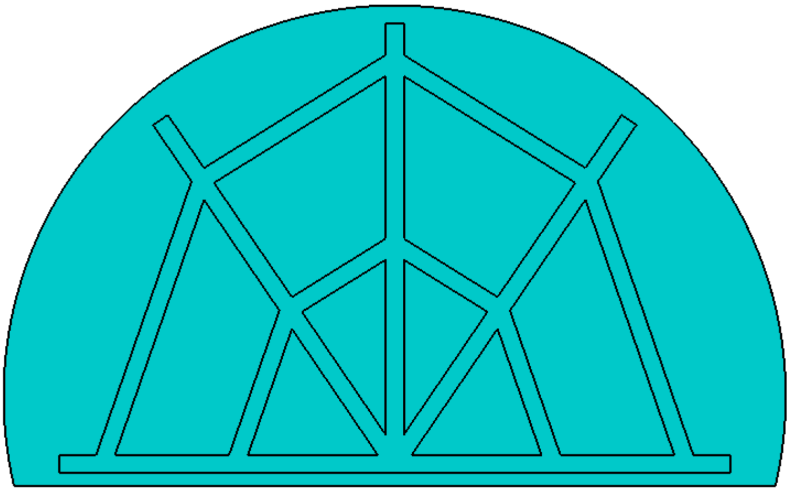


Figure 1.2: Modified upper pressure bulkhead of an aircraft

The introduction of advanced materials changes the day to day manufacturing process, and offer flexibility to design engineers in achieving complex geometries with low material waste and production time. Most of the manufacturing industries opt combination of extrusions and sheet-metal parts to create large scale subassembly structures. The advanced materials with enhanced manufacturing capabilities, the assembly efforts can be reduces by joining the mating parts together.

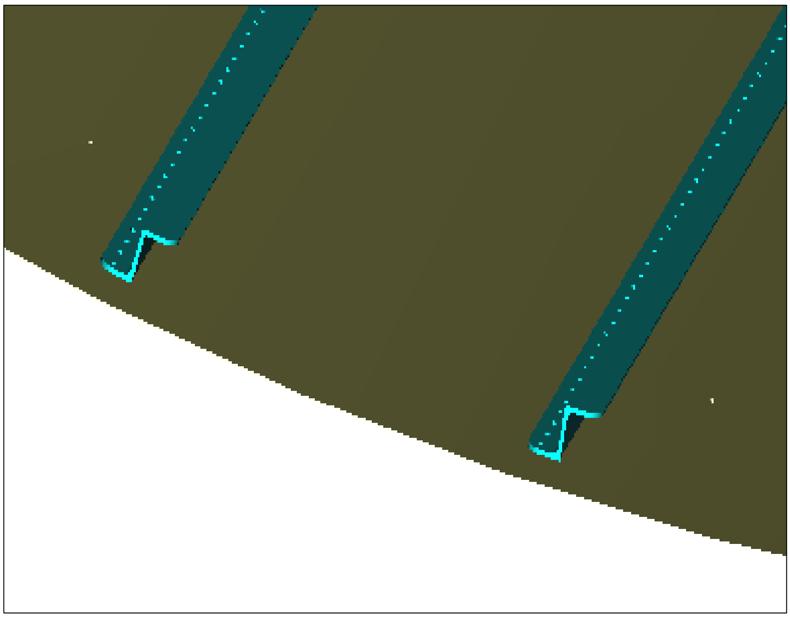


Figure 1.3: Metallic skin-stringer assembly

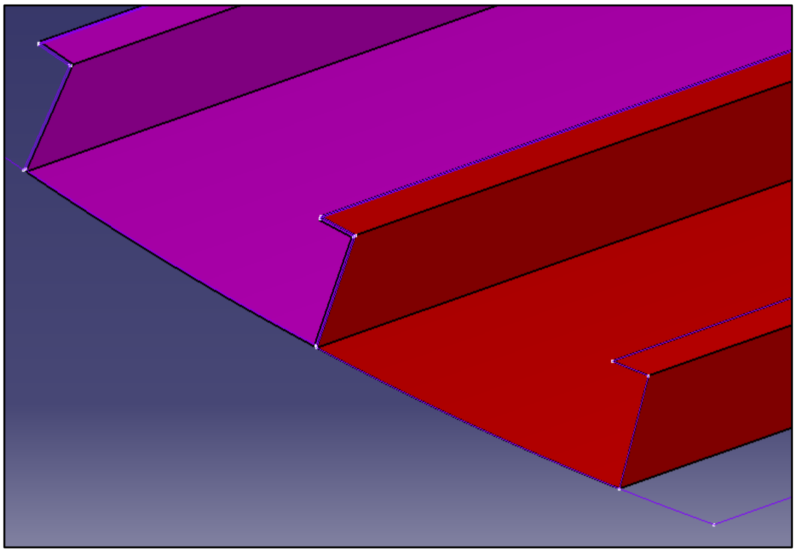


Figure 1.4: Integrated composite skin-stringer

The figures shown in Figure 1.3 and 1.4 represent skin-stringer assembly before and after modification due to advanced materials and their manufacturing methodology. The Figure 1.3 shows configuration of a Sheet metal skin panel attached with extruded metallic stringers joined by riveting, Figure.1.4 shows the configuration of integrated skin stringer composite panel manufactured by autoclave process.

Numerous industries are adopting DFMA concepts to reduce the assembly efforts by reducing the part count and minimizing the number of assembly fasteners. (Boothroyd et al., 2010).

Though the design modification offers greater flexibility in manufacturing process and weight saving thereby minimizing assembly production time. It greatly affects the sequence of assembly operations for product with large number of parts, which consumes lot of time for the industrial engineer to define a new feasible and stable assembly sequence for modified assembly structure. Hence there is strong requirement to develop a computer aided method, which interact with CAD environment and is capable of resulting at least one stable and feasible assembly sequence

1.5 History and Evolution of ASG Methods

Assembly process consumes major stake in the overall manufacturing process, an appropriate assembly sequence can reduce the assembly time and cost. The traditional methods of assembly sequence generation aimed to find out at least one feasible assembly sequence for a given product and/or set of all feasible assembly sequences. These methods evolved in the early 1960's and continued till 1990. Later the introduction of CAD softwares simplified the process of assembly sequence planning by considering assembly features of parts, thereafter the research promoted towards assembly modelling methodologies for assembly sequence planning. Feature based assembly sequence generation methods were developed in (1990-2001) to ease the process of identifying and retrieving part precedence relations. Connector based methods were developed during the years 1999 to 2003 aiming to reduce the complexity of problem by assuming connectors as primary parts.

Due to the complexity of retrieving assembly information in manual approach, researchers motivated to represent assembly information in intelligible manner. The graphical methods of assembly information illustration encouraged towards matrix mode of representation for simplicity of data storage capability. Manual method of preparing and representing the assembly attribute information in a specified format is challenging task with lots of time consumption. As geometric feasibility is an essential qualifying criterion for an assembly sequence, lots of researchers made efforts in retrieving necessary information to perform assembly feasibility testing.

Although stability is a vital qualifying criterion, only a few researchers focused on stability representation methods due to involvement of complex reasoning methods for retrieving relative stability status for parts of the mechanical assembly.

The ever increasing thrust towards minimizing production cost motivated the practitioners towards identifying optimal feasible assembly sequence, which can offer multiple benefits based on the assembly facilities and the employer priorities. Optimal feasible assembly sequence is one of the key factors to achieve economical manufacturing process by reducing the overall production time by 20-30%, overall cost by 40-60% and improves the product quality (Kalpakjian and Schmid, 2008).

Most of researchers used artificial intelligent techniques to achieve optimal assembly sequences for several decades (1993-2015). The introduction of computer aided design (CAD) applications for product design offers great flexibility to retrieve assembly attribute information to perform assembly sequence generation.

Computer aided assembly sequence generation methods with minor limitations have been evolved during 1990’s to till date. These methods need high level user skill to interface with CAD softwares. A brief paradigm of technologies in assembly sequence methods over past five decades are indicated in Figure 1.5. However a detailed review on the literature is presented in the next section.

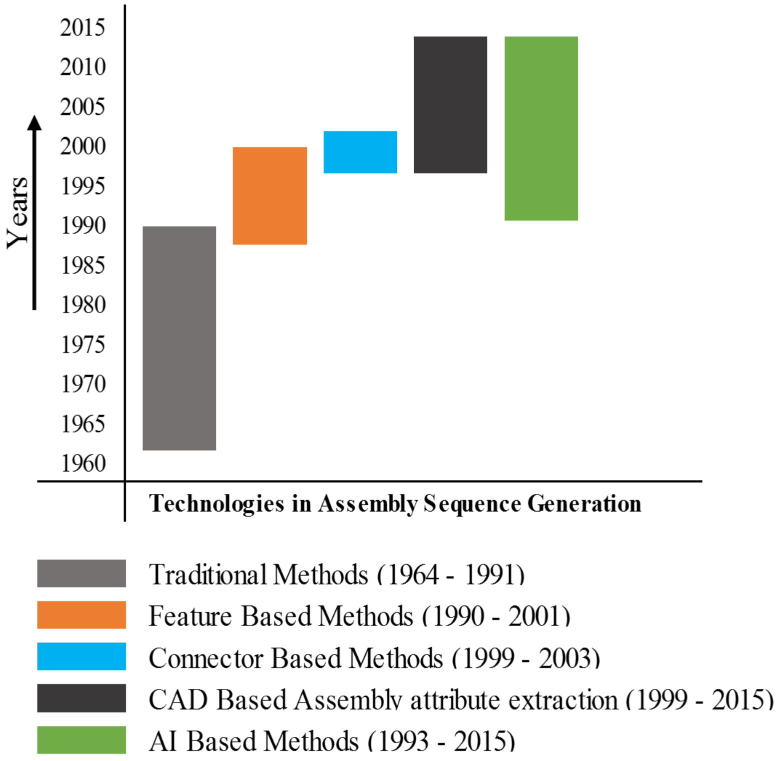


Figure 1.5: Brief paradigm of technologies in assembly sequence generation methods

The objectives of each technology, objectives and their limitations in terms of computational performance, need of skilled user and quality of resulted outcomes have been listed in Table 1.1.

Table 1.1: Objectives and limitations of technologies in assembly sequence planning methods

Technologies in ASG	**Objectives**	**Limitations**
Traditional Methods (1964-1991)	1. To achieve at least one (or) set of all feasible assembly sequence(s).	1. Need highly skilled user intervention. 2. Huge time consumption. 3. Large storage requirements
Feature Based Methods (1990 - 2001)	1. To achieve at least one (or) set of all feasible assembly sequence(s). 2. To reduce efforts at assembly feasibility testing.	1. Need appropriate assembly modelling methodology. 2. Skilled user intervention for assembly mating conditions extraction.
Connector Based Methods (1999 - 2003)	1. To achieve at least one (or) set of all feasible assembly sequence(s). 2. To reduce the complexity by minimizing part count (assuming connectors are joined simultaneously with primary parts)	1. Mechanical feasibility is not considered and leads to in appropriate sequence.
CAD Based Assembly attribute extraction (1999 - 2015)	1. To reduce human intervention in retrieving assembly predicate testing data within less time.	1. CAD interfacing & geometric reasoning skills are required.
AI Based Methods (1993 – 2015)	1. To achieve optimal assembly sequence using assembly attribute data.	1. Only near optimal solutions are assured with stochastic nature. 2. The quality of output is dependent on the input supplied.
CAD Based Assembly Methods (2010 - 2015)	1. To achieve single/optimal assembly sequence from CAD Product.	1. CAD interfacing & geometric reasoning skills are required. 2. User intervention becomes unavoidable to handle products with large part count.

1.6 Broad Objective

The major objective of this dissertation is to develop an efficient computer aided optimal assembly sequence generation methodology. As per survey and analysis of various research literatures in this field of assembly sequence generation recommends that there is certain requirement of some novel and efficient method for solving assembly sequence planning problem. It is also required to be automated with suitable CAD interfacing and should be capable of running without any skilled user intervention. Hence, this work is planned with following key objectives:

1) To develop well-organized CAD interactive methodologies to retrieve assembly attribute information effectively without any skilled user intervention.
2) To develop efficient algorithms for assembly predicate testing considering necessary assembly attributes to ensure practical feasibility.
3) Testing the influence of assembly predicate consideration on computational time and quality of outcomes for different assembly configurations.
4) To develop an efficient and robust automatic method to generate set of all feasible assembly sequences considering all necessary predicates.
5) To enhance the capabilities of the proposed method to find an optimal assembly sequence by considering assembly directional changes, tool changes and assembly energy.

1.7 Research Methodology

The greater objective for any researcher is to develop an efficient method to overcome the above said limitations of existing methods. Considering the above mentioned objectives, the steps for attaining the desired ASG are as follows:

- Review of literature: Literature review has been done related to all types of assembly sequence generation methods to achieve single feasible solution to optimal feasible solutions. The literature review discuss on the techniques implemented to retrieve and store the assembly attributes and their consideration on assembly planning methods. Analysis of the literature survey focused for optimal assembly sequence planning through artificial intelligence techniques and their limitation in achieving the desired objective.

- Computer aided assembly attribute extraction: Literature survey analysis depicts the importance of assembly attributes and their requirement to perform assembly sequence generation. Thus computer aided automatic assembly attribute retrieval methods have been proposed.

- Assembly predicate testing: The resulted assembly sequence output should be practically feasible in the physical environment, for this purpose assembly predicate testing methods have been constructed based on the extracted assembly attribute information.

- Assembly sequence generation method: Unlike the random assembly sequence creation in optimisation approach, constructive part concatenation method has been proposed to build the stable and feasible assembly sequence via intermediate assembly subsets. The intermediate assembly subsets were tested for its practical possibility through assembly predicate testing.

- Optimal Assembly sequence generation: Assembly energy, assembly reorientations and tool changes have been considered to select an optimal assembly sequence from a list of available feasible assembly sequences. To achieve the optimal solution in less span of time, assembly indexing method is implemented at all phases of assembly subset generation.

1.8 Organization of the Thesis

Current *chapter 1(Introduction)* provides brief description about history and evolution of assembly sequence generation methods followed with research motivation. Broad objective of the research and methodology to solve the problem is also clearly stated.

Chapter 2(Literature Review) delivers detailed analysis of cited research literature on the basis of various aspects of assembly sequence generation methods and their elements. Significant work on artificial intelligence based methods are summarized in tabular and graphical format about the type of objectives and input considerations. Finally the objectives of the research work are determined and explained on the basis of literature analysis.

Chapter 3(Methodology) illustrate the product information such as assembly liaisons, mating conditions, connection matrices to perform the assembly sequence generation and methods to extract assembly attributes and assembly predicate testing aiming assembly sequence planning.

Chapter 4(Influence of Assembly Predicate testing) delivers the effect of considering and ignoring an assembly predicate on assembly sequence generation in terms of computational time, performance and quality of resulted outcome. Stability and mechanical feasibility predicates are considered on different assembly configurations and the observations have been discussed for efficient assembly sequence generation.

Chapter 5(Part concatenation Method) proposes method of concatenation to perform assembly subset generation directing towards generation of feasible and stable assembly sequences. Assembly subset generation principles and validation criterions were clearly illustrated. Generating optimal assembly sequences from list of feasible assembly sequences were discussed for specific requirements.

Chapter 6(Direct Optimal Sequence Generation) discusses process of optimal assembly sequence generation for a defined objective function considering assembly directional changes, assembly tool/gripper changes and assembly energy. In this chapter assembly indexing method is proposed to identify similar assembly sets to reduce the redundancy and improve the capability of method to solve large scale assemblies.

Chapter 7(Conclusions) draws the concluding remarks against the stated research objective and further research scope of work.

1.9 Summary

In the current chapter, history of developments in assembly sequence generation methods are presented. Brief evolution of traditional assembly sequence generation methods, optimal assembly sequence generation methods and CAD based methods are discussed in this chapter. Approaches to solve the optimal assembly sequence generation are well discussed. Further need of developing a suitable ASG technique and research methodology involved in development of ASG technique is briefly illustrated.

Chapter 2

REVIEW OF LITERATURE

2.1 Traditional Methods of Assembly Sequence Generation

Traditional assembly sequence methods operate based on user developed instructions/procedures to determine assembly feasibility testing. Most of these procedures are aimed to generate queries to be answered using complex reasoning capabilities. More detailed description on these methods is discussed below.

2.1.1 **Precedence Diagram based Assembly Sequence Generation**

The precedence diagram was initially proposed by Prenting (Prenting and Battaglin, 1964) and later modified by Bullinger which is a graphical illustration of assembly tasks in an efficient manner (Bullinger and Ammer, 1984). In the initial days this diagram was mainly used for assembly line balancing, later stages the bandwidth of this diagram was expanded to computer aided assembly planning.

The precedence diagram looks similar to the assembly network plan with assembly tasks and their dependencies. The tasks and scheduled times are represented in nodes, and the dependences through lines. For a four-part block assembly shown in figure 2.1, assembly precedence diagram with schedule time for each task (assembly operations) and task number is indicated in Figure 2.2.

Figure 2.1: 4-part assembly

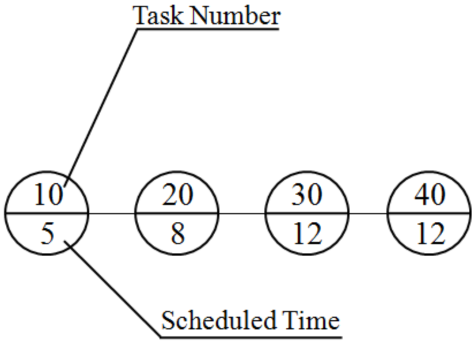


Figure 2.2: Precedence diagram for 4 part assembly

For large products with multiple parts and sub-assemblies, each subassembly of a product has to be represented as a separate rough precedence diagram. This representation demands deep understanding on the product structure, assembly tools and jigs. The major criteria to generate the assembly plan depends based on the principle of rate of utilization of machinery, and secondly the cost of equipment and devices that are used to generate the assembly.

2.1.2 Liaison based of Assembly Sequence Generation (Bourjault method)

Bourjault used the connections between parts to create Liaison diagram structure to represent the assembly surface contacts between mating parts(Bourjault,1984). Liaison diagram is a concept of representing the liaisons between pairs of components to describe the significant relationships between a pair of parts of an assembly. A liaison is a defined contact established between a pair of components. The components in the assembly are represented by nodes and the liaisons between components are represented by hyper-arcs joining the parts. Liaison diagram for the product shown in Figure 2.1 is represented in Figure 2.3.

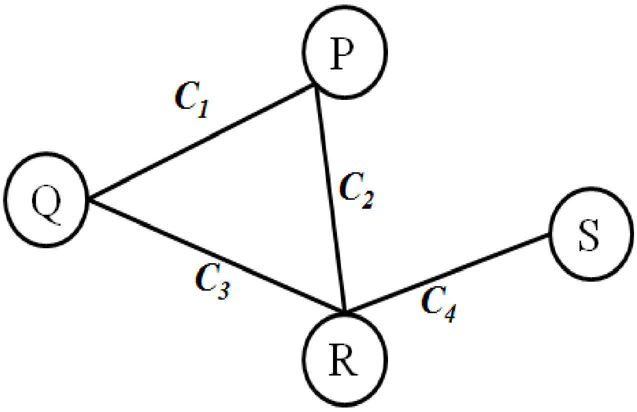


Figure 2.3: Liaison diagram for 4-part assembly

The liaison diagram is used to generate a list of “Yes/No” questions to be addressed by the designer. “Yes” or “No” is determined by the possibility or non-possibility of assembling a component to a product, which is generally dependent on possibility of a clear path without any geometric interference. The Bourjault method of assembly sequence generation is shown in Figure 2.4.

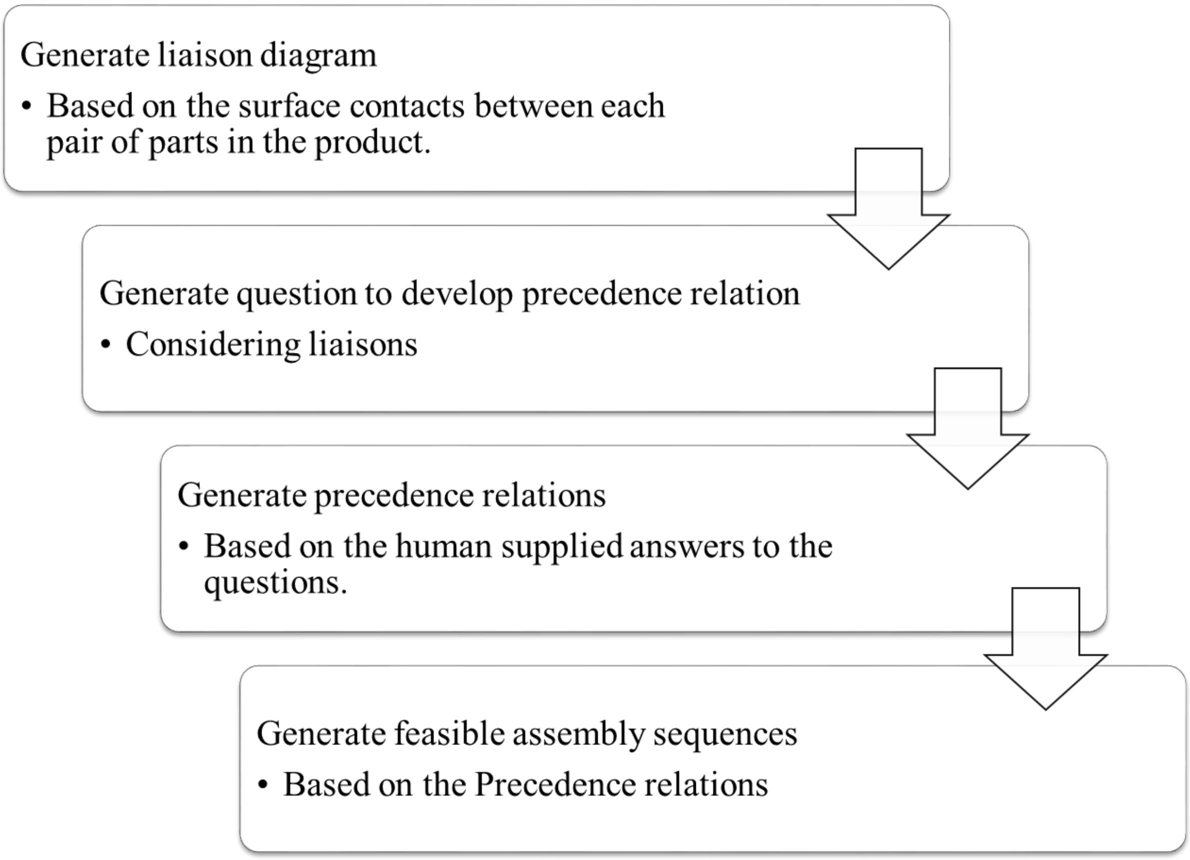


Figure 2.4: Bourjault method of assembly sequence generation

Bourjault method of questions to generate precedence relations

Question:1 Is that true liaison C_i can be established if liaisons (C_j … C_k) have been established?

Question:2 Is that true liaison C_i can be established if liaisons (C_j … C_k) have not been established?

The group liaisons (C_j … C_k) called body of liaisons; the answers "Yes" or "No" to these questions generates precedence relations for an assembly.

The answers to these questions are processed by computer to generate a list of the possible sequences. Computer aid for the method is generated by Lui (Lui, 1988) which generates list of questions based on the number of liaisons, however an expert should answer these questions to generate precedence relations.

2.1.3 Modified Liaison based of Assembly Sequence Generation (De Fazio & Whitney Method)

The method of De Fazio and Whitney is a small variation of Bourjault's method in order to reduce the complexity by reducing the number of queries to generate precedence relations (De Fazio and Whitney, 1987). Like Bourjault method, it requires a liaison diagram of an assembly with establishment conditions. But it is different in the form of questions asked to obtain the precedence relations. Two questions asked to each liaison C_i.

Questions

Question:1 What liaisons must be established prior to establishing C_i?

Question:2 What liaisons must be established until after establishing C_i?

The answers to these questions are directly leads to set of precedence relations for the assembly system, which will be further utilized to generate assembly sequence plans. The increase in number of liaisons raises the number of questions enormously, which need lots of computational time. The user must be highly skilled and should have knowledge on the assembly to answer these questions. The correctness of precedence relations is dependent on the human supplied answers.

2.1.4 Assembly Cut-set Method (Homem de mello)

Homem de mello's assembly cut-set method use liaison diagram and decomposes it into combinations of sub-assemblies at different levels by breaking the liaisons (Sanderson et al., 1990). Each combination is tested for their geometrical feasibility based on human supplied answers considering superset rule and subset rule. The results either can be used to generate precedence relations else can be used to find out stable subassembly sets to perform assembly sequence generation (De Mello and Sanderson, 1991a & 1991b). Assembly cut-set method with all intermediate phases is briefly illustrated in Figure 2.5.

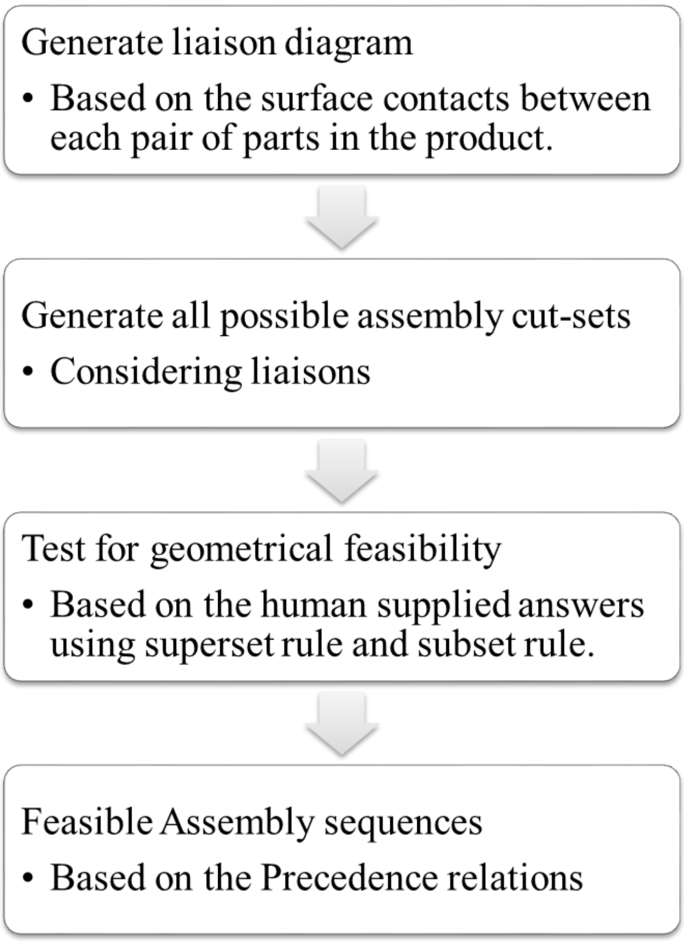


Figure 2.5: Homem de mello method of assembly sequence generation

Assembly cut-set decompositions for the product shown in Figure 2.1 is presented in Figure 2.6, where two part and three part assembly subsets are decomposed.

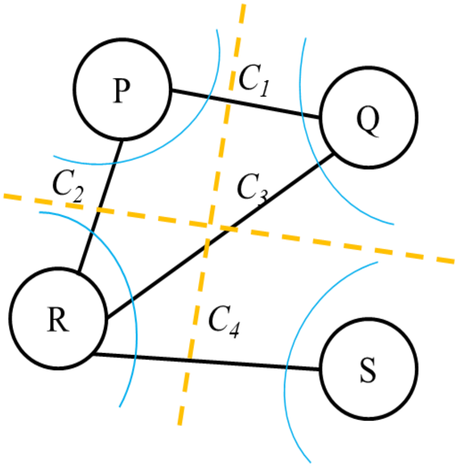


Figure 2.6: Assembly cut-set decomposition for a 4-part assembly

Assembly cut-sets are combination of two assembly subsets SS_i , SS_j where these sets are connected by set of liaisons . The assembly subsets are further used to generate all the necessary queries to find the precedence relations for an assembly. In order to test the geometric feasibility, the queries take below form.

"Can the assembly subset formed by the parts of SS_i be disassembled through a collision free path from assembly subset SS_j?"

Queries are symbolised by the prompt "Q(SS_i; SS_j)?." these queries are answered by a skilled engineers. The superset rule and subset rule are used to test the validity of assembly states. Starting from the largest assembly subsets formed by the complete liaison diagram, the query Q(SSi; SSj)? is checked against all previously obtained precedence relations to assure that the assembly state SSi U SSj is valid (Baldwin et al, 1991). Detailed description of the cut-set method based on the assembly cut-set decomposition shown in Figure 2.6 is listed in Table 2.1.

Table 2.1: Assembly cut-set decomposition for 4-part assembly

Level	**Liaison Graph**	**Assembly Subsets**		**Contact**	**Feasible/Stable**
		SSi	**SSj**		
0	P-Q-R-S	P	Q-R-S		Yes
		Q	P-R-S		Yes
		R	P-Q-S	No Contact	
		S	P-Q-R		Yes
		P-Q	R-S		Yes
		P-R	Q-S	No Contact	
1	Q-R-S	Q	R-S		Yes
		R	Q-S	No Contact	
		S	Q-R		Yes
	P-R-S	P	R-S		Yes
		R	P-S	No Contact	
		S	P-R		Yes
	P-Q-R	P	Q-R		Yes
		Q	P-R		Yes
		R	P-Q		Yes

Two part assembly subsets are assumed to be geometrically feasible, if there exist contact between them. However stability criteria for the two part subsets are not discussed in this method. The major limitation of this method is dependent on skill level of users, who supply answers to generate the precedence relations. This method is highly time consuming and complicated.

2.1.5 Feature Based Methods for Assembly Sequence Generation

An assembly model generated from parts generally controlled by part mating conditions, which are driven by the surface features of solid parts (Lee and Shin, 1990; Kim and Wu, 1990 &1994). While creating an assembly model of a product, it consists multiple standard parts and connectors, which comprises information about the common relations between the assembled parts. Product modelling methodology and representation is not different from part modelling for this purpose. The term product modelling refers to part level and assembly level, hence these strategies store information about parts, products and their relations. In part modelling, there is shift from storing only geometry-oriented information towards more function-oriented information. This is done using part features of assembly models (Shah and Rogers, 1993; Anantha et al., 1996). The functional information is very useful during modelling and planning of parts. Many effective computations can be done on features, instead of on pure geometry of parts.

Assembly illustration and modelling contains physical and spatial relationships between the individual parts at a higher level of generalization than the representation of single parts. Such representation must capable of constructing an assembly from all the given parts, selection of individual parts in the assembly, modifications in the relative positioning of parts, and constrained manipulation of the assembly as a whole. Representations must also support association of form features and mating surfaces involved in kinematic connections, determination of degrees of freedom from the mating conditions, interference checking, and automatic detection of defilements of part envelopes.

Lots of research has been done on extracting part mating features from the assembled product to perform the assembly sequence generation, however the user must be skilled to determine the stability, and geometric feasibility between the parts during the assembly operations (De Fazio et al., 1993; Eng et al., 1999; Van Holland and Bronsvoort, 2000).

Several attempts have been made to retrieve the assembly mating features from various CAD exchanging formats (Prabhakar and Henderson, 1992; Ling and Narayan, 1996; Sung et al, 2001). Gu made an attempt to use graph-based heuristic approach for automatic generation of assembly sequences from a feature-based data base (Gu and Yan, 1995). Mathew used application programme interface (API) to interact with CAD softwares for mating features extraction (Mathew and Rao, 2010a & 2010b). However retrieving such information and using it for assembly sequence validity testing is too complicated and time consuming.

2.1.6 Connector Based Method of Assembly Sequence Generation

Sequence of assembly operations are described based on the attachments used to assemble the parts in the connector based approach. Tseng categorised several types of connectors, those generally used in mechanical assembly process (Tseng and Li, 1999). In their study, connectors worked as assembly components in product depiction and served as concept product building blocks in the design stage. Accordingly, more distinguishing engineering features have been included to reduce the degree of complexity in assembly planning can be effectively reduced.

However assembly modelling and representing for certain connectors like adhesive bonding and pressure fit/press are complicated through this procedure. Furthermore Yin et al, tried to extend the application of connectors considering the reuse context of assembly planning (Yin et al, 2003). However these methods cannot be applicable for all types of possible connectors. The major limitation of these connector based methods is generating the precedence relations between connectors and partial stability between the pair of parts have not been considered; and sometimes it may lead to practically infeasible sequences.

2.2 Assembly Attributes and Computer Aided Extraction

The complexity of assembly sequence generation can be reduced by retrieving the necessary assembly attribute information correctly and efficiently with minimal time. Lots of application program interfaces (API) offer flexibility to interact with CAD software. Linn proposed method to extract assembly liaisons (Linn, 1999) and Mathew used SOLIDWORKS API used to generate liaison relationships from an assembled product (Mathew and Rao, 2010b).

The graphical methods of assembly feasibility representation through connectivity graphs (Shpitalni et al, 1989), interference graphs (Floriani and Nagy, 1989) and assembly constraint graphs (Wolter, 1989) offered flexibility to store the geometric feasibility testing data. Connectivity graph is used to represent the stable assembly and disassembly sequences of a product. The assembly referred here is a robotic assembly where components of the assembly are moved in any single or combination of the six directions along the positive and negative principal axes (+ x, −x, + y, −y, + z, −z). Method for graphical representation of assembly interferences was developed by Floriani in 1989. The concept mainly depends on the interfering surfaces between two mating parts. Each part of assembly is decomposed into multiple surfaces. The contact between the two parts is defined by interference of any two surfaces from those parts (Floriani and Nagy, 1989).

Assembly constraint graphs typically uses huge variety of part trajectories and the constraints which control the part movements in the assembly. This method of representation is more useful in robotic assembly planning. The graph provides information about each part trajectory in the work space (Wolter, 1989).

Non Directional Blocking Graph (NDBG) is proposed by Wilson is more similar to assembly constraint graphs. NDBG created for a pair of parts Part-i and Part-j in contact characterizes the local freedom of each part relative to other part (Wilson and Latombe, 1994).

These graphical methods directed towards representing collision free trajectory possibilities in matrix mode. Mok and Pan proposed methods to extract part boundaries from STEP CAD files, the boundary box upper and lower limit values are further used to generate assembly interference matrices along all the specified directions through collision detection (Mok et al., 2001; Pan et al., 2005). For ease of retrieving the part boundaries effectively from CAD models, polychromatic set based 3D assembly modelling proposed by Li (Li et al., 2012).

Alfadhlani used automatic collision detection between parts during assembly operations from 3D CAD products to perform assembly sequence generation (Alfadhlani et al., 2011). Giri used two dimensional views of part pairs were used to find the geometrical feasibility to ease the process of assembly sequence planning (Giri and Kanthababu, 2015).

Deriving assembly precedence relations between pair of parts from the 3D CAD models is proposed by Su **(Su, 2009).** The parts are treated as polychromatic set to ease the process of retrieving assembly precedence relations and the bounding box coordinates were further used for this purpose. However this configuration cannot assure appropriate results for all assembly configurations.

Most of the research on geometric feasibility testing of parts are limited to principal axes directions. Yu, proposed extended interference matrices for testing assembly feasibility along oblique orientations from CAD models based on the certain assumptions over Local and Global coordinate systems which is far from reality **(Yu et al., 2013 & 2014)**, however performing assembly sequence planning using these matrices is highly time consuming due to large data structures.

Representation of stability using support relations between each pair of parts is proposed by Smith **(Smith et al., 2001)**. Smith proposed representation of assembly connections in matrix format, the connections are categorised into two types; hard and soft. When two parts are connected by physical connectors, the connection is considered as hard and if two components are just maintain their position by surface contact without any physical connection is referred to be soft connection.

The connection data for an assembled product can be represented by an “nxn” matrix. However method to extract such matrix automatically from CAD data is not illustrated. Extracting mating features and assembly connection information from CAD models is helpful to some extent for detecting hard connections.

Generating assembly stability testing methods eases the process of identifying stable subassemblies, which will be useful in parallel assembly systems. Wang proposed a method to identify stable subassemblies using assembly constraints and a set of defined rules through AND/OR method **(Wang and Liu, 2013)**. This method is highly time consuming and needs high skill user intervention for products with large number of parts.

2.3 Artificial Intelligence based Methods

Although the primary goal of assembly sequence generation methods is to obtain at least one stable and feasible assembly sequence by using expertise considering the product information, products with large number of parts results in several feasible assembly sequences, thus very few feasible sequences are considered to select best amongst based on the tool, directional changes and other available facilities. This process is highly time consuming, besides the expert may not come up with the best assembly sequence from set of all available feasible sequences **(Rashid et al., 2012)**. The basic schema of optimization algorithm to solve assembly sequence generation problem is represented in Figure.2.7.

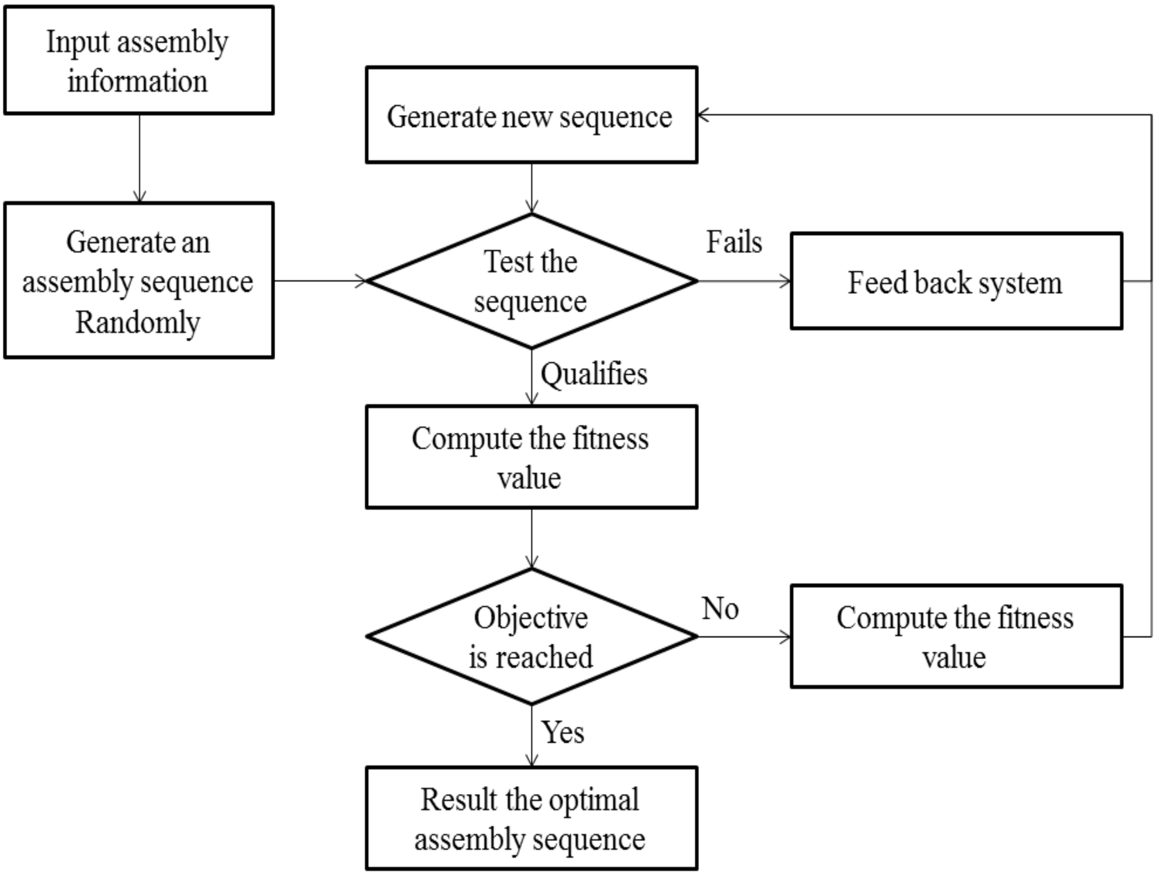


Figure 2.7: Basic schema of optimization algorithm for ASG

The objective of assembly sequence generation is highly dependent on the type of product, availability of machinery and market demand. The industrial need is towards an optimal assembly sequence that can offer multiple benefits. One feasible assembly sequence may offer less assembly time but more assembly cost, the other may be vice versa. The need has motivated the researchers to come up with an optimal assembly sequence that can offer multiple benefits.

Multi objective optimization algorithms are effective in finding optimal assembly sequences, however literature shows that single objective optimization techniques were applied in the initial stage to achieve optimal assembly sequence, hybrid techniques were later used to improve the efficiency of the algorithm.

2.3.1 Genetic Algorithm

Genetic Algorithm is one of the strongest optimization technique evolved from natural biological gene selection process. Classical genetic algorithms were applied wide across to solve science and engineering optimization problems. An adaptive genetic algorithm with continuously varying genetic operations was proposed to generate optimal feasible assembly sequence for primed circuit board assembly by Wong (Wong and Leu, 1993). Part placement and part insertion are only considered as assembly operations for sequence stability. It is proven that adaptive genetic algorithm is superior to classical genetic algorithms.

Genetic algorithm was used to solve assembly sequence generation by Boineville, while most of the assembly planning systems uses the cut–set method to generate the assembly sequences in the earlier days (Bonneville et al., 1995). This method uses a set of valid assembly plans, those are proposed by an expert which is considered as set of initial population to genetic algorithm. The main aim of this method is to select an optimised assembly sequence from set of feasible assembly sequences. The classical genetic algorithm method with two genetic operators knows as cross-over and mutation were considered for populating next level generations. This method has been implemented on a ten part machine component assembly. The proposed method use a set of valid assembly plans instead of populating random assembly sequences, thus the users with assembly knowledge only can use this method.

Genetic algorithm used by Dini to generate optimal assembly sequence to minimize computational time (Dini et al., 1999). The generation of the optimized sequences is performed by taking a fitness function which considers the geometrical constraints, minimising the gripper changes and grouping the similar possible assembly operations along a specified direction. The classical genetic algorithm method with two genetic operators knows as cross over mutation were considered for populating next level generations. The method has been implemented on a seven part machine component assembly. The author also discussed on some open issues like integration with CAD, feasibility testing of sequences, the proposed method has got the usual problem of resulting local optimal solution and premature convergence.

Genetic algorithm initially applied to solve robotic assembly sequence generation problem by Hong and Cho towards minimising the assembly energy. Two types of fitness functions were proposed to calculate assembly energy (Hong and Cho, 1999). Assembly energy is treated as function of assembly stability and assembly directional changes. Like past literature on GA, the classical genetic algorithm with two genetic operators knows as cross over and mutation were considered for populating next level generations. The method has been implemented on a thirteen part automobile alternator assembly. The method states the mutation probability and type of fitness function strongly affect the rate of convergence and computational time. The proposed method has several assumptions in selecting weight factors for assembly energy calculations hence an industrial engineer with in-depth knowledge on the product only can implement this method. Due to classic nature of genetic algorithm, the mutation process was not assured a meaningful assembly sequence caused by gene swapping which further results local optimal solution with huge computational time.

Senin proposed concurrent assembly planning using genetic algorithm to generate all the feasible assembly decompositions starting from the final product and recombining them into possible plans (Senin et al., 2000). The effective ness of algorithm is tested for of reliability and speed in locating the global optimum solution for products with different part count. This approach is more complicated for products with large number of parts.

Robotic assembly sequence generation considering number of changes in assembly directions, gripper changes using genetic algorithm is proposed by Lazzerini (Lazzerini and Marcelloni, **2000).** Replacement of the grippers in regular intervals to grasp the object, number of tool changes to speed up the assembly process is used to compute the fitness value. The classic algorithm with the cross over and mutation genetic operations was considered. This method is implemented on the eleven parts pump assembly and eight part machine assembly with four weight functions have been to achieve optimal sequence from the possible feasible assembly sequences.

Assembly sequence generation with an ordering genetic algorithm (OGA) is proposed by De Lit (De Lit et al., 2001). The algorithm is based on three main concepts. First concept is trace, where transforming any studied assembly sequence into a feasible sequence using `precedence rules', so that an invalid sequence will never be recommended. Second concept is to identify stable subsets, trace is kept all along the sequence of the components membership to a set of parts. Third concept is comparing OGA individuals with each other using a multi-criteria decision aided method. This method is applied on an industrial signalling relay assembly made of 34 parts. Gene mapping relation is defined such that one gene will appear once to avoid redundancy. The proposed algorithm is a meta-heuristic and offer less computational time for products with large number of components. However the proposed method has major limitation such as resulting unstable assembly subsets due to improper consideration of gravitational effects.

Rule based adaptive genetic algorithm is proposed to solve assembly sequence generation by Chen (Chen and Liu, 2001). The rules are used to vary the genetic operator probabilities to enhance the algorithm performance. Unlike the traditional genetic operators, cut-paste is used besides cross over and mutation parameters. Concept of moving wedge is used to determine the geometric feasibility of an assembly operation. A 19-part assembly is considered for implementation, no discussions were made about the assembly stability and premature convergence problem of algorithm.

Smith proposed multi-level genetic algorithm for generating the optimized assembly sequence where two genetic algorithms were used at two different levels low level (level-1) and high level (level-2) (Smith and Liu., 2001). A low level GA generates an assembly sequence based on the genetic operator probability, which is again updated at high level. This multi-level genetic algorithm drastically reduces the search time compared to the conventional GA approaches because during level-1 GA execution, level-2 GA synchronously updates genetic operator probability sequence rather after completion of one level in the conventional GA approaches. In multi-level genetic operator the assembly plan in the level-1 GA will be represented as an assembly tree. In level -1 a local assembly sequence is generated by considering the fitness function as assembly orientation changes with three genetic operators(cross over, mutation and cut paste). This locally generated assembly sequence fitness value will be given as the initial fitness value for the level-2 which uses hybrid mutation and cross over operators for getting the global optimal solution. This method is implemented on 19 component assembly successfully.

A gene-group-based evolution approach is presented to solve optimal assembly sequence planning problem by Guan (Guan et al., 2002). Integrated interference matrix is built to determine geometrical feasibility of assembling a components in a specified sequence. Assembly cost minimization is used to populate the next generation of individuals. Assembly directional changes and tool/gripper changes are considered to compute the assembly cost. The proposed approach is applied to solve a 9-part hypothetical block-assembly and a 19-part practical controller assembly to prove its validity. In this approach the author did not consider the assembly stability. In this approach, each physical connector such as screws are considered as primary parts due to this reason the part count has become increased and thus the search space is exponentially raised.

Enhanced genetic algorithm is proposed by the Greg C. Smith for automated assembly generation (Smith and Smith, 2002). This method considered more genetic parameters like cut-paste, break-and-join and reproduction for improved searching capabilities over the assembly sequence planner by using the traditional genetic algorithm. A 19-part assembly is considered to implement the enhanced GA to minimize the fitness function for minimal assembly direction changes and gripper changes. The authors claimed that the enhanced GA planner is efficient in finding a near-optimal solution more reliably and more quickly. It is observed by the author that cut-paste algorithm is best suited for minimise the assembly directions break-and-join is best suited to maintain the high quality segments of an assembly plan (Smith, 2004).

Del Vallee proposed basic genetic algorithm with cross over and mutation parameters to solve assembly sequence generation problem, for this purpose AND/OR graphs for feasible decompositions were considered (Del Valle et al., 2003). Assembly time for changing the tools was considered for fitness valuation. Assembly stability was ignored in this approach. This method is aimed to find optimal assembly sequences from set of all feasible assembly sequences. However generating such feasible solution space is highly time consuming and complicated. Unlike the previous literature on GA based assembly sequence generation methods, Marian modelled genetic algorithm to generate set of all feasible assembly sequences in the first phase and optimal assembly sequences were found from a set of feasible solutions at second phase (Marian et al., 2003 & 2006). Assembly connections were used to test the stability and assembly constraints were considered to generate the precedence relations between the liaisons. This approach use basic GA to solve the assembly sequence planning problem. No discussions were made about the computational time during generation of set of all feasible sequences and necessity of further GA implementation to find the optimal solution.

The connector based method proposed by Tseng coupled with GA in order to generate optimal assembly sequence **(Tseng et al., 2004).** Object-oriented programming was incorporated with standard template library syntax to combine the connector concept and GA characteristics. The major limitation of these connector based method is generating the precedence relations between connectors and partial stability between the pair of parts have not been considered. Besides it is claimed that Feasibility testing with GA is so complicated and time consuming.

An effective integration approach towards assembly sequence generation and evaluation is proposed by Bai (Bai et al., 2005). This method requires human-computer interactive operations and then switches to operate automatically to create a sub-assembly sequence. Due to this human intervention, number of candidate objects and search space will be reduced. This method uses the time and cost of the assembly as the fitness function for evaluating the optimum assembly sequence. This method uses CAD model to generate the sub assembly sequence to generate automatically. The user intervention during the process demands well knowledge on the assembled product and process.

Huang proposed classical genetic algorithm to solve assembly sequence planning, for this purpose, concept of feature mark was introduced to describe assembly feature and assembly constraints (Huang et al., 2007). Assembly capability function is modelled to test the feasibility of assembly sequence based on the similarity degree between feature mark of similar nature (feature based assembly sequence planning). Reserve strategy for descendants is proposed to reduce the number of useless descendants using superset and subset rules.

Disassembly feasibility information graph (DFIG) has been presented by Hui to describe the product disassembly sequence and operation information (Hui, 2008). The DFIG is further used to solve disassembly sequence planning problem as an optimal path-searching problem. Based on this idea, a genetic algorithm is provided to find out feasible and optimal disassembly solutions efficiently. Assembly reorientations, number of tool/gripper changes and assembly time are considered as cost factors to define the fitness function. However this method is more similar to assembly cut-set method proposed by Homem de mello and the generation of DFIG through feasibility testing is not illustrated.

Multi-criteria assembly sequence planning method using genetic algorithm has been proposed by Choi (Choi et al., 2009). A precedence matrix is considered to determine feasible assembly sequences that satisfy precedence matrix. In the preceding matrix, the relation between pair of components are given such a way which component has to join first and next, thus a feasible assembly sequence has to satisfy this matrix. This method uses two fitness functions one is the minimization of total assembly time and the other one is minimization of the number of reorientations assembly parts. This method selects the weight function randomly for the demonstration hence it demands an engineer with much depth in knowledge on the assembly. Generating precedence data is highly important to test the geometric feasibility to assure the quality of an assembly sequence, however acquiring precedence data from is not discussed anywhere.

Multi-plant assembly sequence generation approach is presented by integrating assembly sequence generation and plant assignment (Tseng et al., 2010). In this method, the components and assembly operations are sequenced considering the assembly constraints and assembly cost objectives. In plant assignment, the components and assembly operations are assigned to the suitable plants to achieve multi-plant cost objectives such as assembly instability cost, assembly accessibility cost, assembly weight effect cost, and general transportation cost, assembly operation cost, assembly tool setup cost.

The feasible assembly sequences are generated using assembly precedence graphs. A genetic algorithm (GA) method is presented to evaluate the multi-plant assembly sequences with an objective of minimizing the total of assembly operational costs and multi-plant costs. However acquiring precedence data to generate precedence matrix is not clearly discussed.

2.3.2 Ant Colony Algorithm

The ant colony algorithm was derived from the observations made on the behaviour of the real ants while collecting the food. Researchers found that the ants always take the shortest path between the food and their nest irrespective of the location of food and presence of obstacles in the path. Many studies on this revealed the capacity of ant is due to the use of a substance named "pheromone" secreted by the ants during their search for the food. The key behaviour of the ant is their bent to follow a path where pheromone is already deposited by the forerunning ants. The simple logic is high pheromone quantity on a trail, the stronger is the bent of the ants to follow that path.

When lots of ants are searching for food around their nest and consider one of them finds food very nearby it. When an ant starts "going and coming" between the food and the nest. The shortest is the path between the nest and the food, an ant can cover the distance within less time. The concentration of pheromone randomly distributed on the ground initially, and increase on the shortest path and this trend will be quickly reinforced by the bent of the ants to follow the trail with the highest concentration of pheromone. The pheromone evaporates in less time and finally only the shortest trail remains. The ants are able to optimize the length of the trail to cover the distance between the food and the nest. This phenomenon transformed to a mathematical tool for solving general and combinatorial optimization problems.

ACAs were found successful in solving many optimization problems. Falli used initially ACA to solve optimal assembly sequence planning problem by considering various assembly constraints (Falli and Dini, 2000). In these approach assembly precedence constraints is considered to determine geometric feasibility of an assembly sequence.

An ant colony algorithm-based approach to solve optimal assembly sequence generation of mechanical products is presented by Wang (Wang et al, 2003 & 2005). For different assemblies, this approach generated diverse amount of ants cooperating each other to find optimal solutions for the objective of minimal assembly reorientations. Assuming the reverse of disassembly sequence is assembly sequence, feasible disassembly operations are derived from search space using disassembly matrix. Though the assembly sequence is geometrically feasible, stability of assembly is not assured and assembly gripper's contribution is not considered to evaluate the fitness value.

The multi-objective disassembly line balancing using an ant colony optimization metaheuristic is presented by McGovern (McGovern and Gupta, 2006). The method generates a sequence that offers multiple benefits such as minimal number of workstations, less idle time. The method is proven in resulting near optimal sequence.

An ant colony optimization to solve assembly sequence planning is presented by Sharma (Sharma et al., 2008). The method generates an optimal sequence that offer minimum assembly cost considering minimum assembly reorientations. Though this method is proven in resulting near optimal sequence, assembly gripper changes and assembly energy were not considered to define the optimality. Interference free matrix is used to test the geometric feasibility.

In order to avoid the most common problem of local optimization with the classical Ant Colony Algorithm (ACA), an improved Ant Colony Algorithm (ACA) was proposed to obtain near optimal assembly sequence by Shi (Shi et al., 2010). In this approach, assembly operation constraints were used to ensure the assembly feasibility. Dynamic ACA parameters were used to improve the convergence of the algorithm.

Yu made efforts to compare existing ACA based assembly sequence generation methods and came up with an improved mini-max ant colony algorithm with five optimisation parameters, including reorientation, parallelism, continuity, stability, and auxiliary stroke to generate optimum feasible assembly sequences (Yu and Wang, 2013a & 2013b). An extended interference matrices (EIM) is used to state the geometrical feasibility of the assembly. Although this method has better performance than the traditional algorithms the major drawback is consideration of physical connectors as regular parts which increases the search space enormously. No discussions were made the about the assembly stability considerations.

Wang solved assembly sequence planning problem in two phases, where phase -1 is dedicated to generate set of all feasible sequences (Feasible solution space), Phase-2 is dedicated to retrieve the optimal solution (Wang et al., 2014). Disassembly Feasibility Information Graph (DFIG) was used to test the geometrical feasibility during the assembly operation. The proposed method is too complicated for products with large part count due to huge feasible solution space. Stability is not considered during the assembly sequence generation.

Lu proposed ant colony algorithm to solve assembly sequence generation and assembly line balancing problem with the help of assembly connection graph and disassembly precedence graph (Lu and Yang, 2015). Assembly time is considered as primary objective which is evaluated by taking the time for changing assembly directions, changing assembly tools, and the time for moving the heavy parts in the workstation into consideration. However the generation of dis assembly precedence graph for a product with large number of parts involved with high skilled user intervention.

2.3.3 Neural Networks

Artificial neural networks (ANN) imitate the human biological nervous system for optimal decision making. ANN works as signal processing system composed of simple inter connect elements called nodes and connection is referred to the links. These are used for parallel distributed processing in order to solve a mathematical optimisation problem. ANN offers minimal computational time by its parallel solving capabilities.

Chen initially proposed neural network based assembly sequence generation by using the knowledge-database developed through the past experiences (Chen and Pao, 1993). The method offers flexibility to a designer for checking similar assembly configuration existed in the design cluster memory. Only two types of connectors were used to simplify the process for linear assembly sequence planning. Assembly stability was assumed at all stages of assembly operations, part mating conditions are used to test geometric feasibility.

Hong proposed a neural network based assembly sequence generation to achieve objective of minimum assembly energy (Hong and Cho, 1993 & 1995). Assembly energy is computed based on the various elements those influence cost and time such as number of directional changes, instability. Precedence constraints and liaison data were considered to generate the precedence relations to test the feasibility of an assembly sequence.

Sinanoglu presented a neural network based optimal assembly sequence generation method (Sinanoglu and Riza Börklü, 2005). Energy required to place a part is considered as optimal criteria. Assembly connections and assembly mating conditions were used to generate interference free matrices, which are further used to detect the geometric feasibility. Permanent stability between the parts were considered to state the assembly stability and sub assembly identification. Assembly directional changes and tool/gripper changes were not considered in this approach.

Chen proposed a three-stage integrated approach for assembly sequence generation using neural networks (Chen et. at, 2008). In this method explosion graphs were created from the liaison model and precedence test was done on these graphs for feasibility testing. This method is more similar to the assembly cut-set method. Generating the explosion graphs demands high skilled intervention at each stage to read the assembly mating conditions. Assembly cost function is created based on the assembly infeasibility/ feasibility.

2.3.4 Simulated Annealing Algorithm

Simulated annealing optimization algorithm is adopted from the mechanical annealing process in which metal is heated to high temperatures and allow the metal to cool in a control manner to achieve minimum global energy. Metropolis initially created an algorithm to simulate the annealing process in 1953. The algorithm simulates a small random displacement of an atom that results in a change in energy. If the change in energy is negative, the energy state of the new configuration is lower and vice versa, if the change in energy is positive, the new configuration has a higher energy state. Hence the algorithm can be tuned for any maximization/ minimization objective.

Milner proposed simulated annealing algorithm to generate probable least-cost assembly sequence (Milner et al., 1994). In this method, assembly cost was considered as the energy function associated with an assembly sequence. The least cost sequences found by SA were often not of good engineering quality because the cost function used in this method does not consider the cost and time related to number of assembly reorientations and re-fixturing for an assembly sequence. This method is applied on the 6 parts pen assembly by considering the only energy parameter. In this method, random assembly sequence is considered by Diamond method which is manually done by the engineer.

A multi-criteria assembly sequence generation using simulated annealing was proposed by the Motavalli (Motavalli and Islam, 1997). In this paper two criteria(s) were considered to define the objective function, the first is assembly time and the second is number of assembly direction change. These two criteria(s) were combined using multi-attribute utility theory to derive a single objective function to evaluate fitness value. This objective functions was subjected to liaison precedence criteria. The developed objective function was utilized further in SA to generate the optimum feasible assembly sequence. A 20-component assembly was considered for obtaining the optimum assembly sequences. The limitation of the proposed method is generation of precedence relations and also the method demands a feasible assembly sequence as input.

Generation of optimal robotic assembly sequences using a simulated annealing presented by Hong (Hong and Cho, 1997 & 1999). Minimizing the assembly cost is considered as objective function underspecified assembly constraints. Assembly operations, assembly motions, and assembly direction changes were used to compute the assembly cost. The assembly with low cost was found by minimizing the energy function iteratively by SA method. In this method a ten part electrical relay assembly is considered. The performance is evaluated by comparing the results with that of neural network-based approach. The proposed method did not consider the effect of gripper changes on assembly cost.

Lee proposed an improved method of assembly sequence generation method for automatic assembly systems. This paper proposed a multi-echelon simulated annealing method to generate optimal assembly sequence based on the cost performance (Lee and Gemmill, 2001). In this method a simple enumerate algorithm is used to achieve the optimised assembly sequence in three echelons. Echelon 1 sets the number of assembly stations required to perform assembly operations, echelon 2 algorithm found the optimum assembly station configuration and echelon 3 is dedicated for optimal liaison assignments for a given optimum assembly station.

The above method is performed on two models, one model is having five liaisons, three assembly stations and the other model is having the eight liaisons and six assembly stations. The limitations of this method is, it does not suite for the product with more number of parts and also no specific rules were defined for assembly stability.

2.3.5 Particle Swarm Optimization

Particle swarm optimization (PSO) is a robust stochastic optimization technique based on the movement and intelligence of swarms. PSO was inspired by social behaviour of bird flocking or fish schooling, which is population based methodology. It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution. Each particle is treated as a point in the search space which adjusts its “moving” according to its own flying experience as well as the flying experience of other particles. Each particle keeps track of its position in the solution space with respect to the best solution (fitness) that has achieved so far by that particle. This value is called personal best. Another best value that is tracked by the PSO, which is the best value obtained so far by any particle in the neighborhood of that particle. This value is called global best, the basic concept of PSO lies in accelerating each particle from particle best towards global best, with a random weighted acceleration at each time.

Yu proposed particle swarm optimization to solve the assembly sequence generation problem efficiently in order to overcome the limitations of classical algorithms (YU et at, 2010). To assure the geometrical feasibility of assembly sequence interference matrices were used. In this method, assembly connection matrix and support matrix were used to result stable assembly sequences. The occurrence of assembly directional changes was considered to define the objective function. However the influence of gripper changes was considered in this approach and the search space is too large for treating physical connector as primary part.

A discrete particle swarm optimization (DPSO) algorithm is proposed to solve the assembly sequence generation problem by Lv (Lv and Lu, 2010). A superior coding method for updating position and velocity of particles to enhance the performance of particle is proposed in order to achieve the minimal computational time. Investigation is made on control parameters setting to improve performance. Physical connectors were considered as primary parts, which increases the search space in this approach. Assembly directional changes are only considered for the fitness evaluation.

Wang proposed an advanced PSO namely chaotic particle swarm optimisation to generate near optimal assembly sequences by considering six assembly process constraints (Wang and Liu, 2010)**.** Interference matrices and precedence matrix were used to ensure the geometrical feasibility, Stability matrix and assembly connection matrix were used to state the assembly stability. Tool matrix was used to determine the number of tool changes. A combined objective is defined considering number of directional and tool changes.

Although the proposed method is proven effective when compared with standard PSO and simulated annealing, the supplied input is highly redundant in nature to test the practical possibility and thus demands high computational time.

Tseng proposed a green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization (PSO) method(Tseng et. al, 2011). In this method instead of proposing two different algorithms for assembly and disassembly purpose, a single closed loop assembly planning is suggested. Assembly and disassembly graphs were created to generate assembly and disassembly precedence data, which was used to test the geometrical feasibility. Various costs elements such as operational cost, stability cost and tool setup cost to perform assembly and disassembly operations were considered to define the objective function. However the concept of sub assembly detection for disassembly planning is not considered in this approach, due to dissimilar solutions for assembly and disassembly the method demands high computational time.

2.3.6 Artificial Immune Systems (AIS)

Artificial immune systems are a class of computationally intelligent systems inspired by the principles and processes of the immunity system. The algorithms typically exploit the immunity system's characteristics of learning and memory to solve a problem. The immunity system refers to the procedure of immune cells to resist infection from microorganisms or viruses, especially as a result of antibody formation. Antibodies were created when B-cells react with the Antigen to fight against the virus or organisms. Bone marrow algorithms, Negative selection algorithm and Clonal selection algorithm are the most used immunological theories to solve reasoning and optimisation problems.

Cao presented artificial immune system inspired from vertebrate immune system for generating optimal feasible assembly sequences (Cao and Xiao, 2007). Clonal selection method is used to generate the next level population by considering a fitness function. Assembly directional changes and number of gripper changes were used to define the fitness function. Assembly precedence relations were used to test the feasibility of an assembly sequence. The proposed method was implemented on various assemblies and compared the performance with classical genetic algorithm. Stability of assembly has not been considered in his approach. The proposed method demand more computational time due to the consideration of the physical connectors as the primary parts.

Chang presented artificial immune system to solve assembly sequence problem within less computational time to overcome the demerits of genetic algorithm memetic algorithms (Chang et al., 2009**)**. Clonal selection was chosen along with the gene reorganisation for effective reproduction process. Assembly directional changes were considered to evaluate the energy function. Assembly precedence constraints were considered to test the geometrical feasibility. It is claimed that the performance of the proposed method is superior when compared to guided genetic algorithms and memetic algorithms in solving assembly sequence planning problems.

Biswal presented clonal selection based artificial immune system to solve the assembly sequence generation problem (Biswal et al., 2013). Affinity maturation principles were coupled with clonal selection to find out near optimal solution from the list possible assembly sequences. Assembly contact information and assembly connection matrices were used to test the geometrical feasibility and assembly instability. A seven part assembly is considered to illustrate the method efficiently in which assembly energy is treated as an objective function.

2.3.7 Memetic Algorithms (MA)

Memetic algorithms are the recent evolutionary computational technique using complex structures such as the combination of memes and simple agents. The evolutionary interactions of these agents lead to complex problem-solving intelligence. Memetic algorithms are now widely used as an interaction of evolutionary or any population-based approach with local improvement or individual learning procedures for solving optimisation problems with huge search space. Tseng proposed memetic algorithms with guided local search to solve assembly sequence planning problem efficiently (Tseng et al., 2007). Connector's precedence matrix has been developed based on the physical connectors used to assemble the product, which is further used to test the geometrical stability. The fitness value is calculated based on the number of directional changes and tool changes. The performance of proposed method is found to be superior when compared with guided GA.

Gao prosed memetic algorithm to generate optimal feasible assembly sequence **(**Gao et al., 2010**).** The proposed method used interference matrices along six principal axes directions to test the geometrical feasibility of an assembly sequence. In this method geometrical and contact relations only used to test the practical possibility of an assembly by ignoring the assembly stability. The objective function is defined for infeasible and feasible assembly sequence by considering only number of directional changes, which consumes lots of computational time while calculating the fitness value. Although the performance of the proposed method is better than heuristic search methods, the objective function has not been considered number of tool changes for more realistic approach.

Zeng proposed a memetic algorithm considering connector based precedence relations to generate optimal feasible assembly sequences **(**Zeng et al., 2011**).** The fitness value is calculated based on the number of directional changes and tool changes. The work presented in this article is more similar to the Tseng's approach of memetic algorithm.

2.3.8 Breakout Local Search Algorithm(BLSA)

Breakout local search (BLS) algorithm is a recent development to solve complex engineering optimisation problems. BLS is applied to solve assembly sequence planning problem by Ghandi **(**Ghandi and Masehian, 2015). In this approach assembly interference matrices were used to determine the geometrical feasibility of an assembly sequence. Assembly directional changes were considered to evaluate the fitness function. The outcomes of the proposed method were compared with many classical optimisation algorithms for different assemblies.

2.3.9 Frog Leaping Algorithm

Shuffled Frog Leaping Algorithm is a metaheuristic optimization method to identify the optimal solution. The process imitates natural searching technique for the food by frog community in the real time environment. This algorithm distributes population into numerous subpopulations, and the evolution of memes is regulated by the exchange of global information within the subpopulations. The population consists of many frogs, grouping behaviour of frogs are simulated by using a grouping operator for a sub population, which is called as memeplex. The frogs jump from the local optimum solution to the global optimum solution through the alternating memetic evolution and global shuffling.

Guo proposed shuffling frog leap algorithm for optimising assembly sequence generation in radioactive environment **(**Guo et al., 2015**).** Interference matrices along six principal axes were used to test the geometrical feasibility of an assembly sequence. Assembly directional changes and gripper changes were considered to evaluate the fitness function. A robotic gripper assembly was considered for the implementation and the results were proven that the performance of the proposed method is superior to classical genetic algorithm and particle swarm optimisation algorithms.

2.3.10 Gravitational Search Algorithm (GSA)

Gravitational search algorithm (GSA) is based on the law of gravity and the motion of mass interactions. The GS algorithm uses the theory of Newtonian physics and the collections of masses are treated as search agents. In GSA, there is an isolated system of masses, using the gravitational force of every mass in the system can observe the situation of other masses. The gravitational force can be considered as a way of transferring information between different masses. In GSA, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by a gravity force, and this force causes a movement of all objects globally towards the objects with heavier masses. The heavy masses correspond to best solutions of the problem. The position of the agent corresponds to a solution of the problem, and its mass is determined using a fitness function. By lapse of time, masses are attracted by the heaviest mass, which would ideally present an optimum solution in the search space.

Ibrahim proposed a multi-level gravitational search algorithm to solve assembly sequence generation efficiently using principles of Newton's law of gravity (Ibrahim et al., 2015). The law of motion, and the rules stated for the assembly precedence diagram that makes each assembly component of each individual solution occur only once based on precedence constraints.

To evaluate the fitness function, overall assembly time was considered based on the assembly setup time and assembly operation time. The result of proposed method is compared with various classical optimisation algorithms such as genetic algorithm, simulated annealing and binary particle swarm optimisation. How-ever the generation of precedence diagram to test assembly feasibility and stability is involved highly skilled user intervention.

2.3.11 Imperialist Competitive Algorithm (ICA)

Zhou used Imperialist competitive algorithm to generate optimal feasible assembly sequences (Zhou et al., 2013). The dis-assembly matrix along all principal axes directions used to test the geometrical feasibly of an assembly sequence. Number of assembly directional changes and gripper changes were considered to define the objective function. The proposed algorithm is found better in performance when compared with the genetic and particle swarm optimisation algorithms.

2.3.12 Hybrid Algorithms

Chen proposed a hybrid genetic algorithm method to solve assembly sequence generation problem with multiple objectives (Chen et al., 2002). The contribution of assembly cycle time, workload smoothness, tool change and complexity of the assembly sequence was used to evaluate the objective function. Assembly precedence graph was used as an input to state the geometrical feasibility of the resulted assembly sequence. The method was successfully employed on a product made of six part machine components.

Shan proposed a hybridized simulated genetic algorithm to obtain an optimum feasible assembly sequence (Shan et al., 2006). The random initial population in GA leads premature convergence and result in local optimal solution, to address this issue simulating annealing is hybridized to refresh the initial population for improved convergence rate. Assembly directional and tool changes were considered to define the objective function. Assembly precedence relations were used to test the geometrical feasibility of an assembly sequence.

Ning proposed a hybrid algorithm to solve assembly sequence generation problem by combining ant colony algorithm with genetic algorithm to upgrade the quality of the solution and reduce the probability of local optimum solution **(**Ning and Gu, 2007**).** Ant colony algorithm was used to generate set of all feasible assembly sequences, genetic algorithm was further used to find the optimal solution. The proposed method considered assembly directional changes and tool changes to define the objective function. The resulted outcomes found to be encouraging in terms of computational performance.

Shuang presented particle swarm optimisation hybridized with ant colony optimisation to solve optimal assembly sequence generation in less computational time (Shuang et al., 2008**).** The proposed method was used on a micro-robot assembly, where geometrical conditions and visibility conditions were considered to generate the characteristic matrices for geometrical feasibility testing. The method considers only assembly reorientations to define the objective function. The computational time of the proposed method is observed to be better when compared with basic ant colony optimisation and genetic algorithm.

Tseng proposed an evolutionary multi-objective algorithms and grouping genetic algorithms together for integrating assembly sequence generation and assembly line balancing to find out Pareto-optimal solutions effectively with greater flexibility to change the assembly system design **(**Tseng et al., 2008**).** Precedence graph along with liaison data was used to test the geometric feasibility of an assembly sequence and overall assembly time was considered as objective function.

The common shortcomings of genetic algorithms such as premature convergence, low searching efficiency and shortcomings of simulated annealing algorithm such as generation of infeasible assembly solutions motivated the author to come up with a hybridized algorithm with better search capabilities and robustness.

Shana proposed a hybridized genetic simulated annealing algorithm to solve the assembly sequence generation efficiently **(**Shan et al., 2009**)**. By using this hybrid technique, the degree of dependence on the initial assembly sequence is reduced. By merging genetic algorithm and simulated annealing, the efficiency of searching and the quality of solution was improved. Interference matrices along six directions were used to test geometric feasibility of the assembly sequence. Assembly directions and tool changes were considered for evaluation of the fitness function.

Zhou proposed a novel hybrid algorithm to solve assembly sequence generation problem by combining bacterial chemotaxis with genetic algorithm to upgrade the quality of the solution and reduce the probability of local optimum solution (Zhou et al., 2011**).** The proposed method considers assembly directional changes and tool changes to define the objective function. The resulted outcomes found to be encouraging in terms of computational performance when compared with genetic algorithm and fuzzy logic genetic algorithm.

Li proposed an improved discrete particle swarm optimization to solve assembly sequence planning **(**Li et al., 2013**).** Although this method result global optimum solution the convergence rate was low compared to standard discrete particle swarm optimization, hence modified evolutionary direction operator was used to accelerate the convergence rate. Interference matrices along the six principal axes were used to test the geometrical feasibility. To evaluate the fitness function directional changes, tool changes and assembly operation type changes were considered. Physical connectors were considered as primary parts, which increases the search space enormously.

Zhang proposed combined artificial immune system with particle swarm optimisation to achieve optimal feasible assembly sequence **(**Zhang et at., 2014**).** In this method coherence data along with the interferences matrices were considered to test the geometrical feasibility. Assembly directional changes were only considered to evaluate the fitness function. The performance of the proposed method was found to be better than the artificial immune system and particle swarm optimisation.

2.3.13 Knowledge Based System (KBS)

The artificial intelligent methods to solve the assembly sequence planning result a near optimal solution and moreover the procedure is stochastic in nature hence correct ness of solution is not promised always. So, few researchers attempted to integrate certain heuristics rules with assembly attributes to generate optimal assembly sequences without using any artificial intelligent technique.

Huang proposed a knowledge-base assembly sequence generation method to generate a feasible assembly sequence **(**Huang and Lee, 1991**).** The method used predicate calculus which was simple and powerful knowledge representation to perform assembly sequence planning. The knowledge data base was created to store the knowledge about assembly structure, precedence constraints and resource constraints. A graph search mechanism was used to locate the optimal assembly plan using the knowledge data base. Although this method assures optimum solution, it consumes high computational time hence it cannot applied for the products with large parts.

Zha proposed a knowledge based assembly sequence planning **(**Zha et al., 1998**)**. A knowledge data base was prepared with liaison model, topological and geometrical constraints, stability and security constraints, partial precedence constraints and assembly cost elements. In order to achieve an optimum assembly sequence a quantitative criteria was formed by considering the assembly time, cost and part priority index.

Design for assembly and motion time measurement analysis were made to estimate assembly time and cost of the product. Chain rule, Commutative property and Distributive property were used to filter the feasible assembly sequences. These rules are very much similar to the super set and sub set rules. However this method cannot be implemented for the products with large number of parts.

Dong proposed connection-semantics-based assembly tree to use as knowledge database to generate optimal feasible assembly sequences **(**Dong et al., 2007**).** This approach considers consider both geometric information and non-geometric knowledge from the CAD model.

In this research, the application showed that the knowledge-based approach has ability to reduce the computational complexity drastically and to obtain feasible and practical plans for small products, however generation of knowledge database need highly skilled user interface for products with larger part count.

Hsu developed a knowledge-based engineering system to support engineers in generating a near-optimal feasible assembly sequence **(**Hsu et al., 2011**).** A three-stage optimization methodology with some working rules was implemented to solve ASG. Explosion graph of assembly model was created at the first stage by using transforming rules along with some assembly attributes. Geometric constraints and assembly precedence diagrams were used to generate a complete relational model graph and incidence matrix at the second stage for geometrical feasibility testing of an assembly sequence.

In the third stage, a back-propagation neural network was developed and integrated with the Siemens NX system through application programme interface to extract the component inertial properties and assembly mating features to perform optimality testing. However generating knowledge based system demands explicit knowledge on the assembly and skills on CAD interfacing.

Kashkoush proposed a novel knowledge-based mixed-integer programming **(**Kashkoush and ElMaraghy, 2015**)** to generate optimal assembly sequence of a given product based on available feasible assembly sequence data of identical products. The proposed mathematical model finds a near optimal assembly sequence tree for an existing product family based on the assembly sequence trees of individual product family members. In this method interferences matrices and precedence data were used to test geometrical feasibility, to test the assembly stability assembly support matrix was considered as input. The optimality criteria was considered based on number of directional changes and tool changes.

2.4 CAD based Methods for ASG

Hoffman presented a method to retrieve mating conditions between pair of objects modelled in Constructive Solid Geometry (CSG) representations with relative position (Hoffman, 1989**)**. These mating conditions were further used to discover a collision free path for extricating one object from the other by assuming that the reverse of disassembly procedure results an assembly sequence. A CSG model is supplied to the system as an arrangement of primitive geometries, where each primitive specifies the primitive type (cube, cylinder or sphere,). Parameters of the primitive were used to simplify the geometrical feasibility testing. This method can be only implemented to generate a single feasible assembly sequence and cannot be applied for optimal assembly sequence generation and further the assembly sequence may not assure stable assembly sequence.

Khosla developed a solid modelling software named Noodles Solid Modeller for purpose of solid and assembly modelling and to perform assembly sequence generation **(**Khosla and Mattikali, 1989**).** To represent assembled product, kinematic constraints were used, which were helpful to determine feasible assembly sequences automatically.

AND/OR graph was used to decompose the product in to possible assembly subsets for a product and geometric reasoning system was applied to get feasible assembly subsets. No detailed descriptions were made on reasoning system capabilities to test geometric feasibility. The proposed method can work for only generating a non-optimal feasible assembly sequence automatically with lots of computational time. This method cannot be adopted to solve optimal assembly sequence generation problem.

Liu presented Part and Assembly Description Language-II for the purpose for finding mating features from geometric boundary models of assembly components **(**Liu and Popplestone, 1989**)**. Solving assembly sequence planning problem using the mating features has the benefit of providing more coherent data for special purpose descriptions.

The major advantage with the above method was direct accessing of Solid modeller data from a design database for geometric analysis and spatial reasoning. This method can be only implemented to generate a single feasible assembly sequence and cannot be applied for optimal assembly sequence generation and further the assembly sequence may not be stable.

Hoffman presented a technique to retrieve mating conditions between pair of objects modelled in B-rep solid representations with the relative position **(**Hoffman, 1990**)**. These mating conditions were further can be used for collision free path detection between a pair of parts by assuming that the reverse of this disassembly procedure creates an assembly sequence. This paper more focuses on the freedom of motion module of the BRAEN system proposed by same author, and introduced reuse theorems that can be applied to deduce freedoms of motion for new configurations.

Kanai proposed a computer-aided Assembly Sequence Planning and Evaluation system to generate optimal feasible assembly sequences **(**Kanai et al., 1996**).** The system automatically finds set of all geometrically feasible assembly sequences by decomposing solid models of a product.

Least operating time was considered as optimality criteria to filter the solution by considering the Methods Time Measurement (MTM) and design for assembly concepts. Assembly model data along with assembly constraints were used to detect the geometric feasibility and stability of an assembly sequence.

Ciszak proposed computer aided automatic assembly attribute extraction to perform optimal assembly sequence planning using certain graph theory and heuristic methods of multi-criterion optimisation strategy **(**Ciszak, 2012**).** This method considered assembly collation matrix in all principal axes directions to ensure geometric feasibility and assembly connection matrix for assembly stability.

Vigano proposed a novel method for assembly sequence generation from CAD model information **(**Vigano and Gomez, 2012 & 2013**).** He made efforts to extract liaison graph and assembly mating conditions from a 3D CAD model and analysed the model to test the geometric feasibility.

This method was capable of finding at least a feasible assembly sequence for the product. It can also propose a near optimal solution from a list of extracted feasible assembly sequences in automatic mode from a 3D CAD model. However, the major limitation of the approach was actually, the great number of impossible sequences that were generated thus a skilled user intervention is required.

Ou proposed relationship matrix based automatic assembly sequence generation from a CAD model (Ou and Xu, 2013**).** The proposed method could able to collect assembly constraint data from CAD models for assembly sequence generation, CAD constraints were used to generate assembly sequences. The generated assembly sequences were validated through interference and stability analysis by using assembly relation matrix for a CAD Product. The system is capable of creating a set of feasible assembly sequences for an operator to evaluate. This method does not consider aspects of assembly cost/time.

Hadj proposed an automatic approach for assembly sequence generation from an assembled product modelled in SolidWorks **(**Hadj et al., 2015**).** The proposed mechanism initially extracts all the necessary assembly attributes from the CAD models using application program interface. This data will be further used to carry out collision analysis in order to determine collision free path.

The proposed method classifies the possible base components and generates only feasible assembly sequence instances. The proposed method was successful to determine a feasible assembly sequence but due to large number of alternate feasible solutions, it cannot be implemented for optimal assembly sequence generation.

2.5 Review Analysis and Outcomes

Numerous literature have been considered on implementation of soft computing techniques to address ASG problem, in which 30% of cited research articles used genetic algorithms. Next to genetic algorithms, ACA and hybrid algorithms have took the next priority by each 13%. Researchers have also implemented neural networks, immune algorithms, memetic algorithms and evolutionary algorithms and so on to solve the problem efficiently. Numbers of articles based on the optimization algorithms are represented in figure 2.8.

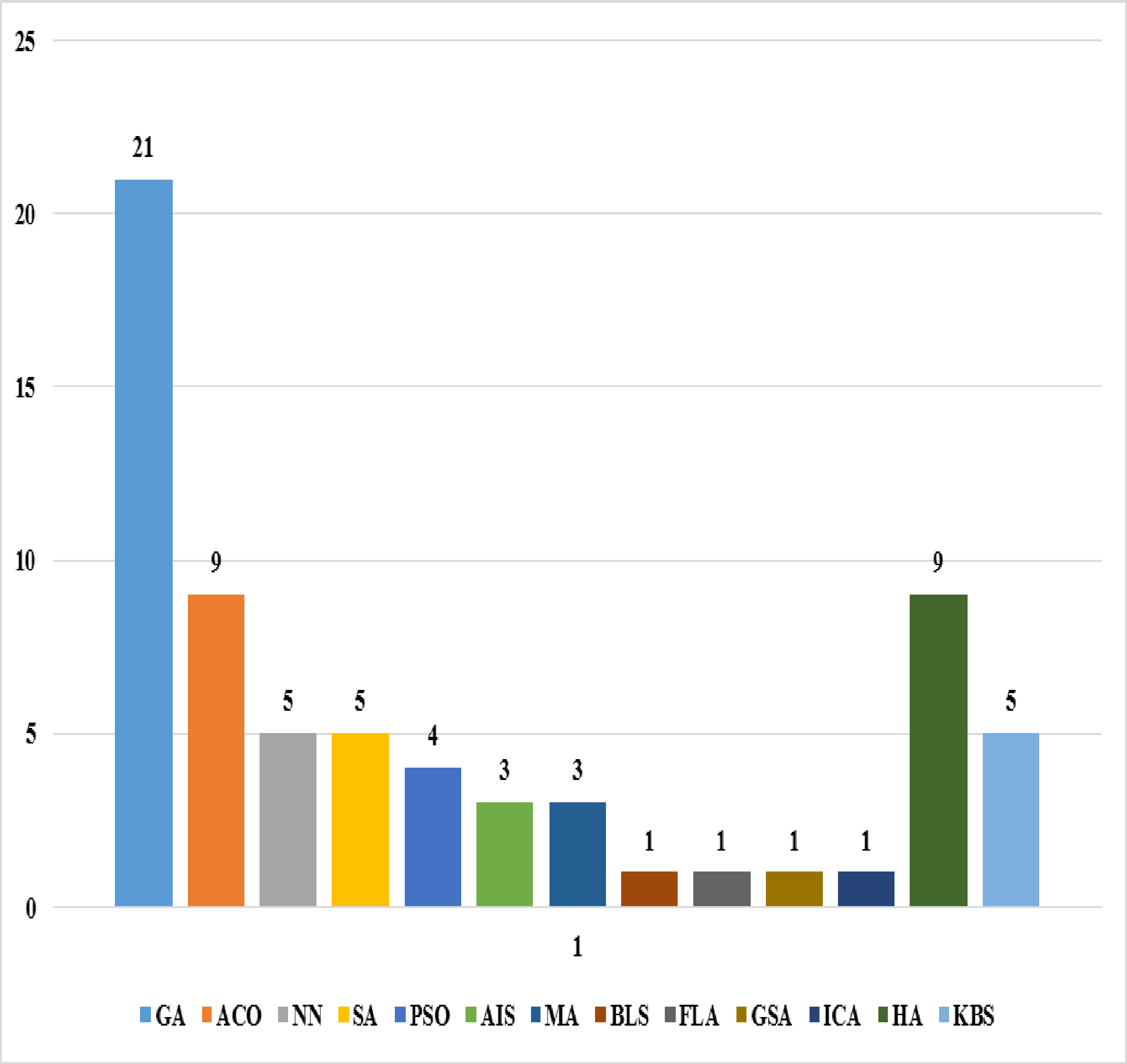


Figure 2.8: Usage of optimisation algorithms for solving ASG problem

Assembly sequence generation methods considered different combination of input parameters to test an assembly sequence for practical feasibility of assembly operations. Few of the Geometric and topology details, Part contact details, Precedence details, Assembly interference matrix, Assembly connection details, Mating conditions and Stability matrix are mostly considered as in input in the cited literature. Input consideration in the cited research literature is listed in table 2.2.

Table 2.2 Input considerations in the cited literature

Algorithm	**Reference**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
GA	Wong and Leu, 1993	X	X			X		
	Bonneville et al., 1995	X	X					
	Hong and Cho, 1999	X	X			X		
	Dini et al., 1999	X	X			X		
	Lazzerini and Marcelloni, 2000	X	X					
	Smith et al., 2001	X	X					
	Chen and Liu, 2001	X	X		X			
	De Lit et.al, 2001		X	X				X
	Smith and Smith, 2002		X		X			X
	Smith, 2004		X		X			X
	Guan et al., 2002	X			X			
	Marian et al., 2003& 2006	X	X	X				
	Del Valle et at., 2003		X	X				
	Tseng et al., 2004		X	X		X		
	Bai et al., 2005	X	X					
	Huang et al., 2007						X	
	Hui et al., 2008		X		X			
	Choi et al., 2009		X	X				
	Tseng et al., 2010		X	X				
ACO	Falli and Dini, 2000		X	X				
	Wang et al, 2003& 2005		X	X				
	McGovern and Gupta, 2006		X	X				
	Sharma et al., 2008		X		X			
	Shi et al., 2010	X		X				
	Yu and Wang, 2013			X	X			
	Wang et al., 2014		X	X				
	Lu, C. and Yang, Z., 2015	X	X	X				
NN	Chen and Pao, 1993		X	X				
	Hong and Cho, 1993 & 1995		X	X				
	Sinanoglu and Riza Börklü, 2005		X	X				
	Chen et al, 2008	X		X				X
SA	Milner et al., 1994		X	X				
	Motavalli and Islam, 1997		X	X				
	Hong and Cho, 1997 &1999		X	X				
	Lee and Gemmill, 2001		X	X				
PSO	YU et al., 2010		X		X			
	Lv and Lu, 2010		X		X			
	Wang and Liu, 2010		X		X			X
	Tseng et. al, 2011		X	X				

Table 2.2 Input considered in the cited literature (contd..)

AIS	Cao and Xiao, 2006		X	X				
	Chang et al., 2009		X	X				
	Biswal et al., 2013		X		X			
MA	Tseng et al., 2007		X	X				X
	Gao et al., 2010	X			X			
	Zeng et al., 2011		X	X				
BLS	Ghandi and Masehian, 2015	X		X	X			
FLA	Guo et al., 2015			X	X			
GSA	Ibrahim et al., 2015		X	X				
ICA	Zhou et al., 2013	X			X			
HA	Chen et al., 2002		X	X				
	Shan et al., 2006		X	X				
	Ning and Gu, 2007		X	X				
	Shuang et al., 2008	X	X					
	Tseng et al., 2008		X	X				
	Shan et al., 2009			X	X			
	Zhou et al., 2011		X	X				
	Li et al., 2013		X		X			
	Zhang et at., 2014			X	X			X
KBS	Huang and Lee, 1991		X	X				
	Zha et al., 1998	X	X	X				X
	Dong et al., 2007		X	X				X
	Hsu et al., 2011		X	X				
	Kashkoush and ElMaraghy, 2015			X	X			X

Note: In the above table, A-Geometric and topology details, B- Contact details, C- Precedence details, D- Assembly interference matrix, E- connection details, F-Mating conditions and G-Stability matrix.

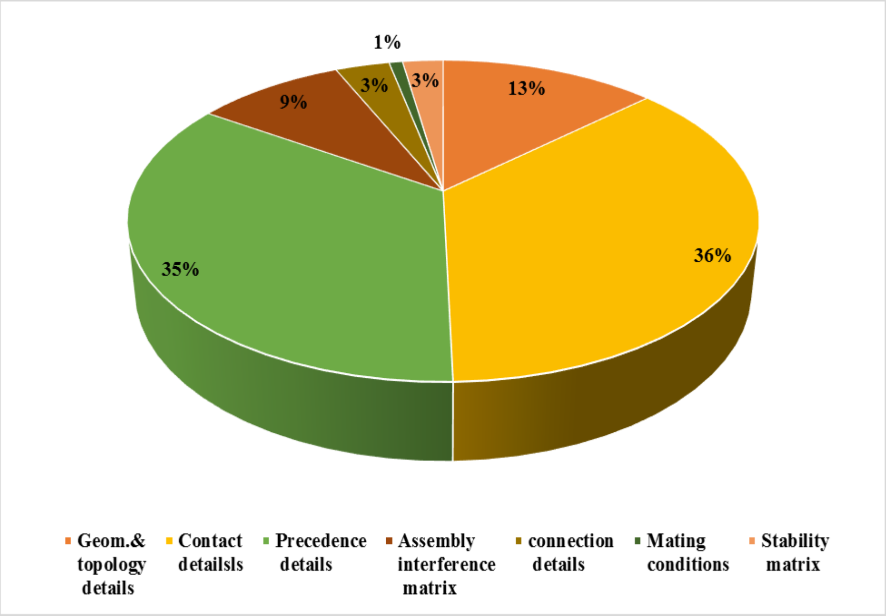


Figure 2.9: Input considerations in optimal ASG methods

It is the observed that most of the researchers' considered combination of assembly precedence data and assembly contact data for assembly sequence generation and to perform geometric feasibility testing. The recent literature indicates interference matrix and assembly stability matrices were helpful to assure assembly stability besides geometric feasibility. The contribution of Input in ASG methods is represented in Figure 2.9. Minimization of the assembly time/cost is considered a primary goal of optimal ASG, for which combination of assembly directional changes, assembly tool/gripper changes, assembly instability, assembly operational time, assembly energy were used. Consideration of these elements in the cited research literature is listed in Table 2.3.

Table 2.3: Objective function formulation in the cited literature.

Algorithm	**Reference**	**AA**	**BB**	**CC**	**DD**	**EE**	**FF**	**GG**
GA	Wong and Leu, 1993	X						
	Bonneville et al., 1995	X						
	Hong and Cho, 1999	X			X			
	Dini et al., 1999	X	X					
	Lazzerini and Marcelloni, 2000	X	X	X				X
	Smith et al., 2001	X	X					
	Chen and Liu, 2001	X	X					
	De Lit et.al, 2001	X						
	Smith and Smith, 2002	X	X		X			
	Smith, 2004	X	X		X			
	Guan et al., 2002	X	X		X			
	Marian et al., 2003& 2006	X	X					
	Del Valle et at., 2003	X						
	Tseng et al., 2004	X						
	Bai et al., 2005			X				
	Huang et al., 2007					X		
	Hui et al., 2008		X	X				
	Choi et al., 2009	X		X				
	Tseng et al., 2010	X	X		X			
ACO	Falli and Dini, 2000	X						
	Wang et al, 2003& 2005	X						
	McGovern and Gupta, 2006	X		X				
	Sharma et al., 2008	X						
	Shi et al., 2010	X						
	Yu and Wang, 2013	X	X					
	Wang et al., 2014	X	X					
	Lu, C. and Yang, Z., 2015	X	X	X				

Table 2.3 Objective function formulation in the cited literature. (Contd..)

NN	Chen and Pao, 1993		X					
	Hong and Cho, 1993 & 1995	X			X			
	Sinanoglu and Riza Börklü, 2005	X						
	Chen et al, 2008				X			
SA	Milner et al., 1994	X						
	Motavalli and Islam, 1997	X	X					
	Hong and Cho, 1997 &1999	X	X					
	Lee and Gemmill, 2001	X						
PSO	YU et. at, 2010	X	X					
	Lv and Lu, 2010	X						
	Wang and Liu, 2010	X	X					
	Tseng et. al, 2011	X					X	
AIS	Cao and Xiao, 2006	X	X					
	Chang et al., 2009	X						
	Biswal et al., 2013	X						
MA	Tseng et al., 2007	X	X					
	Gao et al., 2010	X						
	Zeng et al., 2011	X	X					
BLS	Ghandi and Masehian, 2015	X						
FLA	Guo et al., 2015	X	X					
GSA	Ibrahim et al., 2015			X				
ICA	Zhou et al., 2013	X	X					
HA	Chen et al., 2002		X					
	Shan et al., 2006	X						
	Ning and Gu, 2007	X	X					
	Shuang et al., 2008	X						
	Tseng et al., 2008	X		X				
	Shan et al., 2009	X	X					
	Zhou et al., 2011	X	X					
	Li et al., 2013	X	X					
	Zhang et at., 2014	X	X					
KBS	Huang and Lee, 1991	X						
	Zha et al., 1998			X	X		X	
	Dong et al., 2007	X						
	Hsu et al., 2011	X						
	Kashkoush and ElMaraghy, 2015	X			X			

Note: AA-Directional Changes, BB-Tool/Gripper changes, CC-Assembly Time, DD-Assembly Instability, EE-Energy for part handling, FF-Assembly Cost, GG-Reliability.

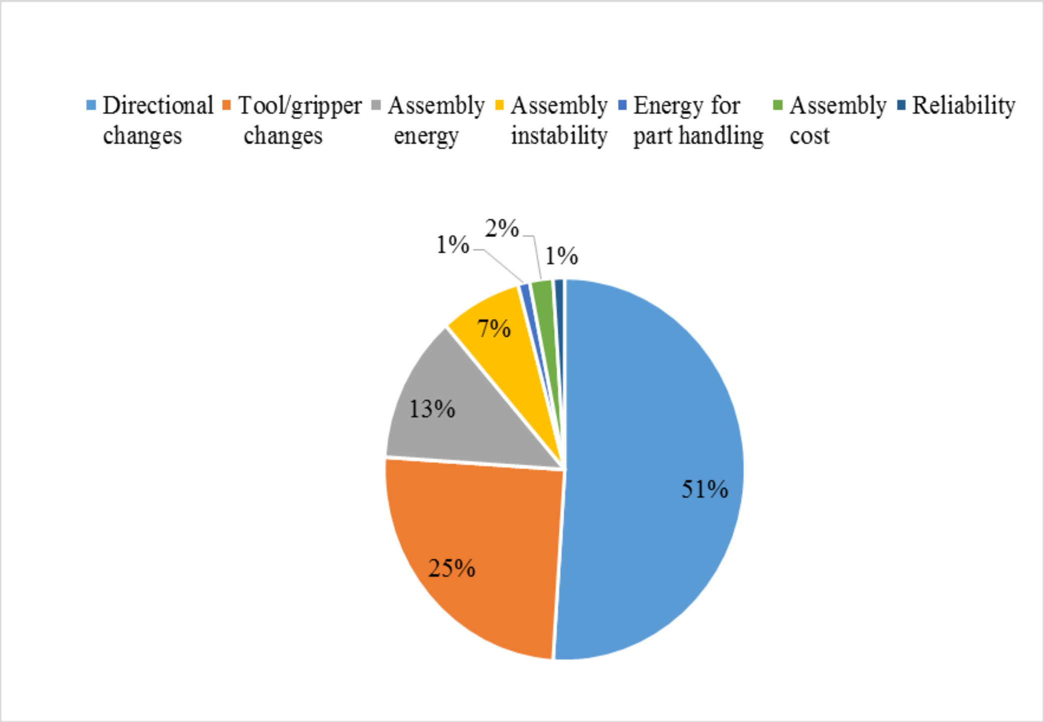


Figure 2.10: Objective function consideration from cited research literature

It is observed from Figure 2.10, most of the cited research literature given highest priority to assembly directional changes and tool changes, due to their major influence in assembly time. Assembly energy and assembly stability were given next priority for objective function consideration. The assembly predicates considered to perform the assembly sequence planning assure the quality of assembly sequence and computational performance of the method. The below table 2.4 lists the assembly predicate consideration in the cited research literature.

Table 2.4: Assembly predicate consideration in the cited literature

Algorithm	**Reference**	**Liaison**	**Geometric Feasibility**	**Assembly Stability**	**Mechanical feasibility**
GA	Wong and Leu, 1993	C	C	C	NA
	Bonneville et al., 1995	C	C	NC	NA
	Hong and Cho, 1999	C	C	C	NA
	Dini et al., 1999	C	C	C	NA
	Lazzerini and Marcelloni, 2000	C	C	C	NA
	Smith et al., 2001	C	C	NC	NA
	Chen and Liu, 2001	C	C	C	NA
	De Lit et.al, 2001	C	C	NC	NA
	Smith and Smith, 2002	C	C	C	NC

Table 2.4 Assembly predicate consideration in the cited literature.(Contd..)

GA	Smith, 2004	C	C	C	NC
	Guan et al., 2002	C	C	C	NA
	Marian et al., 2003& 2006	C	C	C	NA
	Del Valle et at., 2003	C	C	NC	NA
	Tseng et al., 2004	C	C	NC	NA
	Bai et al., 2005	C	C	C	NA
	Huang et al., 2007	C	C	NC	NA
	Hui et al., 2008	C	C	NC	NA
	Choi et al., 2009	C	C	NC	NA
	Tseng et al., 2010	C	C	C	NC
ACO	Falli and Dini, 2000	C	C	NC	NA
	Wang et al, 2003& 2005	C	C	NC	NC
	McGovern and Gupta, 2006	C	C	NC	NA
	Sharma et al., 2008	C	C	NC	NC
	Shi et al., 2010	C	C	NC	NA
	Yu and Wang, 2013	C	C	NC	NA
	Wang et al., 2014	C	C	NC	NA
	Lu, C. and Yang, Z., 2015	C	C	C	NA
NN	Chen and Pao, 1993	C	C	NC	NA
	Hong and Cho, 1993 & 1995	C	C	NC	NA
	Sinanoglu and Riza Börklü, 2005	C	C	C	NA
	Chen et al, 2008	C	C	C	NA
SA	Milner et al., 1994	C	C	NC	NA
	Motavalli and Islam, 1997	C	C	NC	NA
	Hong and Cho, 1997 &1999	C	C	NC	NA
	Lee and Gemmill, 2001	C	C	NC	NA
PSO	YU et. at, 2010	C	C	NC	NA
	Lv and Lu, 2010	C	C	NC	NA
	Wang and Liu, 2010	C	C	C	NC
	Tseng et. al, 2011	C	C	NC	NA
AIS	Cao and Xiao, 2006	C	C	C	NA
	Chang et al., 2009	C	C	C	NA
	Biswal et al., 2013	C	C	NC	NA
MA	Tseng et al., 2007	C	C	NC	NA
	Gao et al., 2010	C	C	NC	NA
	Zeng et al., 2011	C	C	NC	NA
BLS	Ghandi and Masehian, 2015	C	C	C	NA
FLA	Guo et al., 2015	C	C	C	NA
GSA	Ibrahim et al., 2015	C	C	C	NA

Table 2.4 Assembly predicate consideration in the cited literature.(Contd..)

ICA	Zhou et al., 2013	C	C	NC	NA
HA	Chen et al., 2002	C	C	NC	NA
	Shan et al., 2006	C	C	NC	NA
	Ning and Gu, 2007	C	C	NC	NA
	Shuang et al., 2008	C	C	NC	NA
	Tseng et al., 2008	C	C	C	NA
	Shan et al., 2009	C	C	C	NA
	Zhou et al., 2011	C	C	C	NA
	Li et al., 2013	C	C	NC	NA
	Zhang et at., 2014	C	C	C	NC
KBS	Huang and Lee, 1991	C	C	NC	NA
	Zha et al., 1998	C	C	C	NA
	Dong et al., 2007	C	C	C	NA
	Hsu et al., 2011	C	C	C	NA
	Kashkoush and ElMaraghy, 2015	C	C	C	NA

A detailed analysis on the effect of predicate consideration on performance of assembly sequence planning for different assembly configurations are discussed in Chapter 4.

2.6 Problem Statement

The problem of assembly sequence generation is envisaged in a manner that takes care of all necessary assembly predicates and the process could be carried out either by manual computation or could be automated to take care of large products with more number of parts by directly considering CAD data.

Study the significance of assembly sequencing in product manufacturing, various methods developed and adopted for the purpose.

Develop an integrated efficient and easy to understand process of generating assembly sequence of large assembled products with multiple parts.

2.7 Summary

Research literature from past five decades in the domain of assembly sequence generation through accessible sources is studied. The paradigm of research is well captured for different objectives of ASG. The review analysis is prepared and presented by covering each aspect of assembly sequence generation methods for the assistance of the readers.

From the stated literature reviews it is observed that

- Assembly sequence planning is involved with multiple assembly predicates to ensure practical possibility, there exist several methods to retrieve the necessary assembly information manually and through CAD interface.
- The manual methods of assembly attribute extraction lot of skill and extremely time consuming, and the computer aided methods save computational time and assures accuracy.
- Due to huge number of possible assembly sequences, generation of set of all feasible assembly sequences is complex and time consuming, which further motivated researcher towards soft computing techniques.
- Numerous researcher used Artificial intelligence (AI) based methods to find out optimal assembly sequence, It is also observed that (AI) based techniques did not consider some assembly predicates for computation performance.
- AI techniques do not assure optimal solution always for all assembly configurations.
- Although KBS techniques and CAD based techniques require less human intervention, but demands skilled user intervention.

By considering these limitations the necessity of an effective ASG method is defined as problem statement.

Chapter 3

METHODOLOGY

3.1 Overview

In order to perform assembly sequence generation and assembly sequence validity testing, product information is required. The necessary product information is described in the first section and extraction of the information from the 3D CAD models through computer aided techniques is described in Methods section. Assembly predicate testing is to validate assembly sequences also described in this section.

3.2 Product Information

The research aims at developing the process of assembly sequence generation for mechanical products. It involves lot of numerical data and information related to parts and part relations. Essentially the problem is informative/ data intensive. Information about the product and relationship of parts with each other in the product assembly and mutual mating behaviour of the parts during assembly is described. The following are the important in the view of the proposed solution methodology.

Liaison matrix
Bounding box coordinates
Assembly interference matrix
Stability matrix
Mechanical feasibility matrix

3.2.1 Liaison Matrix

Liaison is a significant connection between two parts, when the parts are in assembled position. For an assembled product, liaisons between the parts can be shown through a graphical representation, which is called as liaison graph (Bourjault, 1984). The concept of liaison graph is initially proposed by Bourjault in 1984 for the purpose of assembly sequence generation. Liaison graph is defined as LG= G(V,E) graph of vertices and edges(connecting hyper arcs). Each vertex indicates an assembled part and each edge signifies the liaison between two components in state of contact.

Liaison graph is created and presented in Figure 3.1 for the transmission assembly by considering the contacts between all 11-primary parts. However, physical connectors are not considered for this purpose. Total 18 liaisons are found, which are represented by symbol -C_i .

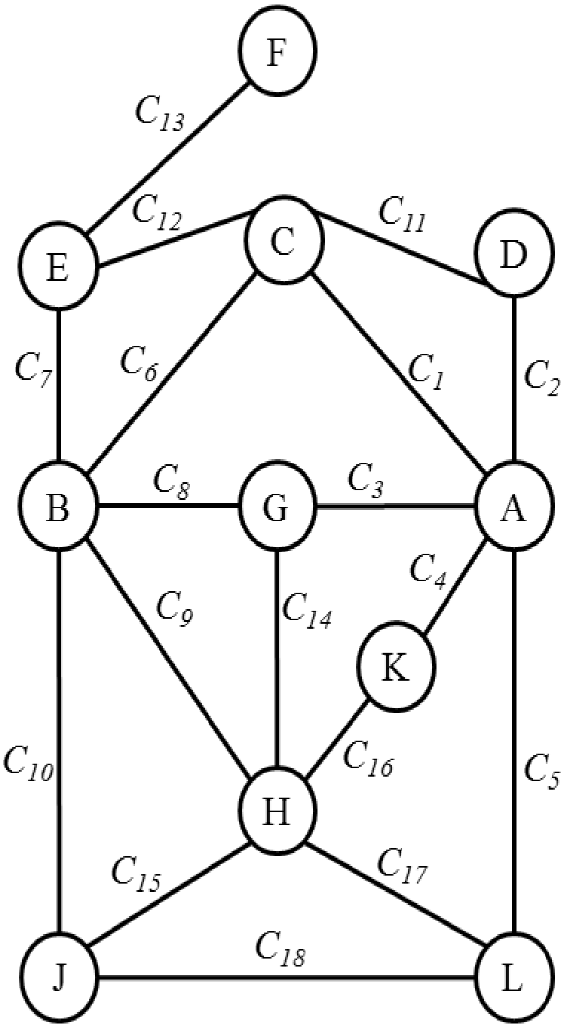


Figure 3.1: Liaison diagram for transmission assembly

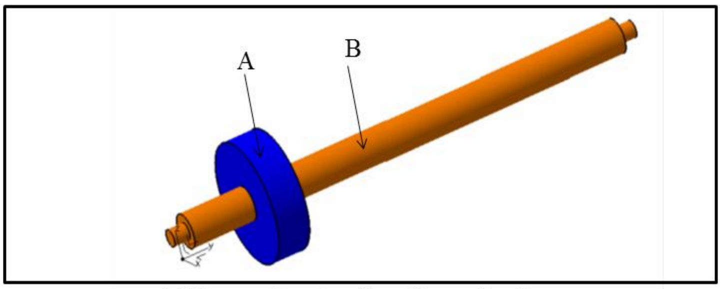
Matrix representation of liaisons is proposed by Dini using binary codes 1, 0 (Dini et al., 1999). An *"n-by-n"* matrix is required to represent all the liaisons of a product assembled by "n" components. The diagonal elements of this matrix always consist null values, and a row of matrix represents the liaisons for one component with the other components in the assembly. The column of matrix represents the components connected by liaison relationships. The sub-matrices of *"nxn"* matrix represent the local liaison relationships in subassemblies. The liaison matrix for the transmission assembly is as follows.

$$\begin{array}{c|ccccccccccc|}
 & A & B & C & D & E & F & G & H & J & K & L \\
\hline
A & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
B & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
C & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
D & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
F & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
H & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
J & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
K & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
L & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}$$

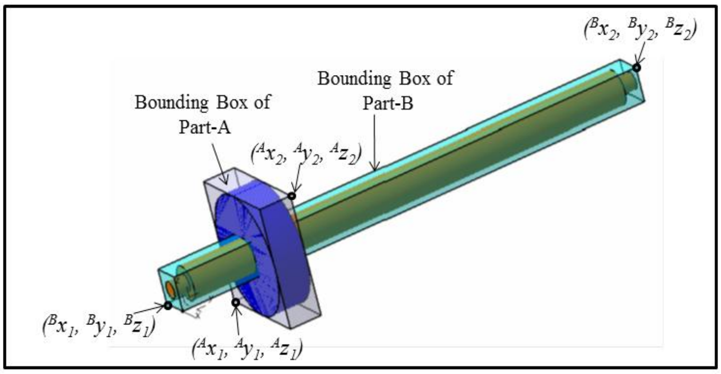
Liaison matrix for transmission assembly

3.2.2 Bounding Box Coordinates

Bounding box (Axis Aligned Bounding box) is a minimum cuboid envelope for a part or assembly generated with edges parallel to the principal axes. Bounding box can be represented with two extreme diagonal points of the cuboid. The part bounding box coordinates are further used to determine the distance to be travelled by a part along a specified orientation to perform assembly operations. While generating the interference matrices along all principal axes directions part bounding box coordinates are used for defining part trajectory intervals. Figure 3.2 shows part bounding box coordinates.



(a) Two parts set without bounding box



(b) Two parts set with bounding box coordinates

Figure 3.2: Representation of Bounding box and their coordinates

The bounding box coordinates (x_1, y_1, z_1) represents lower limit of the part geometry and (x_2, y_2, z_2) represents upper limit of the part geometry. The difference between upper and lower limit elements along a specified direction is considered to test the geometric feasibility of along it. For example elements $({}^{B}y_2\ ,{}^{A}y_1)$ are required to test the geometrical feasibility of assembling part-A in the existence of part-B along “y-”direction and assembling part-B in the presence of part-A along “y+” direction.

3.2.3 Assembly Interference Matrix

An assembly sequence is said to geometrically feasible, when all parts can bring into contact in a defined sequence without any collision. The geometrical feasibility is a function of part geometries. Precedence relations are used to test for feasibility of a part to assemble. Precedence relations give information about preceding and succeeding assembly operations to achieve feasible assembly sequence. Bourjault proposed a method to generate list of questions based on the liaison’s graph, the answers to these questions create precedence relations between assembly connections. The Bourjault method of questioning based on liaison is as followed.

Question1: Is that true liaison C_i can be established if liaisons $(C_j\ldots.. C_k)$ have been established.
Question1: Is that true liaison C_i can be established if liaisons $(C_j\ldots.. C_k)$ have not been established.

The group of liaisons $(C_j\ldots.. C_k)$ are called body of liaisons. These questions must be answered with “YES” or “NO” to determine the possibility or non-possibility to assemble a component to product. The user must have the knowledge on the feasibility of assembly operation to answer “Yes/No”.

De Fazio and Whitney modified the format of questions to minimize the efforts by reducing the number of questions. Lui created a computer aided program to generate the set of questions by considering liaison graph as input. Baldwin developed an algorithm to generate precedence relations by using assembly cut-sets (Baldwin et al., 1991). The program receives user supplied answers to generate the precedence relations. Connectivity graph often called as supported graph initiated by Shpitalni to represent the stable assembly and disassembly connections of an assembled product (Shpitalni et al., 1989). Components of the assembly can be moved in any single or combination of the six directions along the positive and negative principal axes (+x, -

x, +y, -y, +z, -z). Graphical representation of assembly constraints is proposed by Wolter (Wolter, 1989). This method typically uses six variety of part trajectories and the constraints which control the trajectories. This method of representation is more useful in assembly sequence feasibility test. The graph provides the information about each part trajectories in the work space. Figure 3.3 represents the exploded view of a scissor assembly to demonstrate the assembly constrain graph.

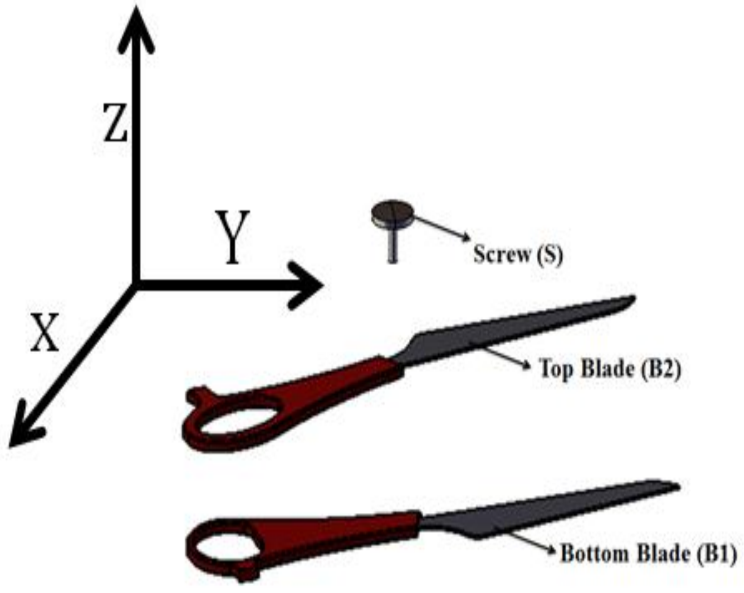


Figure 3.3: Exploded View of a Scissors assembly

Each part is represented in nodes and the part-trajectories are mentioned as sub nodes inside the part node. The arcs are drawn from part trajectory node to a part, which constraints the motion. Assembly constraint graph for scissors assembly is shown in Figure.3.4.

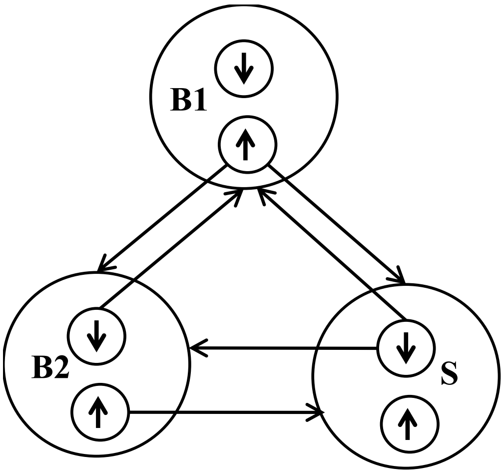


Figure 3.4: Assembly constraint graph for scissors assembly

The above graphical representation methods routed towards generating assembly mating conditions and assembly interference matrix representation. Assembly interference matrix is a *'nxn'* matrix along each principal direction to indicate the interference or collision between two components during assembly process using binary codes.

When a part (Part-i) is moved in a specified assembly direction to join with the other parts existed already in assembled position, If any part (part-m) exist already in the assembled position causing interference in the specified direction then the collision indicated by "0" in the matrix and the collision free assembly operation indicated by "1" in the matrix for ith row and mth column. The interference matrix always consist null diagonal elements.

The interference matrix for scissors assembly along "+z" and "-z" directions are as follows:

$$\begin{array}{c|ccc|} & b1 & b2 & s \\ b1 & 0 & 1 & 1 \\ b2 & 0 & 0 & 1 \\ s & 0 & 0 & 0 \end{array} \quad \text{and} \quad \begin{array}{c|ccc|} & b1 & b2 & s \\ b1 & 0 & 0 & 0 \\ b2 & 1 & 0 & 0 \\ s & 1 & 1 & 0 \end{array}$$

3.2.4 Stability Matrix

Stability predicate plays significant role in assembly sequence generation. An Assembly sequence is said to stable, when each of the part in the assembly maintain its position with respect to the other parts at all stages of assembly operations. Representation of stability using stable relations between each pair of parts is proposed by Smith (Smith and Liu, 2001).

Smith proposed representation of assembly connections in matrix format; the connections are categorised into two types; hard and soft. When two parts are connected by physical connectors, the connection is considered as hard and if two components are just maintain their position by surface contact without any physical connection is referred to be soft connection. In the current research, stability is broadly classified as partial and permanent stability.

A component is treated as partially stable when it does not lose its contact with all mating parts due to application of gravitational force. However, if the assembly is oriented, the parts may lose its contacts. Partial stability of part is more considered in linear assembly planning process. Further classification of permanent stability is made by the usage of external attachments, and surface features. A component is treated as permanently stable, when it is connected through its surface features or by external connectors in order to maintain all its contacts with mating parts irrespective of the orientation. Permanent stability is an essential criterion for sub-assembly detection.

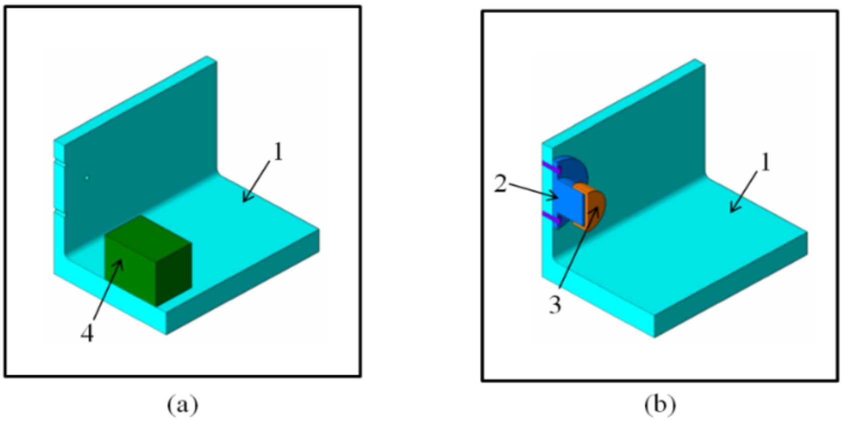


Figure 3.5: Assembly sub-sets for stability demonstration

Figure 3.5 (a) indicates a partially stable assembly subset (1-4), in which parts do not maintain its contact with respect to all mating parts when the assembly sub-set is rotated. Figure 3.5 (b) represents a permanently stable assembly sub-set (1-2-3) which can be treated as a stable sub-assembly for further level of assembly possess.

The connection data for an assembled product can be represented by a "*nxn*" matrix for a product with "n" number of primary parts.. Element [i][j] of the connection matrix represents how part-i is connected with part-j. Element value 1 represents soft connection, i.e., part-i is stable at its position with respect to part-j against gravity without any physical connection. Element values 0, 1, 2 and 3 successively represent no stability, partial stability, permanent stability due to part features and permanent stability by external physical connectors. An assembly subset with 4 parts and 4 attachments cutaway is shown in Figure 3.6.

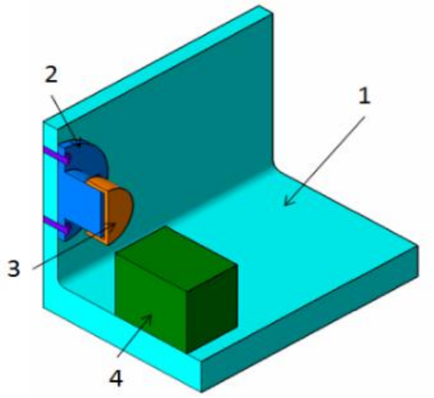


Figure 3.6: Cutaway of 4 part assembly subset with 4 attachments

For the assembly shown in the Figure 3.6, Part-4 is supported by part-1 and exhibits partial stability, part-2 and part-3 are connected by means of surface threading exhibits permanent stability. Physical connectors were used to join part-2 with part-1. Assembly stability matrix based on the type of stability between the parts are indicated in the below.

$$\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 0 & 0 \\ 2 & 3 & 0 & 2 & 0 \\ 3 & 0 & 2 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 \end{array}$$

Stability matrix for 4parts assembly

3.2.5 Mechanical Feasibility Matrix

Mechanical feasibility is true for an assembly sequence when the assembly tools can perform the specified assembly operation without any collision; hence it is dependent on tools and methods used to perform the assembly operations. The hard connectors trajectory constraints can be represented through a three dimensional matrix of n-by-n-by-n. The third dimension represents, whether the part represented in it offers any interference to place hard connections between parts represented in first two dimensions. For example, joining part “A” to a subassembly “CD” is mechanically infeasible due to the existent part “D”. Part-D does not allow to join the hard connectors between part A and C hence the mechanical infeasibility representation is as follows mfm(A,C,D)=1. Mechanical feasibility matrix for the 4-part assembly shown in Figure 3.6 is indicated in Figure 3.7.

$$4\quad\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$

$$3\quad\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$

$$2\quad\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$

$$1\quad\begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{array}$$

Figure 3.7: Mechanical feasibility matrix for 4 part assembly

3.3 Methods

A product with “n” number of parts can have factorial-n number of set of all assembly sequences, very few of these sequences are only practically possible due to several geometrical and assembly operational constraints. In order to test the validity of an assembly sequence, it is necessary to extract the required information from the CAD based models through manual mode or automatically.

The methods in the present research problem relate to the following

Extraction of liaison matrix, interference matrices, stability matrix and

Mechanical feasibility matrix.

Methods for assembly predicate testing.

3.3.1 Liaison Matrix Extraction

Liaison matrix has to be prepared by considering only primary parts of the assembly and thus all the mechanical connectors must be hidden before generating the liaison matrix from assembled product. In an assembled product, between any pair of two parts generally three types of mating possibilities exit due to their geometric boundaries. The first possibility is interference where two part boundaries are overlapped with one another. The second state refers to clearance due to the gap between the boundaries whereas third state of possibility is perfect contact between boundaries at single or multiple faces of the parts. The interference between part boundaries indicates improper alignment of parts in the CAD model, which is not desirable.

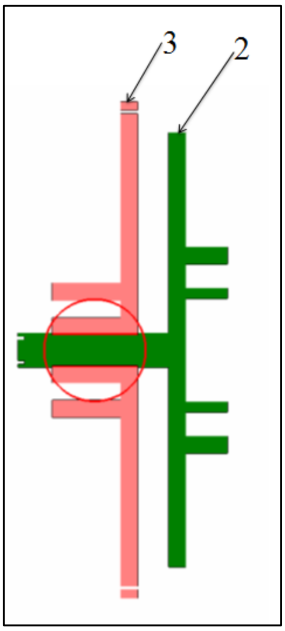
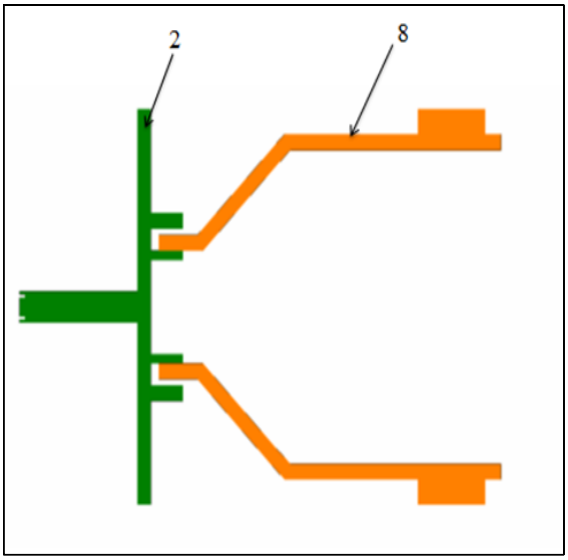
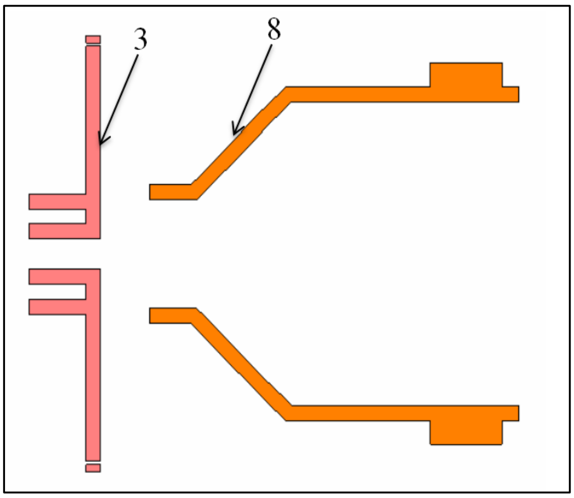


Figure 3.8: Clash enabling mode cut-section of (2-3) assembly subset

Each product must be tested for interference mating condition before extracting liaison matrix. Figure 3.8 depicts interference between two pair of parts. Contact and clearance possibilities are shown in figure 3.9(a) and 3.9(b) successively by considering parts from transmission assembly.



(a) Instance for Contact



(b) Instance for Clearance

Figure 3.9: Assembly subsets to demonstrate contact and clearance possibilities

Assembly contact analysis is more useful to determine the above stated conditions between each pair of parts in the assembled product. An n-by-n null matrix is generated too store the liaison information and then assembly contact analysis is performed on the product.

Once all the assembly mating conditions are obtained, filter the instances for each state of contact condition between two parts. Both the first part represented row and second part represented column element and its symmetric element must be replaced with “1”. The process will be iterated for each resulted instance of assembly contact analysis results the liaison matrix. Flow chart is shown in figure 3.10 to extract liaison matrix from CAD environment.

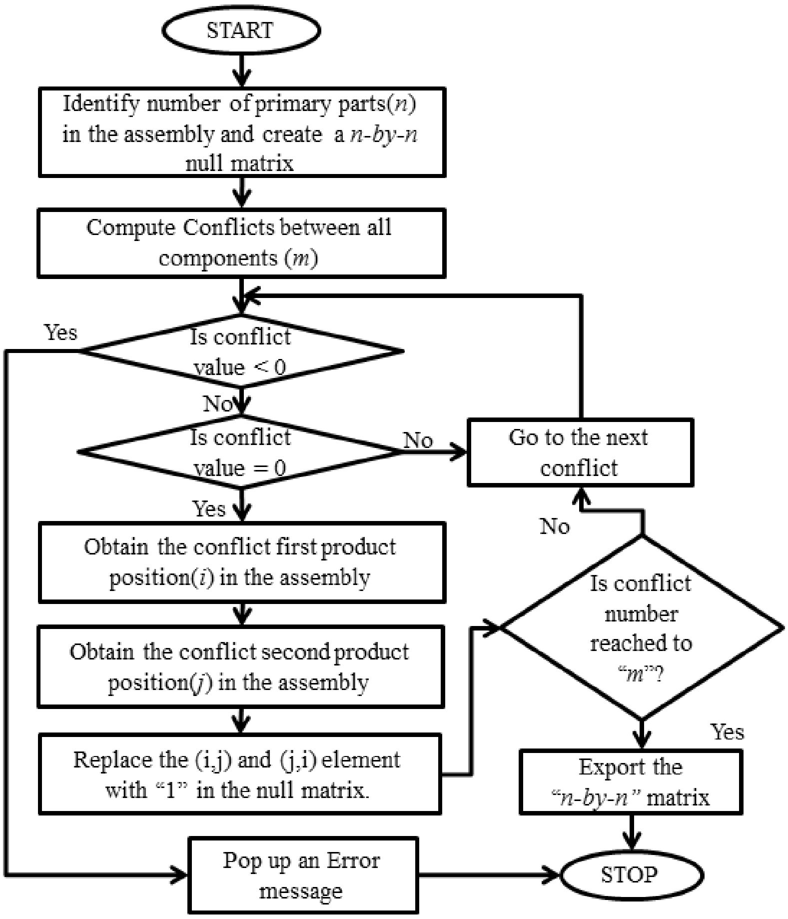


Figure 3.10: Mechanism to extract liaison matrix from 3D CAD environment

```
Pseudo code for Liaison matrix extraction based on the flow chart is mentioned below.
Step 1: Open an assembled product
Step 2: Hide all the physical connectors
Step 2: n ← Obtain the number of primary parts in the Product
Step 3: Create a null matrix of n-by-n
Step 4: Obtain all the possible mating instances
Step 5:  m ← Total number of instances
Step 6:  For each instance 1 to m
    Define the type by its value
    If Value < 0
        Popup an error and exit from the loop
    If Value = 0
        Identify the conflict part.1 name in the parts list (say i^th part)
        Identify the conflict part.2 name in the parts list (say j^h part)
        Replace the null value with “1” for the (i,j) and (j,i) elements of null matrix
Step 7: Export the matrix data
```

Macro is developed using VB Script (Visual Basic) to interface with CATIA environment to generate the liaison matrix for different assemblies. Gear assembly shown in Figure 3.11 and transmission assembly shown in Figure 3.12 were considered for this purpose. These liaison matrices were validated with respect to the liaison diagrams stated in the References.

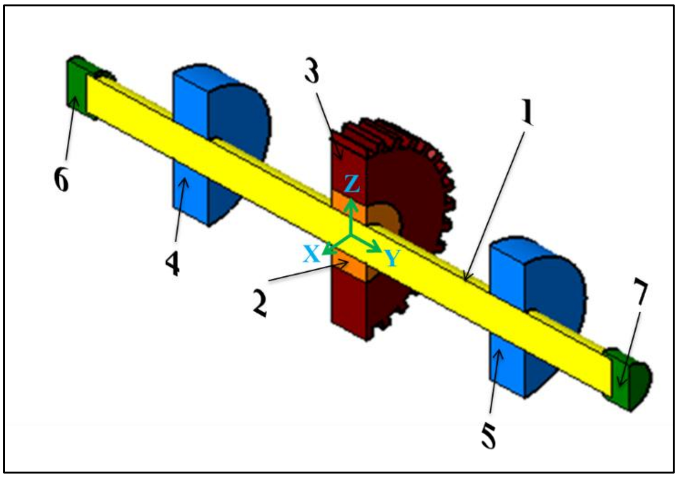


Figure 3.11: Gear assembly cutaway for liaison matrix extraction

To avoid the confusion, Gear assembly parts are named with numeric(s) and transmission assembly parts are named using alphabets; 3D-Cutaway of transmission assembly with axis system is represented in Figure 3.12 with nomenclature.

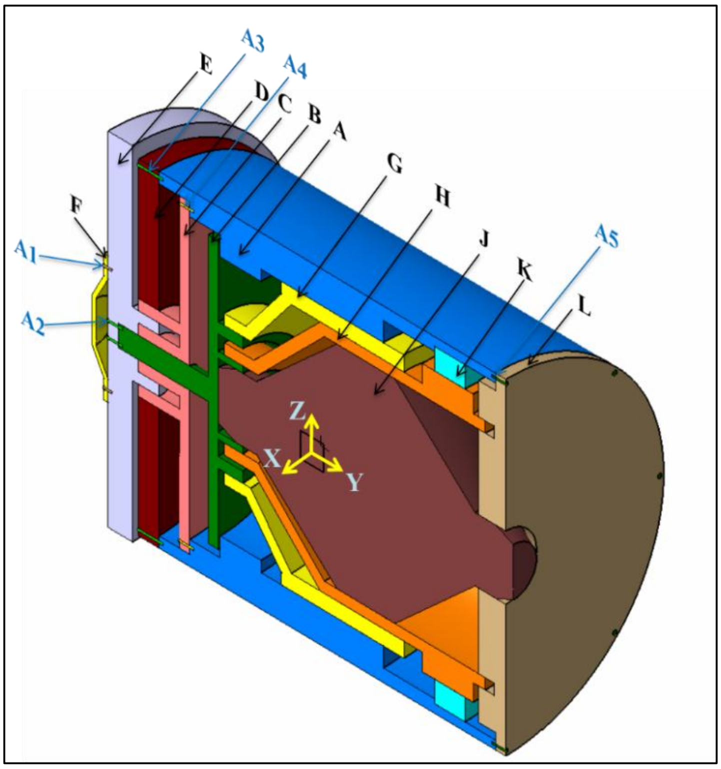


Figure 3.12: Transmission assembly cutaway for liaison matrix extraction

Resulted liaison matrices for Gear assembly and Transmission assembly are as follows:

$$\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Liaison matrix for Gear assembly

$$\begin{array}{c|ccccccccccc|}
 & A & B & C & D & E & F & G & H & J & K & L \\
A & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
B & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
C & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
D & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
F & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
H & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
J & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
K & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
L & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}$$

Liaison matrix for Transmission assembly

3.3.2 Interference Matrices Extraction

In order to test the interference between a pair of parts along a specified assembly orientation, there exist three instances based on their geometrical boundary. Bounding box coordinates are used to describe the geometry of parts.

Instance-1: If the lower limit of part-i is more than upper limit of part-j, and hence part-j does not interfere while part-i is disassembled along the specified positive direction. These instances must be ignored from the feasibility testing by reading bounding box coordinates of the parts and unit value has to be assigned to the directed interference matrix element.

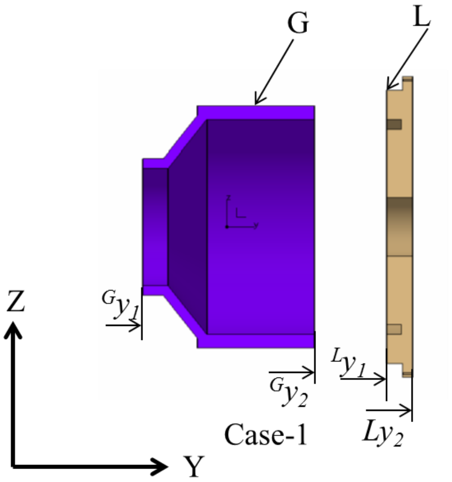


Figure 3.13: First instance of feasibility testing conditions

For example, consider pair of parts part-G and part-L from transmission assembly, the lower limit of part-L (${}^{L}y_1$) is higher than upper limit of part-G (${}^{G}y_2$). Thus in the presence of part-G, part-L can be either disassembled along "y+" direction or assembled along "y-" direction without any collision.

Instance-2: Lower limit of part-i is less than upper limit of part-j and hence part-j may interfere while part-i is disassembling along the specified positive direction. Difference between upper limit of part-j to lower limit of part-i will be considered as distance to be travelled by part-i without any collision. These instances must be considered for the feasibility testing.

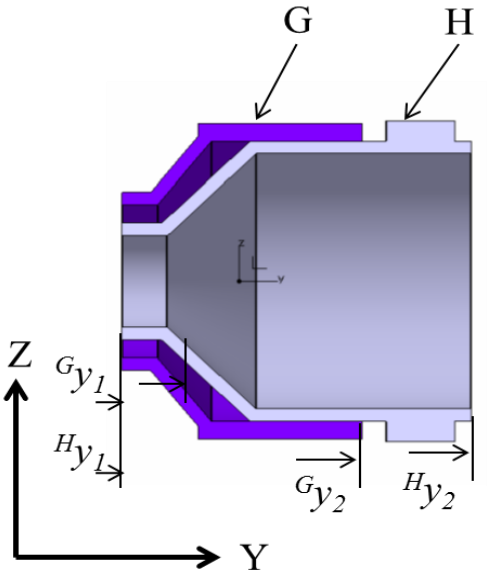


Figure 3.14: Second instance of feasibility testing conditions

For example, considering pair of parts part-G and part-H from transmission assembly, the lower limit of part-H ($^{H}y_1$) is smaller than upper limit of part-G ($^{G}y_2$). Thus in the presence of part-G, feasibility of disassembling part-H has to be tested along "y+" direction.

Instance-3: If upper limit of part-i is less than lower limit of part-j, then part-i will be moved towards part-j till both the values are matched and Instance-3 will be turned out to instance-2 category. An example for Instance-3 is indicated in Figure 3.15.

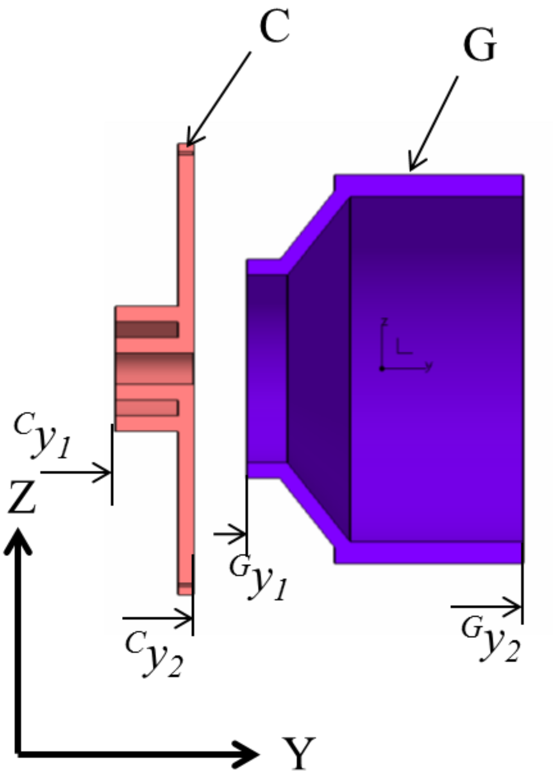
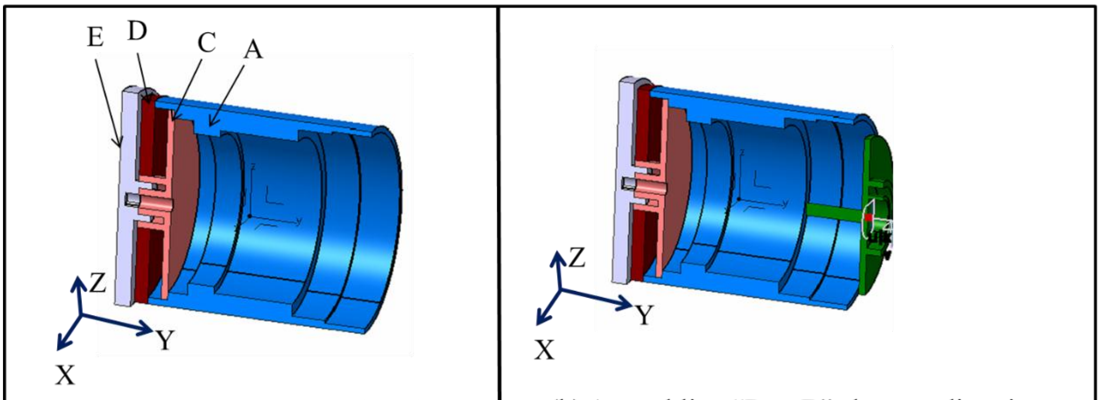


Figure 3.15: Third instance of feasibility testing conditions

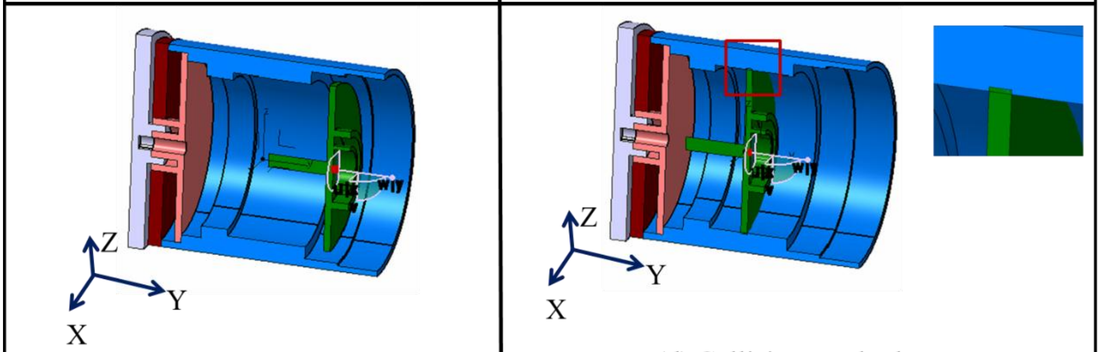
Considering pair of parts part-G and part-C from transmission assembly shown in Figure 3.15 , the upper limit of part-C ($^{C}y_2$) is lower than lower limit of part-G ($^{G}y_1$). Thus Part-C must be moved to a distance ($^{G}y_1$-$^{C}y_2$) and then feasibility of disassembling part-3 has to be tested along "y+" direction.

In order to detect the feasibility of dis assembling a component, the component must be checked for interference with respect to the existent part throughout its trajectory. Collision free trajectory indicates feasibility of disassembly operation indicated by "1" for the interference matrix element. Any interference during the part trajectory indicates infeasibility and the interference matrix element will be represented with null value. Occurrence of interference while assembling part-B to an assembly subset (A-C-D-E) is shown in Figure 3.16.



(a) (A-C-D-E) Assembly subset

(b) Assembling “Part-B” along y- direction



(c) Part trajectory in No-collision zone

(d) Collision resulted zone

Figure 3.16: Collision detection during part trajectory demonstration

An algorithm to extract the interference matrix along a specified positive direction(X+, Y+ ,Z+) is presented here below. The code is to test whether part-i, can be disassembled without any collision during the existence of part-j in a specified positive orientation.

```
Pseudo code to extract interference matrix along "y+" direction

Step 1: Consider an assembled product
Step 2: Hide all the physical connectors
Step 3: n ← Obtain the number of primary parts in the Product
Step 4: Create a null matrix of n-by-n
Step 5: For i=1 to n
            For j=1 to n & j≠i
                Hide all primary parts other than Part-i & part-j
                Use bounding box coordinates for Part-i & part-j
                Compute the distance   $d = ({}^{j}y_2 - {}^{i}y_1)$
                (Instance 1)
```

```
        If d <=0
            Feasibility matrix element (i,,j) =1
        End If
       (Instance 3)
If d >0 and ($^jy_1$-$^iy_2$)>0
            Move Part-i along "y+ direction" to a distance of ($^jy_1$-$^iy_2$)
            (This will convert Instance 3 to Instance 2)
       End If
      (Instance 2)
       If d >0
            For k=1 to d
                  Move the part-i along y+ direction distance of "k"
                  Test for the interference between part-i & part-j
            If k=d
                        Feasibility matrix element (i,,j) =1
                        Bring the part-i to its original position
                  End If
            If there exist interference
                        Feasibility matrix element (i,,j) =0 and k=d
                        Bring the part-i to its original position
                  End If
            End For
      End If
      Unhide all primary parts
     End For
     End For
Step 7: Export the matrix data
```

The algorithm presented above is presented indicated in Figure 3.17. The above code can be used to extract interference matrices along "x+" and "z+" directions by considering their respective bounding box coordinates.

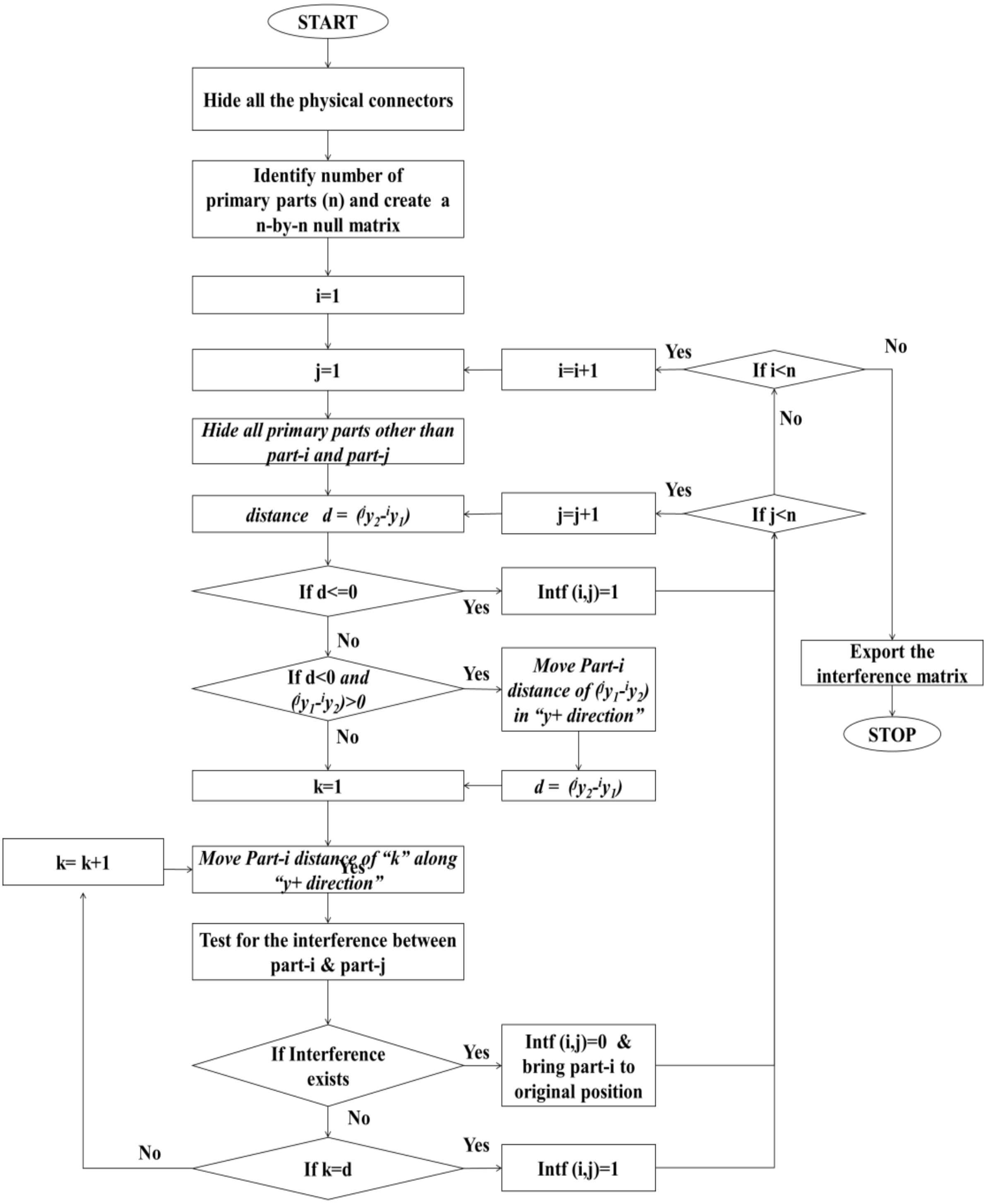


Figure 3.17: Mechanism to extract interference matrix along "y+" direction

The algorithm is programmed using VB script to interface with CATIA v5 in order to extract the interference matrices. The interference matrices obtained for the Gear assembly shown along "X+, Y+ and Z+" directions are as follows.

$$\begin{array}{c|ccccccc} X+ & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 4 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 5 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 7 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{array} \quad \begin{array}{c|ccccccc} Y+ & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 4 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 5 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 6 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 7 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array} \quad \begin{array}{c|ccccccc} Z+ & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 4 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 5 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 7 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{array}$$

Interference matrices along (X+, Y+ & Z+) directions for Gear assembly

The interference matrices obtained for the transmission assembly along “X+, Y+ and Z+” directions are as follows.

$$\begin{array}{c|ccccccccccc} (X+) & A & B & C & D & E & F & G & H & J & K & L \\ \hline A & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ D & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ E & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ F & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ G & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ H & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ J & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ K & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ L & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array} \quad \begin{array}{c|ccccccccccc} (Y+) & A & B & C & D & E & F & G & H & J & K & L \\ \hline A & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ F & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ G & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ H & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ J & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ K & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ L & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array}$$

$$\begin{array}{c|ccccccccccc} (Z+) & A & B & C & D & E & F & G & H & J & K & L \\ \hline A & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ D & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ E & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ F & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ G & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ H & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ J & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ K & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ L & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{array}$$

Interference matrices along (X+, Y+ & Z+) directions for transmission assembly

It is observed from the above, interference matrices about “X” axis and “Z” axis are same due to the axisymmetric nature of assembly about “Y” axis. To test the geometric feasibility about positive direction, distance is measured using upper limit of part-j and lower limit of part-i *(*${}^{j}y_{2}$*-*${}^{i}y_{1}$*, along “y+” direction)* and part-i is moved to this distance for collision detection. Similarly to test the geometric feasibility along negative directions, lower limit of part-j and upper limit of part-i *(*${}^{i}y_{2}$*-*${}^{j}y_{1}$*, along “y-” direction)* are required. The part-i is moved along negative direction for collision free disassembly operation. The interference matrices obtained for the Gear assembly along “X-, Y- and Z-” directions are as follows.

$$\begin{array}{c|ccccccc|}
X- & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
4 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
5 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
7 & 0 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}
\quad
\begin{array}{c|ccccccc|}
Y- & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
2 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
3 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
4 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
5 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
7 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\quad
\begin{array}{c|ccccccc|}
Z- & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
4 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
5 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
7 & 0 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}$$

Interference matrices along (X-, Y-, & Z-) directions for Gear assembly

The interference matrices obtained for the transmission assembly shown along “X-, Y- and Z-” directions are as follows.

$$\begin{array}{c|ccccccccccc|}
(X-) & A & B & C & D & E & F & G & H & J & K & L \\
A & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
B & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
C & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
D & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
E & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
F & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
G & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
H & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
J & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
K & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
L & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}
\quad
\begin{array}{c|ccccccccccc|}
(Y-) & A & B & C & D & E & F & G & H & J & K & L \\
A & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
B & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
C & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
D & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
E & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
F & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
H & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
J & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
K & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

$$\begin{array}{c|ccccccccccc|}
(Z-) & A & B & C & D & E & F & G & H & J & K & L \\
A & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
B & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
C & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
D & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
E & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
F & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
G & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
H & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
J & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
K & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
L & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}$$

Interference matrices along (X-, Y-, & Z-) directions for transmission assembly

There exist three categories of stability between a pair parts as stated. In the current section methods to extract each type of stability is discussed separately. The final stability matrix will be considered by merging all three matrices by updating with the higher level of stability condition.

3.3.3 Partial Stability Matrix Extraction

Laws of equilibrium and stability of physical objects states: If a normal from COG of a part towards gravitational force (“z-” direction) intersects its mating intersection surface then the part is stable over that surface. Figure 3.18 indicates partially stable part combinations, in which normal from part-A COG intersects common mating surface. Figure 3.19 shows unstable part combinations, in which normal from part-A COG does not intersect common mating surface.

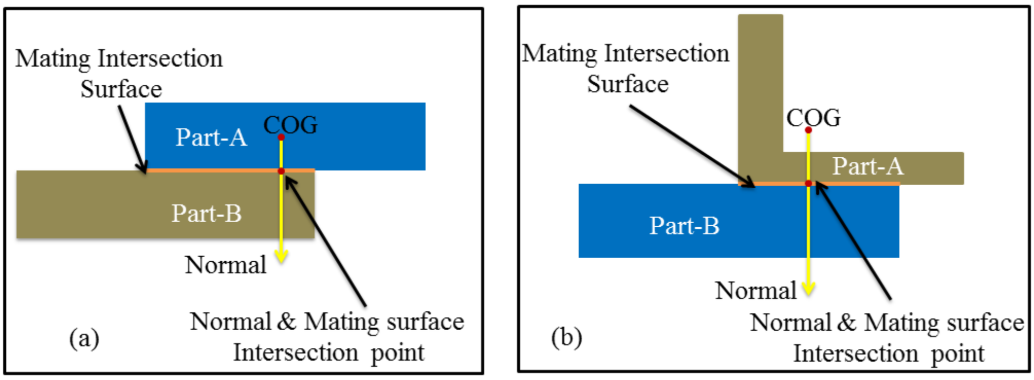


Figure 3.18: Partially stable assembly subsets

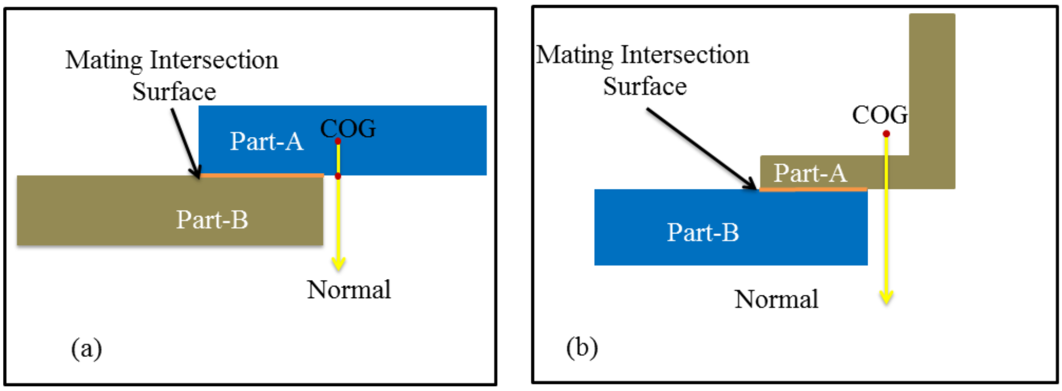


Figure 3.19: Instances of instable assembly subsets

When the normal from COG of part lies far the mating intersection surface, leads to moment and results instability of the part. In certain cases, few parts are stable on the mating surfaces, though the normal from COG does not intersect the mating surface. Example for such assembly sets are shown in Figure 3.20. For these parts, moment about the COG tends to zero to ensure the stability. Stability of the part can be identified by testing for the possibility rotation of over COG without interference.

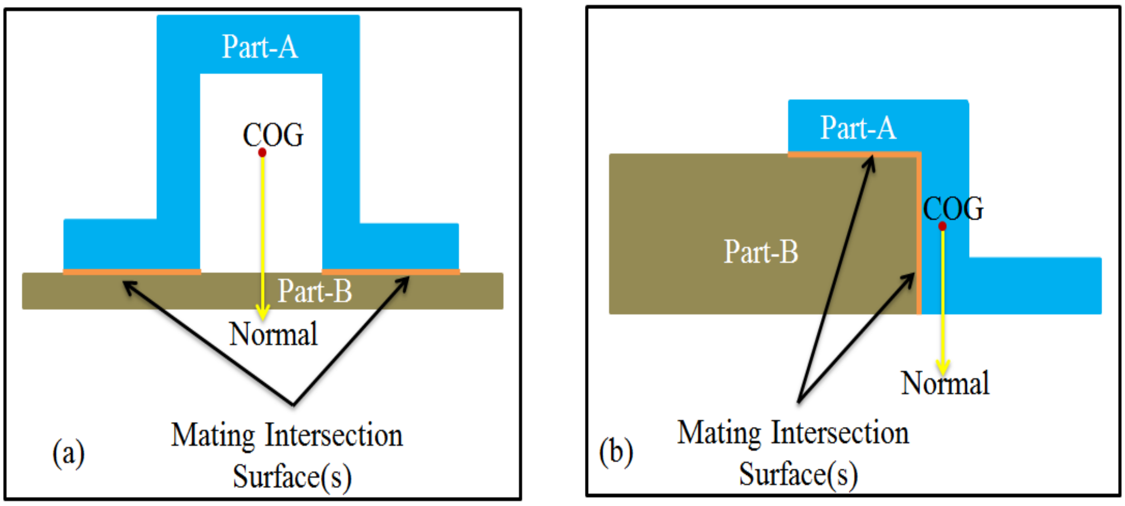


Figure 3.20: Partially stable assembly sets with non-intersected COG normal.

From the assembly subsets shown in figures 3.18,3.19 and 3.20 it is well observed that If a part can be rotated about an axes parallel to ground (XY-Plane) along X- direction and Y-direction through COG in both the orientations (clockwise or counter clock wise) leads to interference then that the part is stable about its mating surface. For axis-symmetric parts, testing along the symmetric axis direction is sufficient to generate the partial stability information. Figure 3.21 briefs the possible instance of stability due to resulted interferences while rotating the part in both clockwise ad counter clockwise orientations.

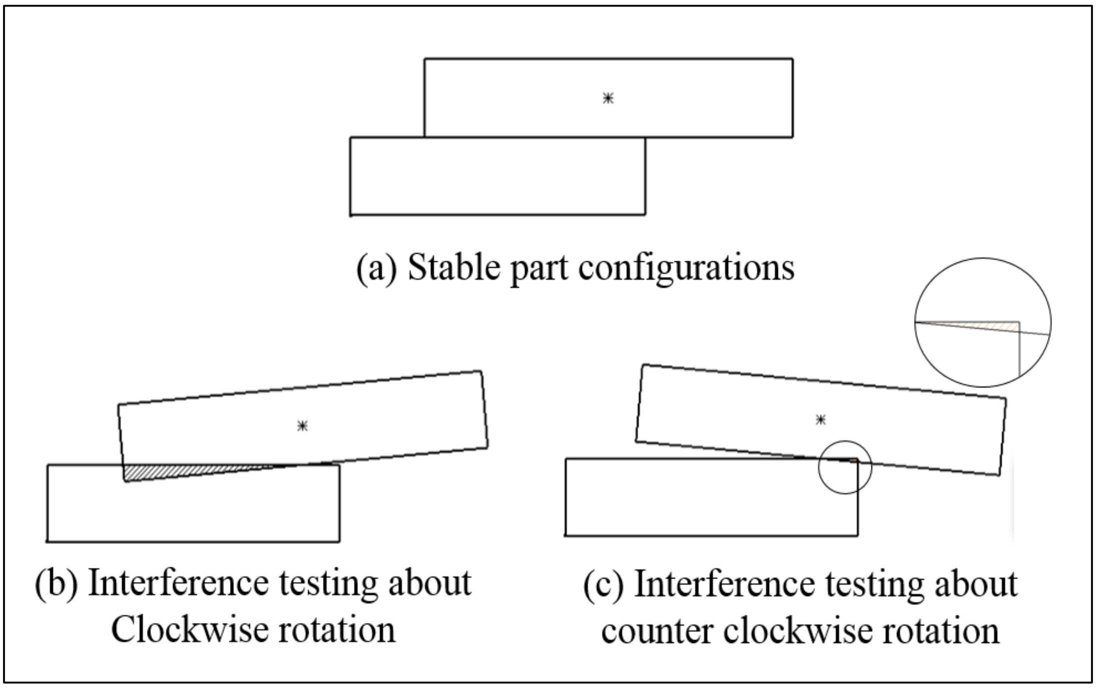
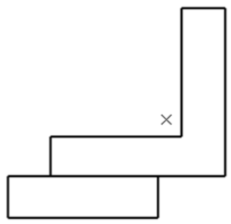
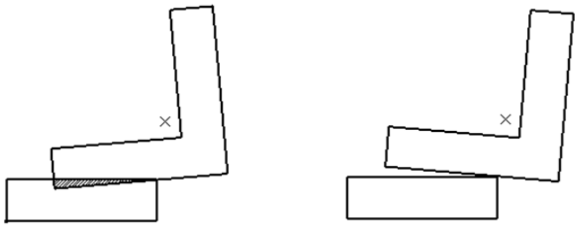


Figure 3.21: Test for partially stable assembly sets with intersected COG normal.



(a) Stable part configurations



(b) Interference testing about Clockwise rotation

(c) Interference testing about counter clockwise rotation

Figure 3.22: Test for partially stable assembly sets with non-intersected COG normal

Figure 3.22 briefs the possibility of instability due to clearance between the parts while rotated in counter clockwise orientation. Liaison matrix and interference matrix about “ z-“ orientation will reduce the further efforts of partial stability matrix extraction. Partial stability has to checked for the pair of parts for which liaison matrix element and the interference matrix element values must be “1” and “0” respectively i.e. indicating that the pair of parts are in contact and part-i (represented in row) cannot be disassembled along “z-“ direction in the presence of part-j(represented in column). An algorithm to extract partial stability matrix is mentioned below.

```
'Liaison, geometric feasibility and COG data are represented through lm(n,n), gfm(6,n,n) and cog(3)
Step 1: Consider an assembled product
Step 2: For i=1 to n
Step 3: For j=1 to n
Step 4:   Hide all primary parts other than part-i & part-j
Step 5:   stb1(i,j)=0
Step 6:   If (lm(i,j)=1 and gfm(6,i,j)=0) then
Step 7:         Obtain center of gravity coordinates for part-i
Step 8:         draw line through COG of "part-i" along "X" direction
Step 9:         rotate part-i about line in clockwise direction
Step 10:        perform contact analysis against "part-j"
Step 11:        cv1=resulted interference value
Step 12:         rotate part-i about line in counter clockwise direction
Step 13:                and perform contact analysis against "part-j"
Step 14:        cv2= resulted interference value
```

```
Step 15:        If (cv1<0 and cv2<0)
Step 16:          stb1(i,j)=1
Step 17:          Rotate part-i to bring back its original position
                End If
            End If
          End For
        End For
```

Flow chart is presented in Figure 3.23 for the partial stability extraction method stated above.

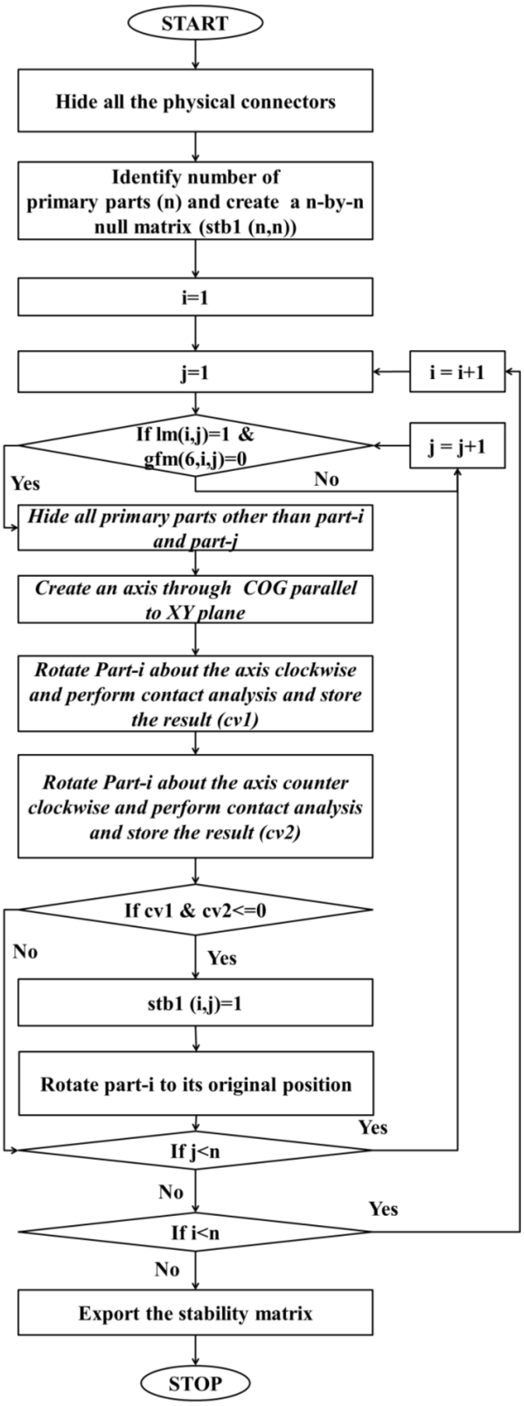


Figure 3.23: Partially stability matrix extraction method.

A program has been written in VB script to interface with CATIA v5 environment and the resulted partial stability matrix for the products shown in Figure 3.11 and 3.12 are given below.

$$
\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}
$$

Partial Stability matrix for Gear assembly

$$
\begin{array}{c|ccccccccccc|}
 & A & B & C & D & E & F & G & H & J & K & L \\
\hline
A & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
B & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
C & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
D & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
H & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
J & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
K & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
L & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
\hline
\end{array}
$$

Partial Stability matrix for transmission assembly

It is observed from the results, the liaison matrix is similar to partial stability matrix for gear assembly due the reason, each contact between the pair of parts establishing stability by supporting each other. However for transmission assembly, few liaisons do not offer the partial stability for certain pair of parts (A-D, E-F), since the stability matrix is not similar to the liaison matrix.

3.3.4 Permanent Stability Matrix Extraction (Due to Mating Features)

The part surface feature recognition is mainly dependent on the modelling methodology and the software interface compatibility to retrieve the information. Most of the advanced CAD softwares (CATIA V5, Solidworks, Pro E, Unigraphics, etc,) offers flexibility to users in feature modelling and data extraction. Parts connected by their surface features generally possess only one degree of freedom for assembly or disassembly operation. Hence, geometric feasibility matrices and liaison matrix data further minimizes the complexity and computation time in retrieving the permanent stability information due to mating features. The method involves in extracting part feature information (for example internal thread with a defined diameter and pitch) and tests for counter data with its mating part (External threading with same diameter and pitch) at the mating surface. An algorithm to extract permanent stability matrix is stated below.

```
'Liaison, geometric feasibility and COG data are represented through lm(n,n), gfm(6,n,n) and cog(3)
Step 1: Consider an assembled product
Step 2: n ← number of primary parts
Step 3: Identify and Hide all the connectors
Step 4:  For i=1 to n
Step 5: For j=1 to n
Step 6:   stb2(i,j)=0;
Step 7:   sum= gfm(1,i,j)+ gfm(2,i,j)+ gfm(3,i,j)+ gfm(4,i,j)+ gfm(5,i,j)+ gfm(6,i,j)
Step 8:   If  (lm(i,j)=1 and sum=1) then
Step 9:          Test for surface features (internal/external threading) on part-i
at the mating intersection surface
Step 10:         Test for surface features (internal/external threading)  on part-j
at the mating intersection surface
Step 11:          If  (there exist features with the similar pattern)
Step 12:                stb2(i,j)=2
                    End If
                  End If
                 End For
                End For
```

Flow chart is presented in Figure 3.24 for the permanent stability extraction method stated above.

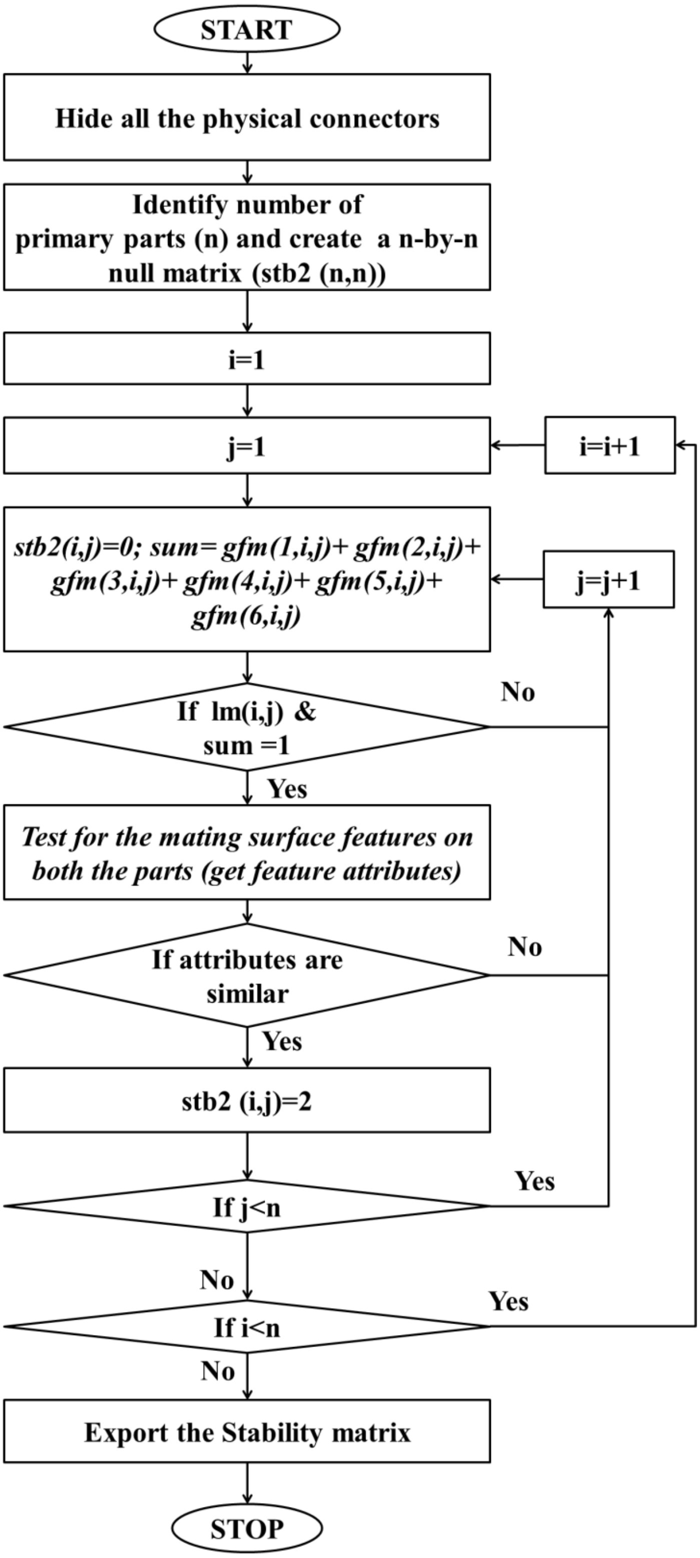


Figure 3.24: Permanent stability matrix extraction method (due to mating features)

Permanent stability matrix extracted through a program written VB scripting to interface with CATIA v5 environment. The resulted matrices for the assembled products shown in Figure 3.11 and 3.12 are given below

$$\begin{array}{c|ccccccc|} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Permanent Stability matrix for Gear assembly (due to mating features)

$$\begin{array}{c|ccccccccccc|} & A & B & C & D & E & F & G & H & J & K & L \\ \hline A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ H & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ J & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ K & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Permanent Stability matrix for transmission assembly (due to mating features)

Threading features on part-6 and part-7 to connect with part-1 resulted “2” for the pair of parts (1-6 and 1-7) in the permanent stability matrix for gear assembly. However for the transmission assembly there are no such pair of parts found connected by surface features and hence a null matrix is resulted.

3.3.5 Permanent Stability Matrix Extraction (Due to External Connectors)

Extraction of permanent stability matrix involved in identifying the connectors in the assembly model by their nomenclature and obtaining the pair of primary parts joined by the connector. Assembly contact analysis for a connector against all primary parts results pair of part those were connected by it, iterating the process for each connector resulting in permanent stability matrix. An algorithm to extract permanent stability matrix is stated below.

```
Step 1: Consider an assembled product
Step 2: Get the list of connectors by their nomenclature (m)
Step 3: For each connector k=1 to m
Step 4:         Hide all connectors other that connector-k
Step 5:         perform assembly contact analysis against all primary parts.
Step 6:         Get the pair of primary parts connected by connector-k
Step 7:         Represent "3" in the stability matrix element for the pair of parts
                Stb3(i,j)=3 (For pair of parts part-i ad part-j)
Step 8: End For
```

Flow chart is presented in Figure 3.25 for the permanent stability extraction method stated above.

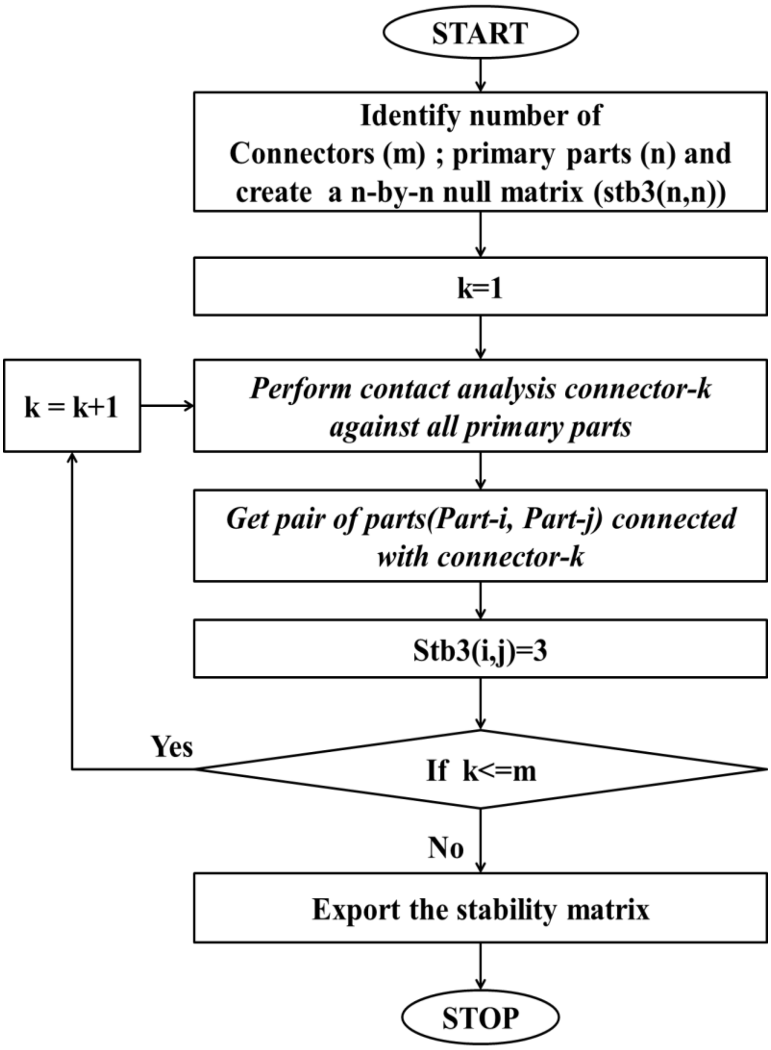


Figure 3.25: Permanent stability matrix extraction method (due to external connectors)

Permanent stability matrix due to external connectors is extracted through a program written in VB script to interface with CATIA v5 environment. The resulted matrices for the products shown in Figure 3.11 and 3.12 are given below.

$$
\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Permanent Stability matrix for Gear assembly due to external connectors

$$
\begin{array}{c|ccccccccccc|}
 & A & B & C & D & E & F & G & H & J & K & L \\
\hline
A & 0 & 0 & 3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\
B & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
C & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
D & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 0 & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\
F & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
H & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
J & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
L & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

Permanent Stability matrix for transmission assembly due to external connectors

Five set of attachments were used to connect the primary parts in the transmission assembly. For five pair of parts, “3” value is arrived in the resulted permanent stability matrix. However the gear assembly does not consist any physical connector and thus a null matrix is resulted.

The stability matrices are merged together to achieve final stability matrix, during the merging process, higher value elements will be switched in place of lower value. During this process most of the elements with “0 and 1” in the partial stability matrix will be replaced with “2 or 3”from the permanent stability matrix. The final stability matrix for the products shown in Figure 3.11 and 3.12 are given below.

$$\begin{array}{c|ccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & 0 & 1 & 0 & 1 & 1 & 2 & 2 \\
2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
7 & 2 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

Combined Stability matrix for Gear assembly

$$\begin{array}{c|ccccccccccc|}
 & A & B & C & D & E & F & G & H & J & K & L \\
A & 0 & 0 & 3 & 3 & 0 & 0 & 1 & 0 & 0 & 1 & 3 \\
B & 0 & 0 & 1 & 0 & 3 & 0 & 1 & 1 & 1 & 0 & 0 \\
C & 3 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
D & 3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 0 & 3 & 1 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\
F & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
H & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
J & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
K & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
L & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}$$

Combined Stability matrix for transmission assembly

3.3.6 Mechanical Feasibility Matrix Extraction

In order to test the mechanical feasibility, firstly number of connectors must be identified for this purpose the stability matrix can be used. Using the stability matrix, pair of parts with element value “3” must be acknowledged and contact analysis should be done for both the parts against all the physical connectors. The common connectors are grouped into one set. While placing set of connectors, any part in the existed assembly other than those pair of parts can offer collision.

Each connector will be tested for geometric feasibility along six directions in the presence of both the mating parts resulting one feasible orientation. Each connector will be tested for the geometric feasibility in the presence of other primary parts by using their bound box coordinates.

The below algorithm uses the stability matrix data and retrieves the possible single direction to join the connector with its mating primary parts and find the infeasibility along the direction in the presence of other primary parts.

```
'Liaison, geometric feasibility and COG data are represented through lm(n,n),
gfm(6,n,n) and cog(3)
Step 1: Consider an assembled product
Step 2: n←number of primary parts
Step 3: Generate a null matrix of size (n-by-n-by-n)
Step 4: for i = 1 to n
Step 5:  for j= 1 to n
Step 6:   if stb(i,j) = 3
Step 7:          hide all primary parts other than part-i and part-j
Step 8:          perform contact analysis for part-i against all connectors
Step 9:          perform contact analysis for part-j against all connectors
Step 10:         con(m)← store common connectors
Step 11:         for k=1 to m
Step 12:           for zz=1 to 6
Step 13:              geometric feasibility of connector against primary parts-i & j
                    End For
Step 12:            unhide all primary parts and hide i and j
Step 13:            for l = 1 to n
Step 14:              dir= zz "zz is the single possible direction for connector
Step 15:         check geometrical feasibility of connector "k" in the existence of "part-
l" along "dir"
Step 16:              If not feasible
Step 17:                 mfm(i,j,l)= 1; l= l+1
                      End If
                    End for
                  End For
                 End If
              End for
             End for
```

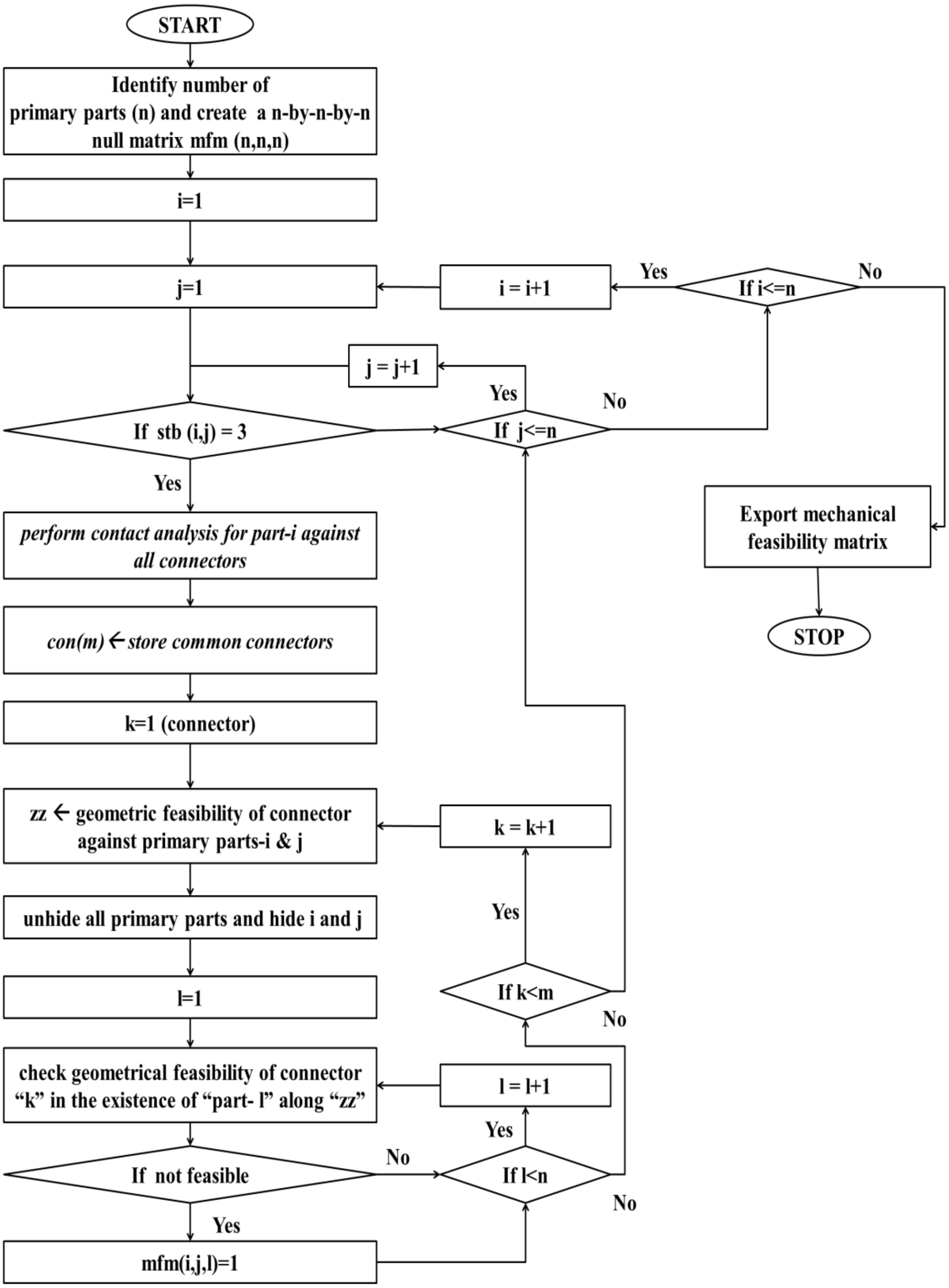
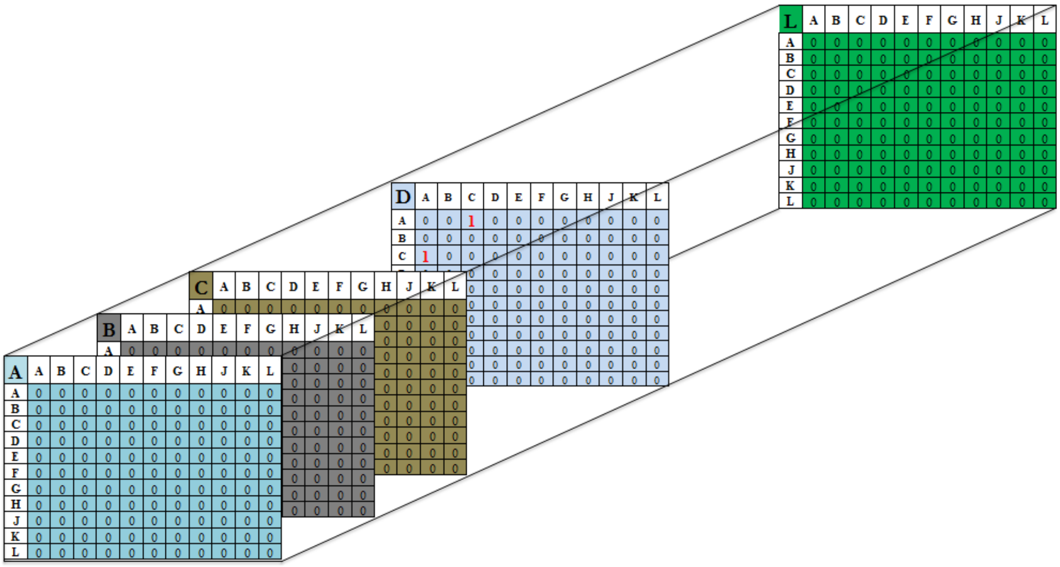


Figure 3.26: Mechanical feasibility matrix extraction method

Mechanical feasibility matrix extraction method flow chart is shown in Figure 3.26, and a program is written in VB script to extract the mechanical feasibility matrix for the transmission assembly shown in Figure 3.12 is given below.



Mechanical feasibility matrix for transmission assembly

Mechanical feasibility matrix for gear assembly is not required due to the reason, the absence of physical connectors.

3.3.7 Part Bounding Box Coordinates and Part Weight Extraction

Part bounding box coordinates are used to test the geometric feasibility between pair of parts to create interference matrices and also used to test the assembly energy and directional changes during optimality testing. Most of the CAD softwares with 3D solid modelling capabilities offer direct compatibility to extract axis aligned bounding box coordinates. To retrieve axis aligned bounding box coordinates, extension modules are commonly used. However lowest an higher X, Y and Z coordinates of a part can be obtained by .STL (STereoLithography) file format file of a 3D solid CAD model. Part bounding box coordinates extracted for the assembled products shown in Figure 3.11 and 3.12 are listed in Table 3.1 and Table 3.2 respectively below.

Table 3.1: Part bounding box coordinates for Gear assembly

Part No.	x1 (mm)	y1 (mm)	z1 (mm)	x2 (mm)	y2 (mm)	z2 (mm)
1	-10	-150	-10	10	150	10
2	-20	-10	-20	20	10	20
3	-50	-10	-50	50	10	50
4	-35	-100	-35	35	-80	35
5	-35	80	-35	35	100	35
6	-12	-160	-12	12	-140	12
7	-12	140	-12	12	160	12

Table 3.2 Part bounding box coordinates for Transmission assembly

Part No.	x1 (mm)	y1 (mm)	z1 (mm)	x2 (mm)	y2 (mm)	z2 (mm)
A	-77	-75	-77	77	90	77
B	-63	-95	-63	63	-34	63
C	-72	-85	-72	72	-60	72
D	-77	-85	-77	77	-75	77
E	-81	-100	-81	81	-75	81
F	-30	-108	-30	30	-100	30
G	-62	-43	-62	62	45	62
H	-63	-43	-63	63	85	63
J	-50	-46	-50	50	98	50
K	-73	60	-73	73	75	73
L	-77	82	-77	77	95	77

Weight of each primary part is required to compute the assembly energy due to part movements during the assembly operation in a specified direction. The multiplication product of geometric volume and density of material used for the part is considered as mass of part. Once material properties are assigned to solid models in CAD environment, mass of the part can be extracted through inertia properties option. Weights of each part in assembled products shown in Figure 3.11 and 3.12 are listed in Table 3.3 and Table 3.4 respectively below.

Table 3.3: Weight of independent parts shown in Gear assembly

Part No.	Weight (g)
1	703.7481703
2	148.1575095
3	896.4719058
4	555.5906607
5	555.5906608
6	64.94237502
7	64.94237502

Table 3.4: Weights of independent parts shown in Transmission assembly

Part No.	Weight (g)
A	6263.852574
B	628.01499
C	760.4184177
D	1365.27145
E	1985.483478
F	46.75103415
G	1621.74217
H	2109.202842
J	4195.654216
K	503.7355324
L	1559.348401

3.3.8 Some Appropriate Tools and Their Capabilities

In order to extract the assembly attribute information from CAD software through the above stated mechanisms, the software must be equipped with basic part design and assembly design/representation modules with several capabilities. CATIA V5 software with VB scripting interfacing is used in the current research, basic requirements of various CAD softwares and their capabilities to extract assembly attribute information is illustrated in below tables. Table 3.5 lists basic requirements and their purpose in assembly attribute extraction. Table 3.6 lists most used mechanical design softwares and their compatibility to extract the assembly attribute information.

Table 3.5: Basic requirement of cad softwares for assembly attribute extraction

S. No.	Assembly Attributes	Basic requirements of CAD software	Purpose
1	Liaison Matrix	Visualization filters	To hide/unhide connectors
		Assembly Clash check	To detect part contacts
2	Bounding Box Coordinates	Stock material measurement (or) .STL conversion capability	To detect distances between parts along all principal axes (For interference checking)
3	Interference matrices	Part transformations	Part trajectories
		Assembly Clash check	Collision detection
		Visualization filters	To hide/unhide connectors and primary parts
4	Partial Stability matrix	Inertia Properties	COG detection
		3D-Rotation	Stability check
		Assembly Clash check	To detect part contacts
5	Permanent Stability matrix (Surface features)	Assembly feature's properties recognition	To detect similar features at contact faces.
6	Permanent Stability matrix (External connectors)	Part detection by nomenclature	Connectors identification
		Assembly Clash check	To detect part contacts
7	Mechanical feasibility Matrix	Part transformations	Part trajectories
		Assembly Clash check	Collision detection
		Visualization filters	To hide/unhide connectors and primary parts

Table 3.6: Mechanical CAD softwares compatibility for assembly attribute extraction

S. No.	Software	Publisher	Direct Compatibility	API Compatibility
1.	Autodesk Inventor	AUTODESK INC.	3D solid Modelling and Assembly modelling.	C#
2.	Creo parametric (Pro Engineer)	Parametric Technology Corporation (PTC)	3D solid Modelling and Assembly modelling.	VB API/C++
3.	CATIA	Dassault Systems	3D solid Modelling and Assembly modelling.	CATScript / VBScript
4.	NX (UG or Unigraphics)	Siemens	3D solid Modelling and Assembly modelling.	C/C++
5.	Solidworks	Dassault Systems	3D solid Modelling and Assembly modelling.	Visual Basic

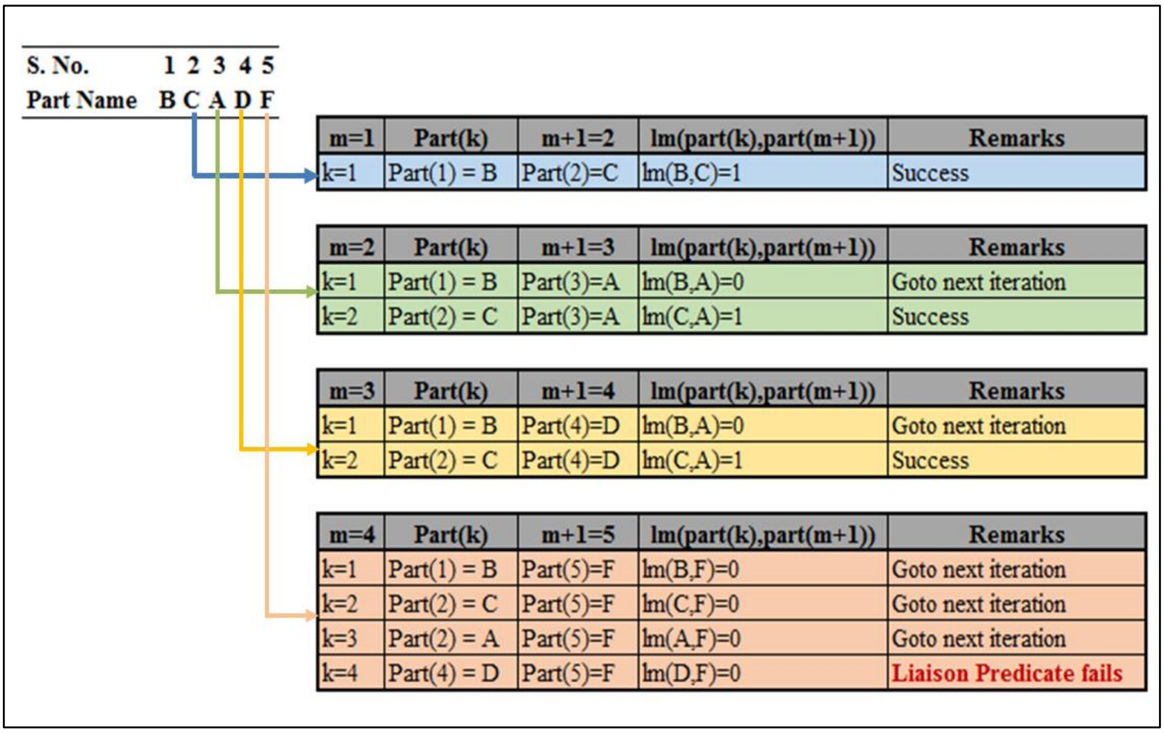
3.3.9 Liaison Predicate Testing

In order to join a part to an existent assembly subset, it must exhibit at least one contact with any of the part from the subset. Liaison matrix elements for the joining part with respect to all the parts existed in the assembly subset are considered for this purpose. An algorithm for the liaison predicate testing is mentioned below.

```
'Part-(m+1) is Appending part ,
"m" is length of the existent assembly subset,
Step 1: For k=1 to m
Step 2:   sum=0
Step 3:   If (lm(part(k),part(m+1))=1)
Step 4:             Part-(m+1) exhibits liaison with part (k)
              sum=1 & exit for loop
              End If
          End for
Step 5:  If (sum=1)
              Part-(m+1) exhibits liaison with one of existent part
              End If
Step 6:  If (sum=0)
              Part-(m+1) does not exhibit liaison with any part
             End If
```

- *The above procedure should be iterated for each part in the assembly subset/assembly sequence to ensure the sequence obeys liaison predicate.*

An assembly subset (B-C-A-D-F) is considered to explain the liaison predicate testing. Detailed working methodology of liaison predicate testing is illustrated in Figure 3.27 and Figure 3.28.



S. No.	**1**	**2**	**3**	**4**	**5**
Part Name	**B**	**C**	**A**	**D**	**F**

m=1	**Part(k)**	**m+1=2**	**lm(part(k),part(m+1))**	**Remarks**
k=1	Part(1) = B	Part(2)=C	lm(B,C)=1	Success

m=2	**Part(k)**	**m+1=3**	**lm(part(k),part(m+1))**	**Remarks**
k=1	Part(1) = B	Part(3)=A	lm(B,A)=0	Goto next iteration
k=2	Part(2) = C	Part(3)=A	lm(C,A)=1	Success

m=3	**Part(k)**	**m+1=4**	**lm(part(k),part(m+1))**	**Remarks**
k=1	Part(1) = B	Part(4)=D	lm(B,A)=0	Goto next iteration
k=2	Part(2) = C	Part(4)=D	lm(C,A)=1	Success

m=4	**Part(k)**	**m+1=5**	**lm(part(k),part(m+1))**	**Remarks**
k=1	Part(1) = B	Part(5)=F	lm(B,F)=0	Goto next iteration
k=2	Part(2) = C	Part(5)=F	lm(C,F)=0	Goto next iteration
k=3	Part(2) = A	Part(5)=F	lm(A,F)=0	Goto next iteration
k=4	Part(4) = D	Part(5)=F	lm(D,F)=0	**Liaison Predicate fails**

Figure 3.27 Liaison predicate testing methodology illustration

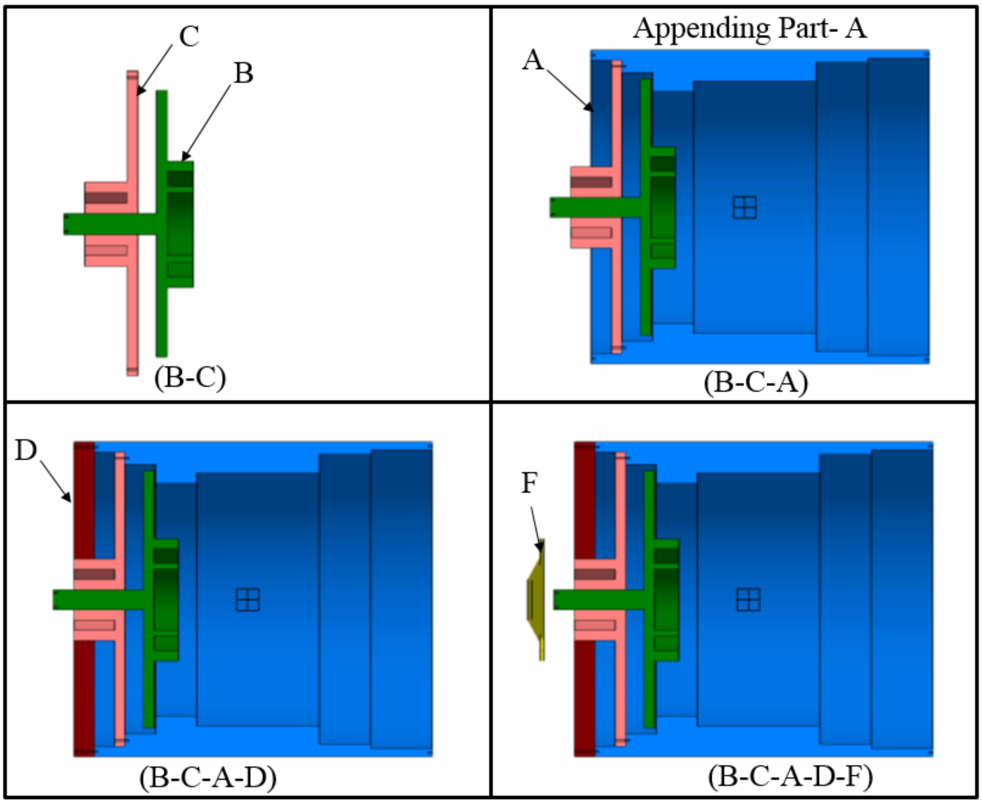


Figure 3.28: Liaison predicate testing demonstration for an assembly subset

3.3.10 Geometrical Feasibility Predicate Testing

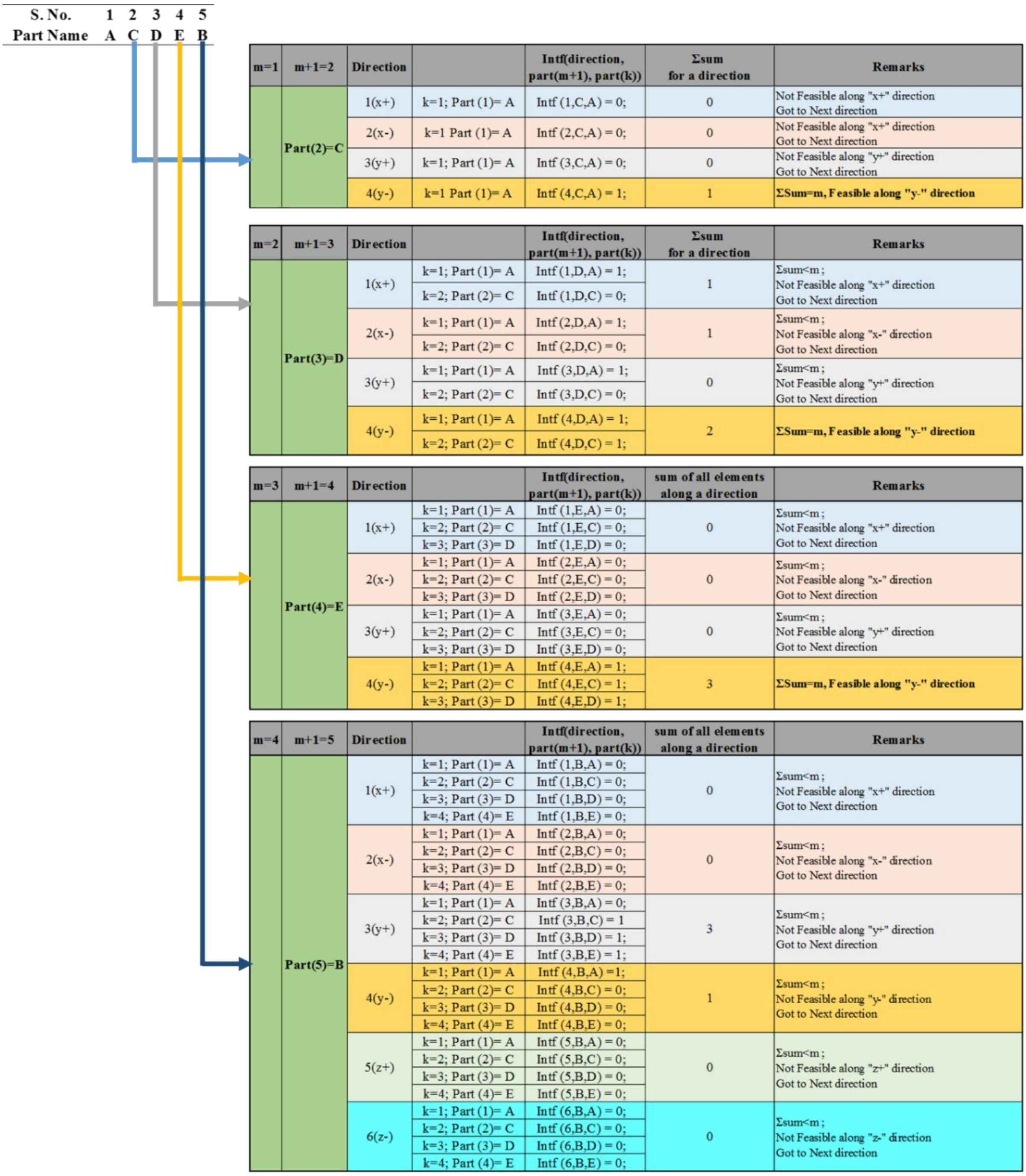
Feasibility of joining a part to an existent subassembly/assembly subset has to be tested for all directions to achieve collision free path. No collision free path for any part in the assembly sequence indicates its unfeasibility. The below proposed method tests whether appending part can be joined along a specific direction during the presence of each part existed in the assembly subset using interference matrix elements. Failing in a direction will leads to check for the next direction till it completes all the six directions.

An algorithm for geometrical feasibility predicate testing is mentioned below.

```
'Part-(m+1) is Appending part
"m" is length of the existent assembly subset,
Step 1: sum1=0
Step 2: For dir = 1 to 6 (successively represents x+, x-, y+, y-, z+ and  z-)
Step 3:         sum=0
Step 4:         For k=1 to m
Step 5:             sum= sum + Intf (dir, part(m+1), part(k))
                Next
Step 6:         If (sum= m)
Step 7:             Part-(m+1) can be disassemble along the direction.
Step 8:             Feasible direction ← dir
Step 9:             dir =6
                End If
Step 10:        If (dir = 6 and sum < m)
Step 11:            Not feasible is any direction
Step 12:        End If
            Next
```

- *The above procedure should be iterated for each part in the assembly subset/assembly sequence to ensure the sequence is geometrically feasible.*

The proposed method is employed on an assembly set (A-C-D-E-B) and detailed pictorial representation is indicated Figure 3.29.



S. No.	1	2	3	4	5
Part Name	A	C	D	E	B

m=1	m+1=2	Direction		Intf(direction, part(m+1), part(k))	Σsum for a direction	Remarks
	Part(2)=C	1(x+)	k=1; Part (1)= A	Intf (1,C,A) = 0;	0	Not Feasible along "x+" direction Got to Next direction
		2(x-)	k=1 Part (1)= A	Intf (2,C,A) = 0;	0	Not Feasible along "x+" direction Got to Next direction
		3(y+)	k=1; Part (1)= A	Intf (3,C,A) = 0;	0	Not Feasible along "y+" direction Got to Next direction
		4(y-)	k=1 Part (1)= A	Intf (4,C,A) = 1;	1	**ΣSum=m, Feasible along "y-" direction**

m=2	m+1=3	Direction		Intf(direction, part(m+1), part(k))	Σsum for a direction	Remarks
	Part(3)=D	1(x+)	k=1; Part (1)= A	Intf (1,D,A) = 1;	1	Σsum<m ; Not Feasible along "x+" direction Got to Next direction
			k=2; Part (2)= C	Intf (1,D,C) = 0;		
		2(x-)	k=1; Part (1)= A	Intf (2,D,A) = 1;	1	Σsum<m ; Not Feasible along "x-" direction Got to Next direction
			k=2; Part (2)= C	Intf (2,D,C) = 0;		
		3(y+)	k=1; Part (1)= A	Intf (3,D,A) = 1;	0	Σsum<m ; Not Feasible along "y+" direction Got to Next direction
			k=2; Part (2)= C	Intf (3,D,C) = 0;		
		4(y-)	k=1; Part (1)= A	Intf (4,D,A) = 1;	2	**ΣSum=m, Feasible along "y-" direction**
			k=2; Part (2)= C	Intf (4,D,C) = 1;		

m=3	m+1=4	Direction		Intf(direction, part(m+1), part(k))	sum of all elements along a direction	Remarks
	Part(4)=E	1(x+)	k=1; Part (1)= A	Intf (1,E,A) = 0;	0	Σsum<m ; Not Feasible along "x+" direction Got to Next direction
			k=2; Part (2)= C	Intf (1,E,C) = 0;		
			k=3; Part (3)= D	Intf (1,E,D) = 0;		
		2(x-)	k=1; Part (1)= A	Intf (2,E,A) = 0;	0	Σsum<m ; Not Feasible along "x-" direction Got to Next direction
			k=2; Part (2)= C	Intf (2,E,C) = 0;		
			k=3; Part (3)= D	Intf (2,E,D) = 0;		
		3(y+)	k=1; Part (1)= A	Intf (3,E,A) = 0;	0	Σsum<m ; Not Feasible along "y+" direction Got to Next direction
			k=2; Part (2)= C	Intf (3,E,C) = 0;		
			k=3; Part (3)= D	Intf (3,E,D) = 0;		
		4(y-)	k=1; Part (1)= A	Intf (4,E,A) = 1;	3	**ΣSum=m, Feasible along "y-" direction**
			k=2; Part (2)= C	Intf (4,E,C) = 1;		
			k=3; Part (3)= D	Intf (4,E,D) = 1;		

m=4	m+1=5	Direction		Intf(direction, part(m+1), part(k))	sum of all elements along a direction	Remarks
	Part(5)=B	1(x+)	k=1; Part (1)= A	Intf (1,B,A) = 0;	0	Σsum<m ; Not Feasible along "x+" direction Got to Next direction
			k=2; Part (2)= C	Intf (1,B,C) = 0;		
			k=3; Part (3)= D	Intf (1,B,D) = 0;		
			k=4; Part (4)= E	Intf (1,B,E) = 0;		
		2(x-)	k=1; Part (1)= A	Intf (2,B,A) = 0;	0	Σsum<m ; Not Feasible along "x-" direction Got to Next direction
			k=2; Part (2)= C	Intf (2,B,C) = 0;		
			k=3; Part (3)= D	Intf (2,B,D) = 0;		
			k=4; Part (4)= E	Intf (2,B,E) = 0;		
		3(y+)	k=1; Part (1)= A	Intf (3,B,A) = 0;	3	Σsum<m ; Not Feasible along "y+" direction Got to Next direction
			k=2; Part (2)= C	Intf (3,B,C) = 1		
			k=3; Part (3)= D	Intf (3,B,D) = 1;		
			k=4; Part (4)= E	Intf (3,B,E) = 1;		
		4(y-)	k=1; Part (1)= A	Intf (4,B,A) =1;	1	Σsum<m ; Not Feasible along "y-" direction Got to Next direction
			k=2; Part (2)= C	Intf (4,B,C) = 0;		
			k=3; Part (3)= D	Intf (4,B,D) = 0;		
			k=4; Part (4)= E	Intf (4,B,E) = 0;		
		5(z+)	k=1; Part (1)= A	Intf (5,B,A) = 0;	0	Σsum<m ; Not Feasible along "z+" direction Got to Next direction
			k=2; Part (2)= C	Intf (5,B,C) = 0;		
			k=3; Part (3)= D	Intf (5,B,D) = 0;		
			k=4; Part (4)= E	Intf (5,B,E) = 0;		
		6(z-)	k=1; Part (1)= A	Intf (6,B,A) = 0;	0	Σsum<m ; Not Feasible along "z-" direction Got to Next direction
			k=2; Part (2)= C	Intf (6,B,C) = 0;		
			k=3; Part (3)= D	Intf (6,B,D) = 0;		
			k=4; Part (4)= E	Intf (6,B,E) = 0;		

Figure 3.29: Geometric feasibility predicate testing methodology illustration

Although parts (C, D, E) have feasible directions to assemble successively to part-A (A, A-C, A-C-D), part-B failed to assemble (A-C-D-E) due to nonexistence of collision free path. Cut Section of (A-C-D-E-B) is shown in Figure 3.30, in which Part-A doesn't allow to assemble part-B directions other than "y-" , However, Parts(C,D,E) do not allow in the "y-" direction. Hence Part-B is not feasible to join (A-C-D-E) in any possible direction.

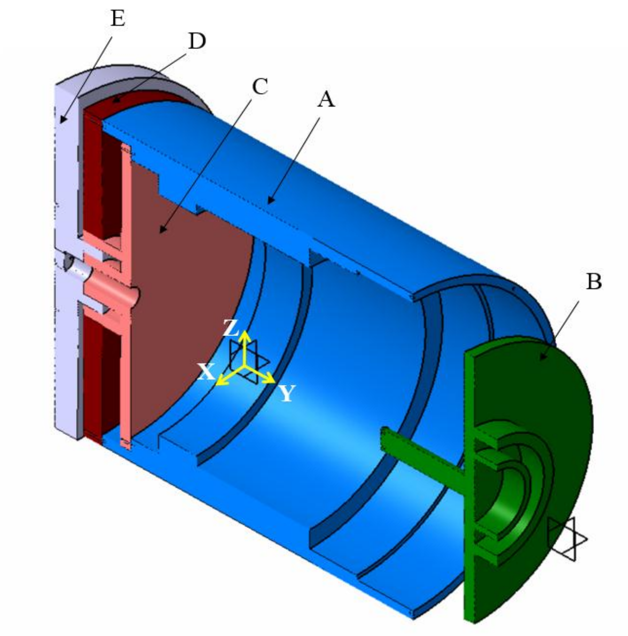


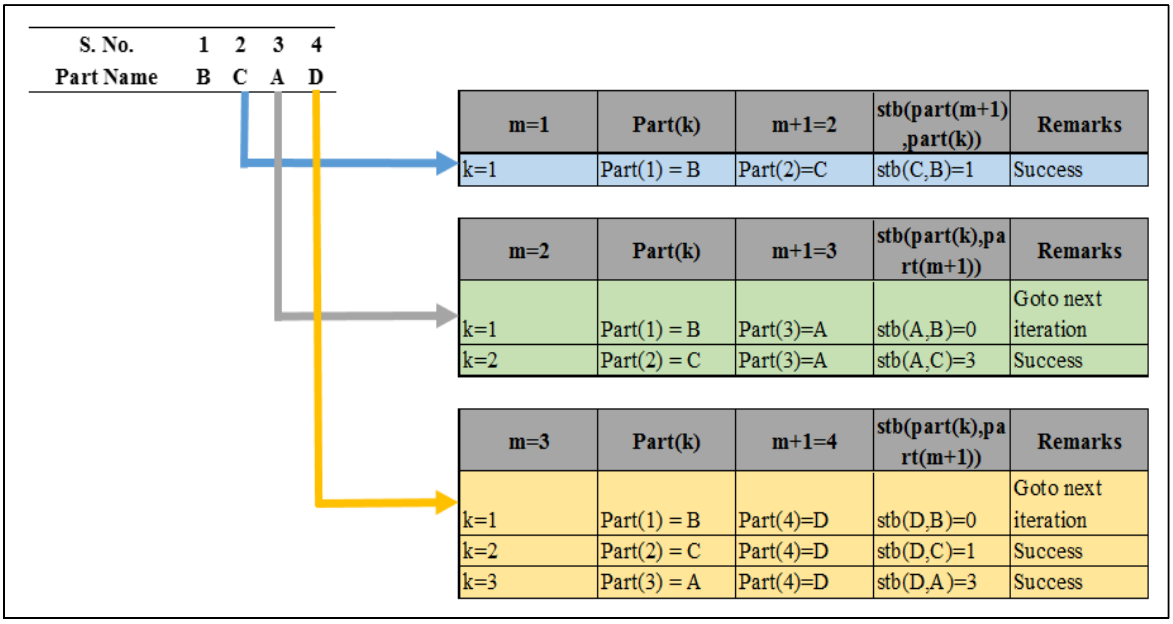
Figure 3.30: Geometrical feasibility testing demonstration for an assembly subset

3.3.11 Stability Predicate Testing

Stability predicate testing is most similar to liaison predicate testing to check whether appending part is stable by means of contact/surface-features/usage of external connectors with respect to the mating parts present on the existent assembly subset. Combined stability matrix is considered for this purpose, an algorithm to test the stability predicate testing is mentioned below.

```
‘Part-(m+1) is Appending part
“m” is length of the existent assembly subset,
Step 1: For k=1 to m
Step 2:         If (part(m+1), stb(part(k))>=1)
Step 3:                 Part-(m+1) is stable with respect to part (k)
Step 4:                 k=m & exit for loop
                End If
Step 5:         If (k=m and part(m+1), stb(part(k))=0)
Step 6:                 Part-(m+1) does not exhibits stability with respect to any part
                End If
        End For
```

Generally stability predicate testing will be done after liaison predicate test, hence liaison predicate test failed sequences will not appear for stability testing. The proposed stability predicate testing method is employed on an assembly set (B-C-A-D) an detailed pictorial representation is indicated Figure 3.31 and Figure 3.32.



S. No.	1	2	3	4
Part Name	B	C	A	D

m=1	Part(k)	m+1=2	stb(part(m+1),part(k))	Remarks
k=1	Part(1) = B	Part(2)=C	stb(C,B)=1	Success

m=2	Part(k)	m+1=3	stb(part(k),part(m+1))	Remarks
k=1	Part(1) = B	Part(3)=A	stb(A,B)=0	Goto next iteration
k=2	Part(2) = C	Part(3)=A	stb(A,C)=3	Success

m=3	Part(k)	m+1=4	stb(part(k),part(m+1))	Remarks
k=1	Part(1) = B	Part(4)=D	stb(D,B)=0	Goto next iteration
k=2	Part(2) = C	Part(4)=D	stb(D,C)=1	Success
k=3	Part(3) = A	Part(4)=D	stb(D,A)=3	Success

Figure 3.31: Stability predicate testing methodology illustration

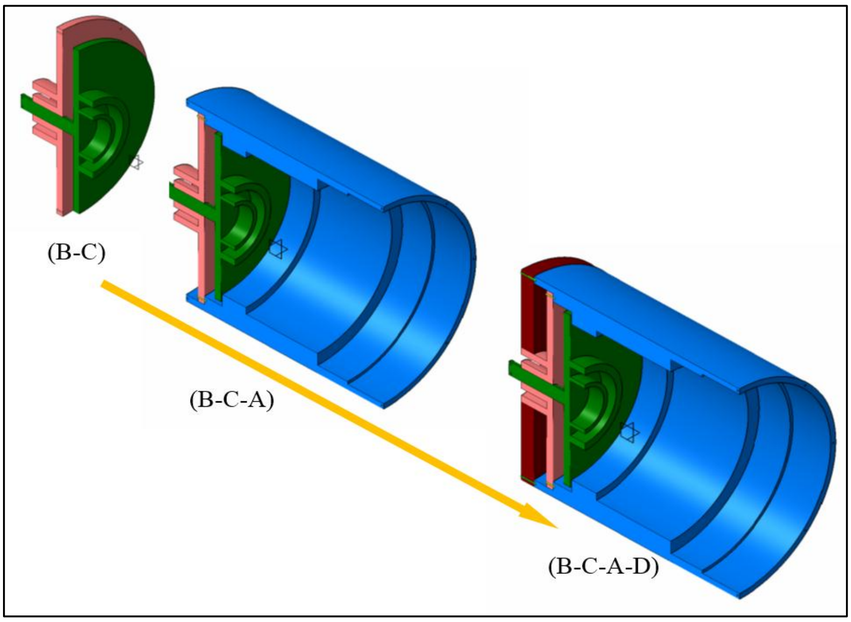


Figure 3.32: Stability predicate testing demonstration for an assembly subset

Part –C is partially stable by its contact with part-B and when Part-A is joined with physical connectors to part –C of subset (B-C). Similarly, Part-D is connected to part-A with physical connectors thus offers a stable assembly subset (B-C-A-D).

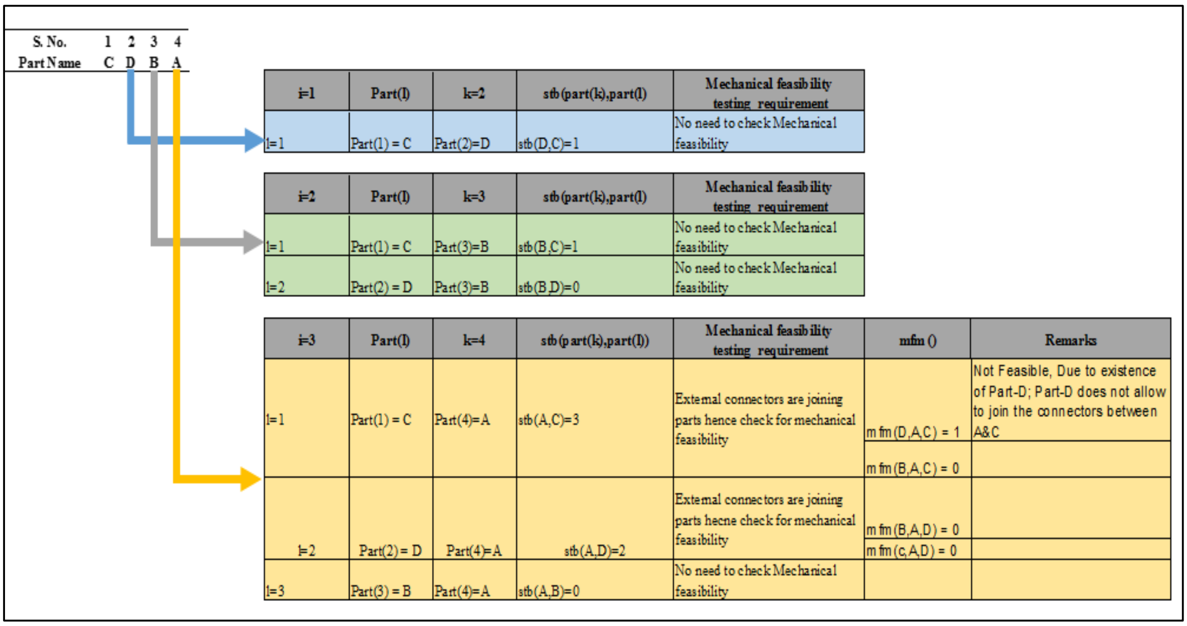
3.3.12 Mechanical Feasibility Predicate Testing

Geometric feasibility of those connectors with respect to all existent parts are to be tested, when the appending part is joined to any existent part in the assembly by means of connectors. Element value of stability matrix indicate type of connection between pair of parts, based on it further need of mechanical feasibility testing requirement is decided and performed. An algorithm to test mechanical feasibility predicate testing is stated below.

```
'Part-(m+1) is Appending part
"m" is length of the existent assembly subset,
Step 1: sum1=0
Step 2:      For k=1 to m
Step 3:          If ( stb (part(k),part(m+1))=3) then
Step 4:              For l=1 to k
Step 5:                  If (mfm (part (l), part (k), part (m+1)) =1)
Step 6:                      Mechanically not feasible; l=k & k=m
                         End If
                     Next
                 End If
Step 7:          If ( k=m and stb (part(k),part(m+1))≠3) then
Step 8:              Mechanically not feasible;
                 End If
             Next
```

- *The above procedure should be iterated for each part in the assembly subset/assembly sequence to ensure the sequence is mechanical feasible.*

The proposed method is implemented on an assembly subset from transmission assembly and its detailed illustration is shown in Figure 3.33 and 3.34.



S. No.	1	2	3	4
Part Name	C	D	B	A

i=1	Part(l)	k=2	stb(part(k),part(l))	Mechanical feasibility testing requirement
l=1	Part(1) = C	Part(2)=D	stb(D,C)=1	No need to check Mechanical feasibility

i=2	Part(l)	k=3	stb(part(k),part(l))	Mechanical feasibility testing requirement
l=1	Part(1) = C	Part(3)=B	stb(B,C)=1	No need to check Mechanical feasibility
l=2	Part(2) = D	Part(3)=B	stb(B,D)=0	No need to check Mechanical feasibility

i=3	Part(l)	k=4	stb(part(k),part(l))	Mechanical feasibility testing requirement	mfm ()	Remarks
l=1	Part(1) = C	Part(4)=A	stb(A,C)=3	External connectors are joining parts hence check for mechanical feasibility	m fm (D,A,C) = 1	Not Feasible, Due to existence of Part-D; Part-D does not allow to join the connectors between A&C
					m fm (B,A,C) = 0	
l=2	Part(2) = D	Part(4)=A	stb(A,D)=2	External connectors are joining parts hecne check for mechanical feasibility	m fm (B,A,D) = 0	
					m fm (c,A,D) = 0	
l=3	Part(3) = B	Part(4)=A	stb(A,B)=0	No need to check Mechanical feasibility		

Figure 3.33: Mechanical Feasibility predicate testing illustration

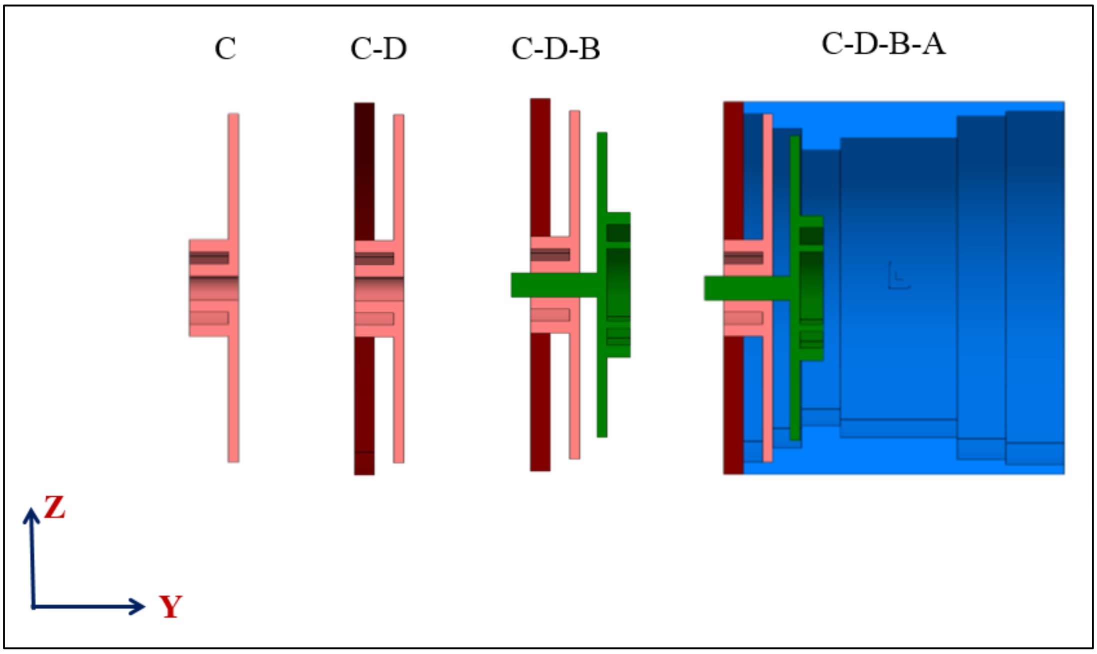


Figure 3.34 Mechanical feasibility predicate testing demonstration for an assembly subset

While joining part-A to assembly subset (C-D-B), part-A is connected to part-C and Part-D by means of physical connectors. Connectors between part-A and part-D can be joined without any collision with respect (part-B and part-C). However part-D offers collision to place the connectors between part-A and part-C, hence C-D-B-A is mechanically in feasible assembly subset.

3.4 Summary

This chapter presents the discussion of product information and their extraction methods to perform assembly sequence generation efficiently. The main purpose of this chapter is to provide the detail description of assembly attributes to perform assembly predicate testing and describe the methodology of assembly information extraction and assembly predicate testing. In the coming chapter influence of assembly predicate consideration on characteristics of optimal assembly sequence generation is discussed.

Chapter 4

ASSEMBLY SEQUENCE GENERATION WITH PREDICATE CONSIDERATION

4.1 Overview

Achieving a feasible assembly sequence involve multiple assembly predicate considerations. The necessity of considering all the assembly predicates is studied on different assembly configurations. The influence of ignoring an assembly predicate on various aspects of assembly sequence planning is discussed in detailed in this chapter. Most of the cited research literature on optimal assembly sequence generation methods considered liaison predicate and geometrical feasibility predicates, due to their significant contribution to ensure the quality of assembly sequence. In this chapter influence of stability and mechanical feasibility predicate consideration is discussed in detail.

4.2 Influence of Stability Predicate Consideration

Stability predicate is used to make sure that the appending part does not lose its contact with its mating parts at all next phases of assembly process once it is positioned. A part can exhibit stability either by support offered by its mating part or by use of external physical connector as discussed.

Most of the part joining operations such as part insertions ensures stability for several assembly configurations, where all the parts movements are possible in single direction either along positive or negative orientations. Few assemblies belong to such category are shown in Figure 4.1 and 4.2.

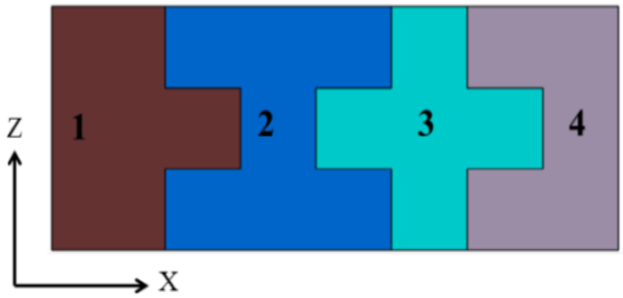


Figure 4.1: 4-Part block assembly

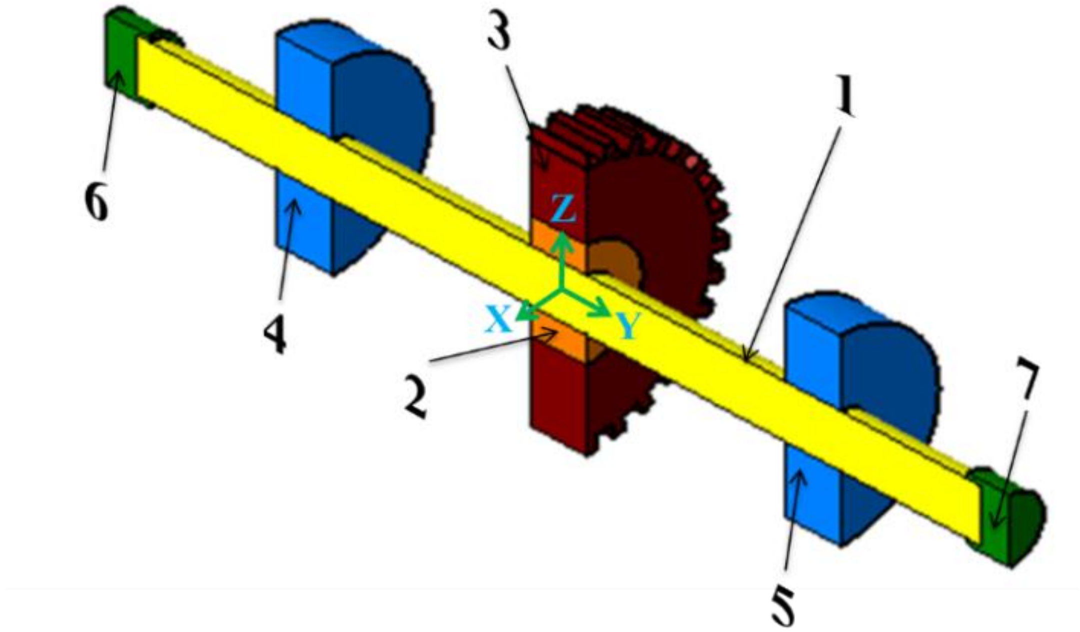


Figure 4.2: Cut a way of a gear assembly

For the products shown in Figure 4.1 and 4.2 set of all possible assembly sequences are generated by considering and ignoring stability predicate and the observations are listed in table 4.1. A system with (i7-3770 CPU @ 3.40 GHz, 2.00 GB RAM) configuration settings is used to test the performance.

Table 4.1: Stability predicate test outcomes for products shown in Figure 4.1 and 4.2

Assembly	**Number of Parts**	**Number of assembly sequences**		**Computational Time (sec.)**		**Number of Unstable assembly sequences**
		By Considering Stability Predicate	By Ignoring Stability Predicate	By Considering Stability Predicate	By Ignoring Stability Predicate	
Figure 4.1	4	8	8	1.856	1.242	Nil
Figure 4.2	7	64	64	3.108	2.218	Nil

While analysing the products shown in Figure 4.1 and 4.2, it is observed that the number of possible assembly sequences are same in both situations. The liaison matrix and the stability matrix for these assemblies exhibit similar characteristics and thus all the liaison predicate qualified subsets also qualify stability predicate. The physical meaning is, each liaison between any two components establishes stability and ensures that the assembly subset is stable for further assembly operations. The liaison matrix and combined stability matrix are analysed in Table 4.2.

Table 4.2: Observations based on liaison and stability matrices

Assembly	**Number of Parts**	**Liaison matrix**	**Stability matrix**	**Remarks**
Figure 4.1	4	$\begin{array}{cc} & \begin{matrix}1&2&3&4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}0&1&0&0\\1&0&1&0\\0&1&0&1\\0&0&1&0\end{bmatrix}\end{array}$	$\begin{array}{cc} & \begin{matrix}1&2&3&4\end{matrix} \\ \begin{matrix}1\\2\\3\\4\end{matrix} & \begin{bmatrix}0&1&0&0\\1&0&1&0\\0&1&0&1\\0&0&1&0\end{bmatrix}\end{array}$	For every lm(i,j)>0 stb(i,j)>0
Figure 4.2	7	$\begin{array}{cc} & \begin{matrix}1&2&3&4&5&6&7\end{matrix} \\ \begin{matrix}1\\2\\3\\4\\5\\6\\7\end{matrix} & \begin{bmatrix}0&1&0&1&1&1&1\\1&0&1&0&0&0&0\\0&1&0&0&0&0&0\\1&0&0&0&0&0&0\\1&0&0&0&0&0&0\\1&0&0&0&0&0&0\\1&0&0&0&0&0&0\end{bmatrix}\end{array}$	$\begin{array}{cc} & \begin{matrix}1&2&3&4&5&6&7\end{matrix} \\ \begin{matrix}1\\2\\3\\4\\5\\6\\7\end{matrix} & \begin{bmatrix}0&1&0&1&1&2&2\\1&0&1&0&0&0&0\\0&1&0&0&0&0&0\\1&0&0&0&0&0&0\\1&0&0&0&0&0&0\\2&0&0&0&0&0&0\\2&0&0&0&0&0&0\end{bmatrix}\end{array}$	

For the assemblies fall under this category, the stability predicate consideration increases the time without any change in the outcomes. If the stability matrix behaviour is closely related to liaison matrix, ignoring stability checking will save lots of computational time for assemblies with huge number of parts. However this is applicable to only linear assembly systems.

In most of the assemblies, stability matrix behaviour is not similar to liaison matrix and hence without testing for stability predicate the quality of assembly sequence cannot be assured. Few assemblies belong to such category are shown in Figure 4.3 and 4.4.

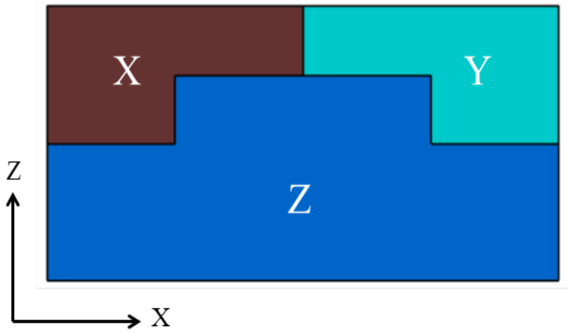


Figure 4.3:3-Part block assembly

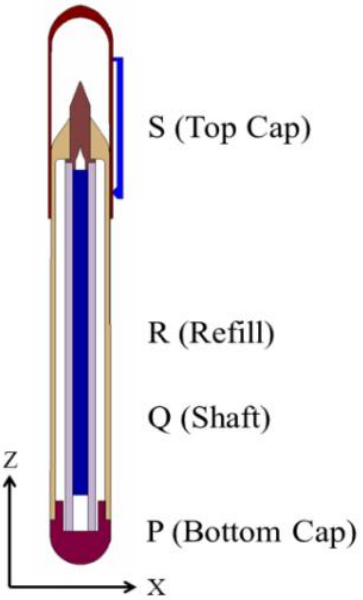


Figure 4.4: 4-Part pen assembly

Set of all possible assembly sequences for the products shown in Figure 4.3 and 4.4 are tested by considering stability predicate and the resulted outcomes and computational time are listed in Table 4.3.

Table 4.3: Outcomes for products shown in Figure 4.3 and 4.4 considering stability predicate

Assembly	**Number of parts**	**Number of assembly sequences**	**List of assembly sequences**	**Computational time (sec.)**
Figure 4.3	3	2	(Z-X-Y) (Z-Y-X)	0.92
Figure 4.4	4	1	(P-R-Q-S)	1.25

Possible assembly sequences are obtained by ignoring the stability predicate and the resulted outcomes are listed in Table 4.4. Comparing the results listed in table 4.3, it is observed there exist not-stable assembly sequences for these assembly configurations.

Table 4.4: Outcomes for products shown in Figure 4.3 and 4.4 by ignoring stability predicate

Assembly	**Number of parts**	**Number of assembly sequences**	**Number of unstable assembly sequences**	**Computational time (sec.)**
Figure 4.3	3	6	4	0.85
Figure 4.4	4	8	7	1.03

While analysing the products shown in Figure 4.3 and 4.4, it is observed that ignoring stability predicate increases the solution space quite largely and thereby reduces the computational time. However probability of resulting an unstable assembly sequence is more in this scenario. It is also observed that considering stability yields to precise solution space and there by enhances the computational time, but it assures the desirable results. These type of assembly configurations can be identified based on dissimilarities between their liaison matrix and combined stability matrix. Table 4.5 characterises the liaison matrix and combined stability matrix for the products shown in Figure 4.3 and 4.4.

Table 4.5: Observations based on liaison and stability matrices

Assembly	**Number of Parts**	**Liaison matrix**	**Stability matrix**	**Remarks**
Figure 4.3	3	$\begin{array}{c\|ccc} & X & Y & Z \\ \hline X & 0 & 1 & 1 \\ Y & 1 & 0 & 1 \\ Z & 1 & 1 & 0 \end{array}$	$\begin{array}{c\|ccc} & X & Y & Z \\ \hline X & 0 & 0 & 0 \\ Y & 0 & 0 & 0 \\ Z & 1 & 1 & 0 \end{array}$	Not for every lm(i,j)>0; stb(i,j)>0
Figure 4.4	4	$\begin{array}{c\|cccc} & P & Q & R & S \\ \hline P & 0 & 1 & 1 & 0 \\ Q & 1 & 0 & 1 & 1 \\ R & 1 & 1 & 0 & 0 \\ S & 0 & 1 & 0 & 0 \end{array}$	$\begin{array}{c\|cccc} & P & Q & R & S \\ \hline P & 0 & 2 & 0 & 0 \\ Q & 2 & 0 & 0 & 2 \\ R & 1 & 0 & 0 & 0 \\ S & 0 & 2 & 0 & 0 \end{array}$	

A comparative study on stability predicate consideration towards quality of resulted assembly sequences and computational time is clearly illustrated in Figure. 4.5 and Figure 4.6.

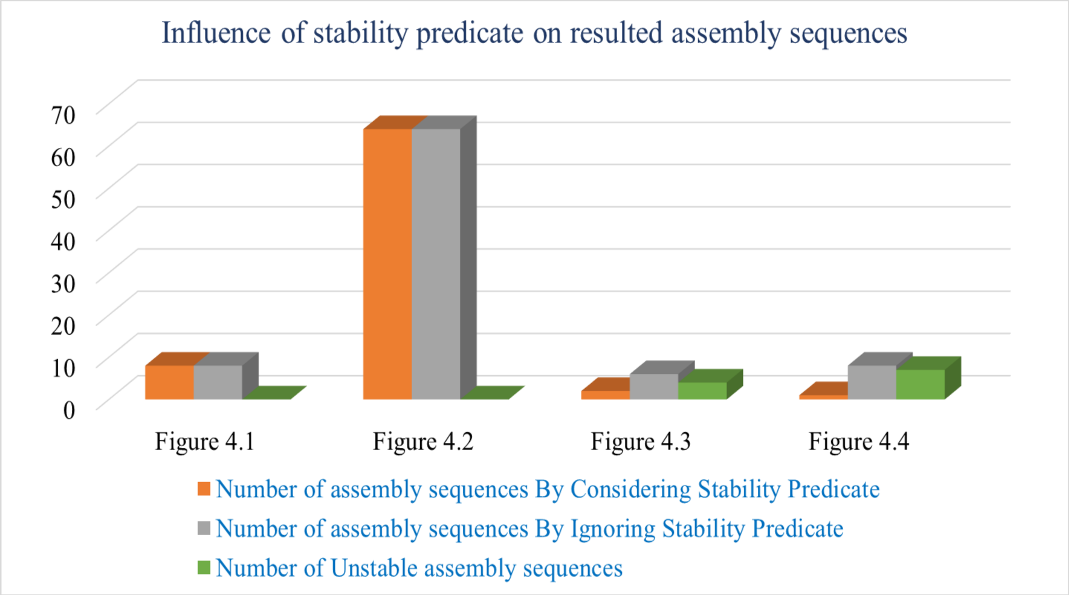


Figure 4.5: Influence of stability predicate on resulted assembly sequence

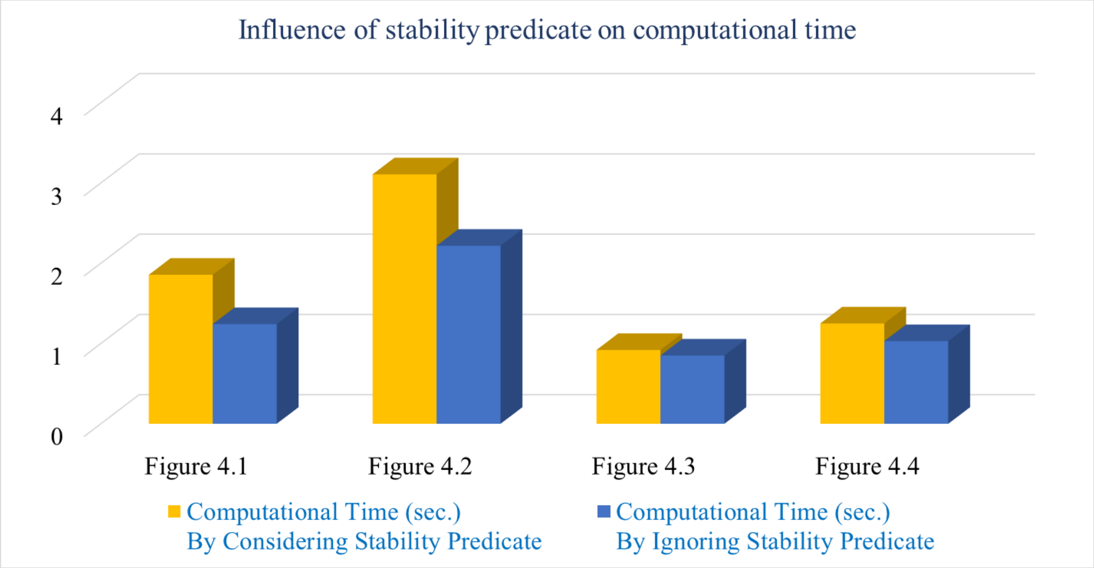


Figure 4.6: Influence of stability predicate on computational time

4.3 Influence of Ignoring Mechanical Feasibility Predicate

Several connectors are used to join primary parts together, in most of the situations these connectors come along with the primary parts. This assumption is valid in certain conditions, which motivated researchers to ignore the connectors. To study the effect of ignoring mechanical feasibility predicate on quality of resulted assembly sequence and computational time, transmission assembly (De Fazio, 1987) shown in Figure 3.12 is considered. Total number of possible sequences and the computational time to retrieve the sequences are listed in table 4.6.

Table 4.6: Performance Characteristics for transmission assembly without considering mechanical feasibility predicate

Performance evaluation parameter	**Without considering mechanical feasibility**
Search Space	11!
No. of resulted sequences (Solution Space)	2320
Computational time (sec)	306.26

In order to test the quality of outcomes, the presence of connectors are treated as primary parts (when the attachments are considered as primary parts, the mechanical feasibility is not essential to consider). The problem is redefined with 16 parts by considering additional six primary parts, each additional part is a set of connectors joining a pair of primary parts. Total number of possible sequences and the computational time to retrieve the sequences are listed in table 4.7.

Table 4.7: Performance Characteristics for 16 Part assembly

Performance evaluation parameter	**Without Considering Mechanical Feasibility**
Search space	16!
Number of resulted sequences (Solution Space)	1808
Computational time (sec)	12605.08

It is observed from the outcomes, ignoring attachments reduces the search space and thus computational time is low to achieve set of all possible sequences. Considering attachments as primary parts increase the number parts and leads to exponential rise in search space. Although the computational time is amplified extremely, the resulted sequences are practically possible. Ignoring mechanical feasibility predicate leads to many mechanically infeasible assembly sequences.

Though the concept of treating each attachment as primary part yield correct results; each attachment must be checked along all six directions for a collision free path to assemble. As the number of attachments increases, the computational time is raised to test their geometrical feasibility.

A comparative study on mechanical feasibility predicate consideration on quality of resulted assembly sequences and computational time is clearly illustrated in Figure. 4.7 and 4.8.

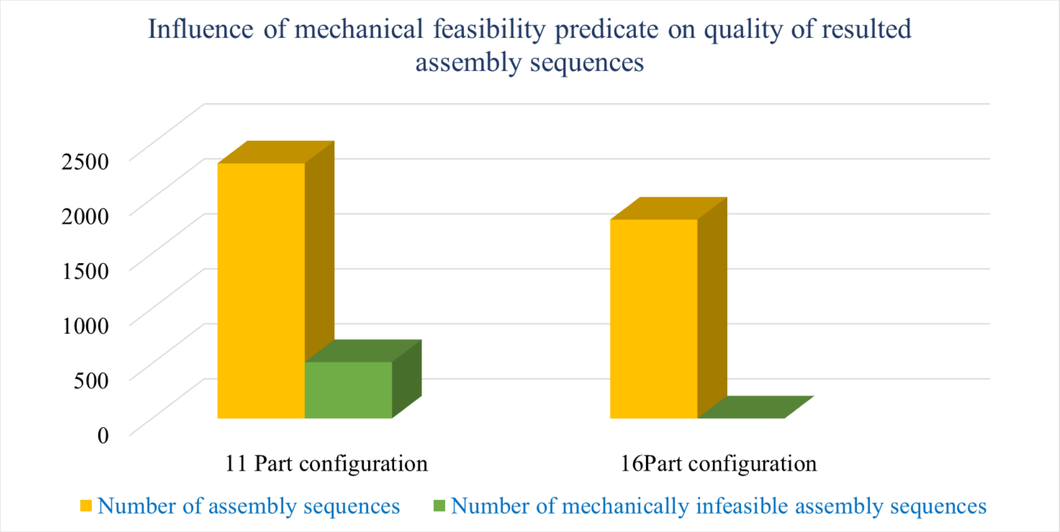


Figure 4.7: Influence of mechanical feasibility predicate consideration on quality of resulted assembly sequences

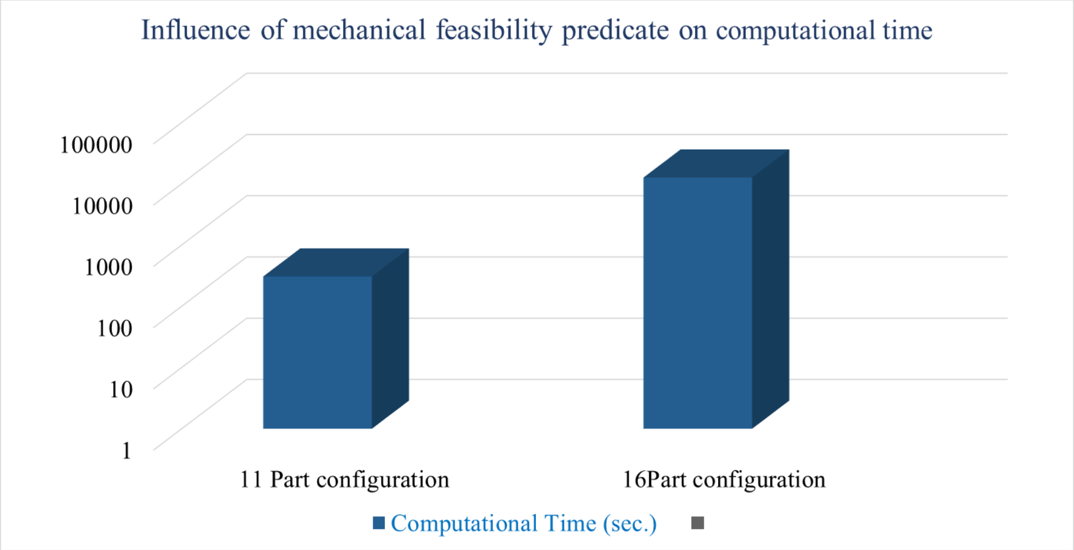


Figure 4.8: Influence of mechanical feasibility predicate consideration on computational time

4.4 Influence of Considering Mechanical Feasibility Predicate

Most of the attachments yield one degree of freedom either to perform assembly/disassembly operation. Concept of mechanical feasibility represents hindrance of a primary part during joining process of attachments. A three dimensional square matrix is used for this purpose as described in chapter 3. Set of all possible assembly sequences are generated by considering mechanical feasibility predicate for 11-part assembly and the resulted outcomes are listed in table 4.8.

Table 4.8: Performance characteristics for 11-part assembly by considering mechanical feasibility predicate

Performance evaluation parameter	**Considering Mechanical Feasibility**
Search space	11!
No. of resulted sequences (Solution space)	1808
Computational time (sec)	406.35

It is observed from the resulted out comes, consideration of mechanical feasibility predicate reduces the computational time to extract set of all possible assembly sequences without compromising the quality. However complexity of problem slightly increased due to extraction of mechanical feasibility matrix for a given product. Comparative analyses between three categories are listed in table 4.9.

Table 4.9: Mechanical feasibility predicate based performance evaluation

Performance evaluation parameter	**Not considering mechanical feasibility**	**Mechanical feasibility Not applicable**	**Considering mechanical feasibility**
Number of Parts	11	16	11(+5 attachments)
Search Space	11!	16!	11!
No. of resulted sequences (Solution Space)	2320	1808	1808
No. of infeasible assembly sequences.	512	-	-
Computational time for optimal assembly sequence planning(sec)	306.26	12605.08	406.35

While performing optimal assembly sequence generation, the search space is dependent only on the number of parts, yet the solution space can be altered by consideration of assembly predicates. Although ignoring the mechanical feasibility predicate results in enhanced solution space and offer low computational time to achieve the objective during optimal assembly sequence generation, but this is not desirable due to probability of resulting mechanically infeasible solution. The problem formulation has become complicated by considering the mechanical feasibility predicate and leads to high computational time due to reduced solution space, on the other hand desired mechanical feasible solution is guaranteed.

4.5 Summary

This chapter delivers the implications of assembly predicate consideration on computational performance, quality of outcomes, solution space and search space of the assembly sequence generation problem. Detailed study on different possible assembly configurations has been discussed.

Chapter 5

PART CONCATENATION METHOD

5.1 Overview

Part concatenation method is described to generate set of all possible assembly sequences considering necessary assembly attributes and assembly predicate testing methods. These set of all feasible sequences are further used to find best assembly sequence considering user defined weights.

5.2 Part Concatenation Method

Part concatenation method use assembly knowledge database, which consist of all assembly attributes such as liaison matrix, interference matrices, combined stability matrix, mechanical feasibility matrix and bounding box coordinates along with assembly gripper data. Initially the method checks for the similarities between liaison matrix and stability matrix to decide necessity stability matrix consideration. If there exist physical attachments, the mechanical feasibility matrix will be verified for non-zero elements to consider the mechanical feasibility predicate.

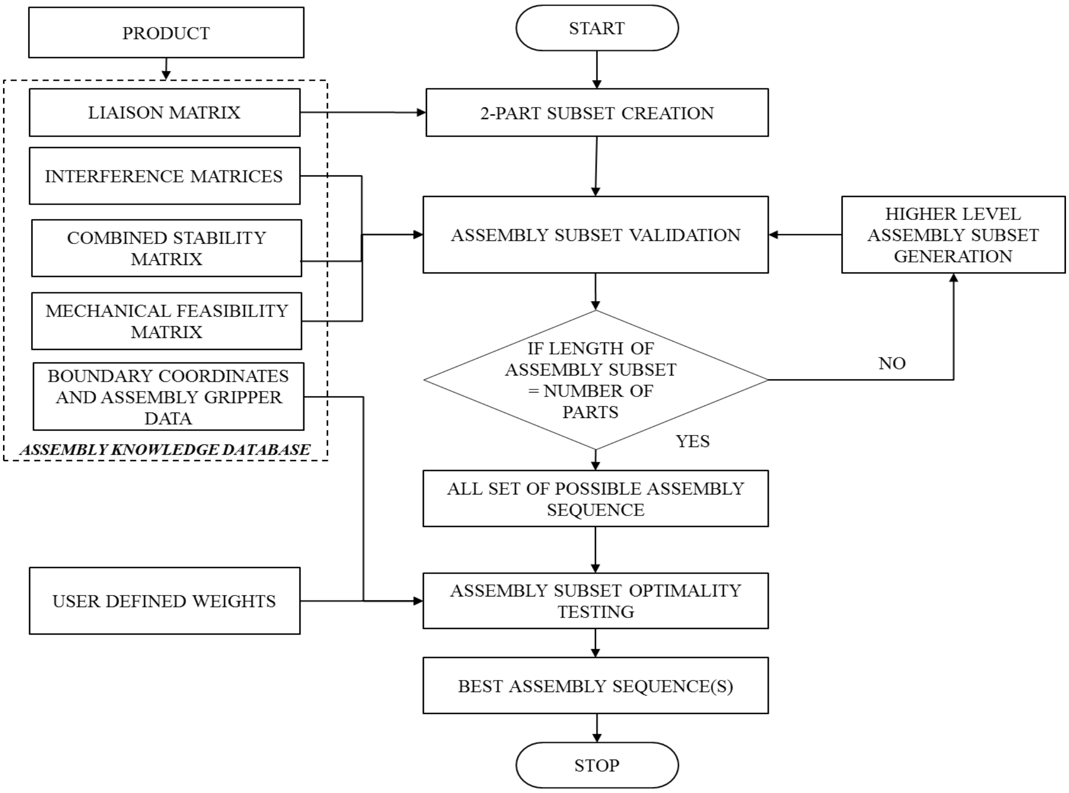


Figure 5.1: Part Concatenation method

The bounding box coordinates are further used to determine the optimal feasible direction and to compute the fitness values for a feasible assembly sequence by considering the user defined weights. Part concatenation method work flow is briefly indicated in Figure 5.1.

The method initially considers liaison matrix from the assembly knowledge database and generate two part assembly subsets. These two-part subsets are further tested for their validity using different assembly predicates and the filtered sets are only passed to generate next level assembly subsets. The process is iterated till the assembly subset length equals to total number of parts in the product. All the resulted assembly sequences are further tested for optimality. More detailed description on each phase is discussed below.

5.3 Two-Part Assembly Subset Generation

Liaison matrix is considered at this phase to create two part assembly subsets. Surface contacts between have been identified from the non-zero liaison matrix element to pair them in two possible means. Though these two possibilities yield same assembly subset, the fitness value may differ from one to another. An algorithm to generate two part assembly subset is given here.

```
Step 1:  For i=1 to total number of parts
Step 2:        For j=1 to total number of parts
Step 3:              If lm(i,j)=1
                           Generate an assembly subset with (part-i, part-j)
                     End if
               Next
  Next
```

Due to the symmetric nature of the liaison matrix, the algorithm is improved to reduce the time consumption for alternate possibility of assembly subset generation in the inverted sequence as stated below.

Pseudo code to generate liaison based subsets is shown below:

```
Step 1:  For i=1 to total number of parts
Step 2:        For j=i to total number of parts
Step 3:              If lm(i,j)=1
                        Generate an assembly subset with (part-i, part-j)
                        Generate alternate possible assembly subset (part-j, part-i)
                     End if
               Next
         Next
```

Products with dissimilar assembly configurations are shown in Figures 5.2 – 5.5 are considered to illustrate the part concatenation method.

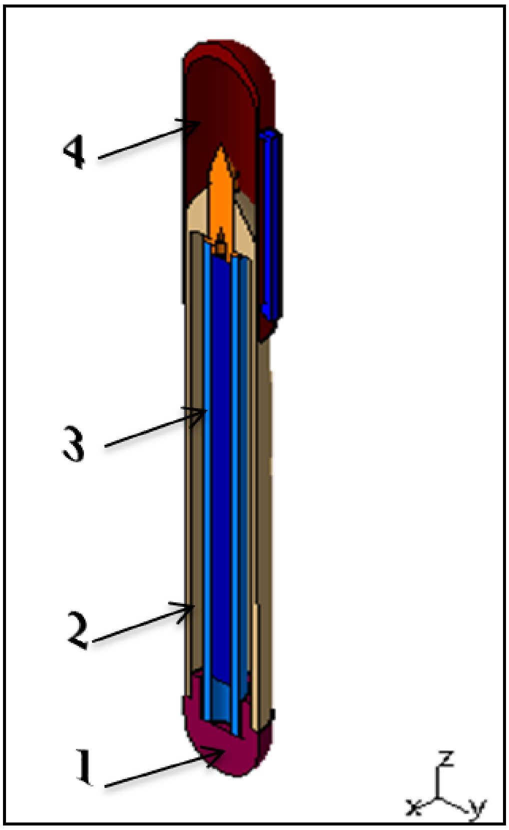


Figure 5.2: Cut-section of 4-part pen assembly

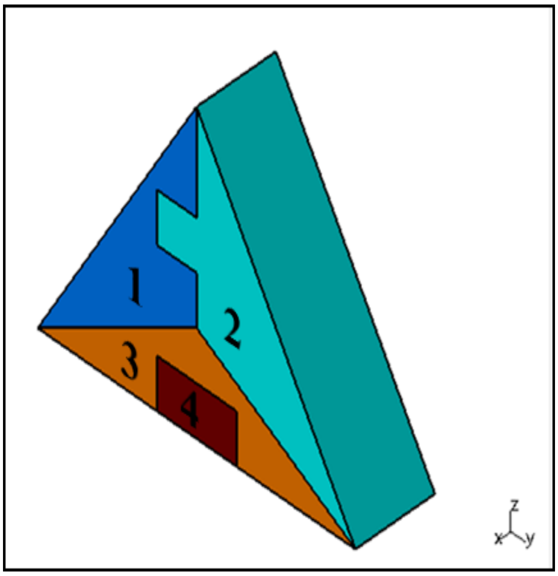


Figure 5.3: 4-Part Block assembly

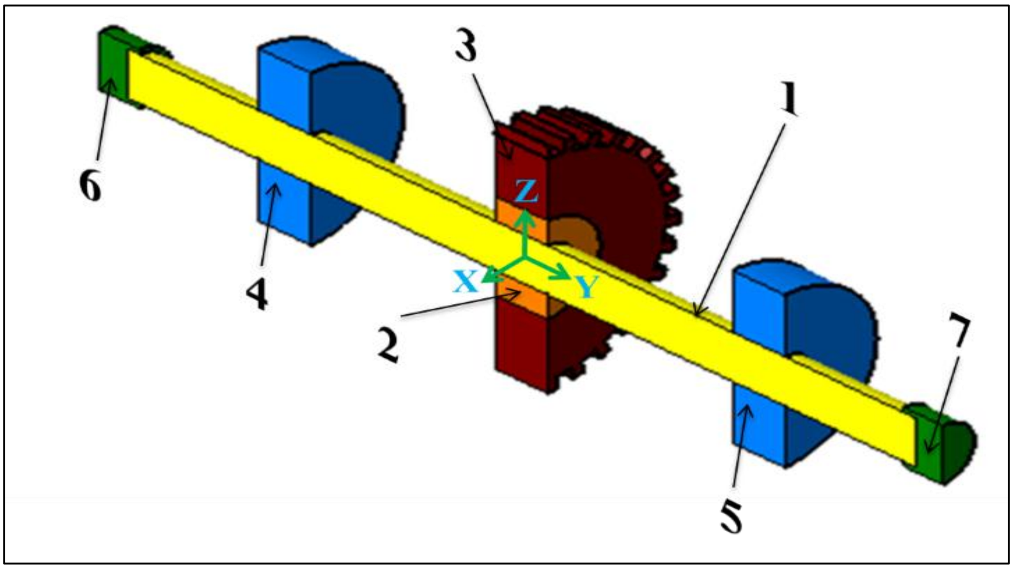


Figure 5.4: Cut-section of 7-part gear assembly

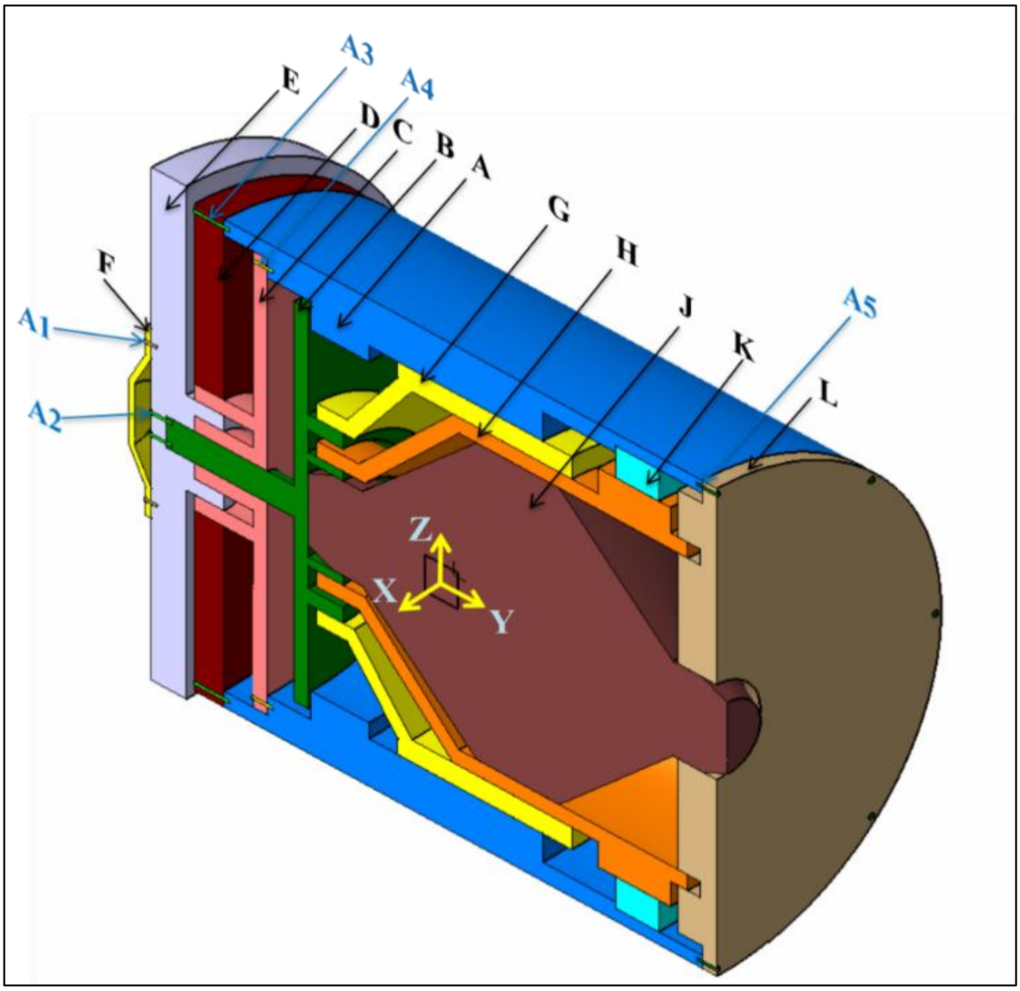


Figure 5.5: Cut-section of 11-part transmission assembly

The two part subset generation mechanism is implemented on the products shown in Figure 5.2 to 5.5, and the resulted two part assembly subsets are listed in Table 5.1.

Table 5.1: List of 2-part assembly subsets

S. No.	**Assembly**	**No. of Parts**	**Liaison matrix**	**Number of two part assembly subsets**	
1	Figure 5.2	4	$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$	1-2 1-3 2-3 2-4	2-1 3-1 3-2 4-2
2	Figure 5.3	4	$\begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$	1-2 1-3 2-3 3-4	2-1 3-1 3-2 4-3
3	Figure 5.4	7	$\begin{array}{cc} & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$	1-2 1-4 1-5 1-6 1-7 2-3	2-1 4-1 5-1 6-1 7-1 3-2
4	Figure 5.5	11	$\begin{array}{cc} & \begin{array}{ccccccccccc} A & B & C & D & E & F & G & H & J & K & L \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ J \\ K \\ L \end{array} & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$	A-C A-D A-G A-K A-L B-C B-E B-G B-H B-J C-D C-E E-F G-H H-J H-K H-L J-L	C-A D-A G-A K-A L-A C-B E-B G-B H-B J-B D-C E-C F-E H-G J-H K-H L-H L-J

5.4 Assembly Subset Validation

Each assembly subset is tested for all the necessary assembly predicates stated in assembly predicate testing (chapter: 3). Once the assembly subset is qualified for all assembly predicates, these subsets are used for higher level assembly subset generation. Assembly subset validation procedure for two-part subsets of products shown in Figure 5.2 and Figure 5.3 is listed in Table 5.2.

Table 5.2: Assembly predicate testing of 2 parts subset

Assembly	**S. No.**	**No. of Parts**	**Is appending part geometrically feasible to create assembly subset?**	**Is appending part stable with respect to existing part?**
Figure 5.2	1	1-2	Yes	Yes
	2	1-3	Yes	Yes
	3	2-3	Yes	No
	4	2-4	Yes	Yes
	5	2-1	Yes	Yes
	6	3-1	Yes	No
	7	3-2	Yes	No
	8	4-2	Yes	Yes
Figure 5.3	1	1-2	Yes	Yes
	2	1-3	Yes	No
	3	2-3	Yes	No
	4	3-4	Yes	No
	5	2-1	Yes	Yes
	6	3-1	Yes	Yes
	7	3-2	Yes	Yes
	8	4-3	Yes	Yes

Only qualified subsets are further send for next level iteration. For the products shown in Figure 5.4 and 5.5 all the two part assembly subsets qualify necessary assembly predicates and thus all sets are passed for next phase of three part assembly subset generation. However, assembly subset validation process goes in hand with assembly subset generation to reduce the computational time and to eliminate non possible assembly subsets.

5.5 Higher Level Assembly Subset Generation

Higher level assembly subset generation process use part concatenation mechanism, considering higher order rule. The algorithm finds parts, which are in contact to any existent part in the assembly subset and appends to it with the aim of creating next level assembly subset. An algorithm to generate higher level assembly subset generation is described below.

```
Step 1:  For i=1 to total number of assembly subsets
Step 2:      For j=1 to total number of parts
Step 3:          Append part-j to assembly subset-i;
                 If liaison predicate fails go to next part else go to feasibility test
           If geometrical feasibility test fails go to next assembly subset else test for stability
          If stability predicate test fails go to next part else test for mechanical feasibility
         If mechanical feasibility test fails go to next part else generate next level subset
             Next
         Next
```

During the assembly subset generation, assembly predicate testing is done to qualify each assembly subset. The mechanism to generate three part assembly subsets for the products shown Figure 5.2, Figure. 5.3 and Figure 5.4 are listed in Table 5.3, Table 5.4 and Table 5.5 successively.

Table 5.3: Three part subset generation for assembly shown in Figure 5.2.

S. No.	**Assembly**	**Assembly Subset**	**Non Existent Part**	**Assembly Predicate Testing**				**Resulted Assembly subset**
				Liaison	Geometrical Feasibility	Stability	Mechanical Feasibility	
1	Figure 5.2	1-2	3	Yes	No			
2			4	Yes	Yes	Yes	N/A	1-2-4
3		1-3	2	Yes	Yes	Yes	N/A	1-3-2
4			4	No				
5		2-4	1	Yes	Yes	Yes	N/A	2-4-1
6			3	Yes	Yes	No		
7		2-1	3	Yes	No			2-1-3
8			4	Yes	Yes	Yes	N/A	2-1-4
9		4-2	1	Yes	Yes	Yes	N/A	4-2-1
10			3	Yes	Yes	No		

Table 5.4: Three part subset generation for assembly shown in Figure 5.3

S. No.	**Assembly**	**Assembly Subset**	**Non Existent Part**	**Assembly Predicate Testing**				**Resulted Assembly subset**
				Liaison	Geometrical Feasibility	Stability	Mechanical Feasibility	
1	Figure 5.3	1-2	3	Yes	Yes	No		
2			4	No				
3		3-1	2	Yes	Yes	Yes	N/A	3-1-2
4			4	Yes	Yes	No		
5		3-2	1	Yes	Yes	Yes	N/A	3-2-1
6			4	Yes	Yes	No		
7		2-1	3	Yes	Yes	No		
8			4	No				
9		4-3	1	Yes	Yes	Yes	N/A	4-2-1
10			2	Yes	Yes	Yes	N/A	4-3-2

Table 5.5: Three part subset generation for assembly shown in Figure 5.4.

S. No.	**Assembly**	**Assembly Subset**		**Non Existent Part**	**Assembly Predicate Testing**				**Resulted Assembly subset**
					Liaison	Geometrical Feasibility	Stability	Mechanical Feasibility	
1	Figure 5.4	1-2	2-1	3	Yes	Yes	Yes	N/A	1-2-3 / 2-1-3
2				4	Yes	Yes	Yes	N/A	1-2-4 / 2-1-4
3				5	Yes	Yes	Yes	N/A	1-2-5 / 2-1-5
4				6	Yes	Yes	Yes	N/A	1-2-6 / 2-1-6
5				7	Yes	Yes	Yes	N/A	1-2-7 / 2-1-7
6		1-4	4-1	2	Yes	Yes	Yes	N/A	1-4-2 / 4-1-2
7				3	No				
8				5	Yes	Yes	Yes	N/A	1-4-5 / 4-1-5
9				6	Yes	Yes	Yes	N/A	1-4-6 / 4-1-6
10				7	Yes	Yes	Yes	N/A	1-4-7 / 4-1-7
11		1-5	5-1	2	Yes	Yes	Yes	N/A	1-5-2 / 5-1-2
12				3	No				
13				4	Yes	Yes	Yes	N/A	1-5-4 / 5-1-4
14				6	Yes	Yes	Yes	N/A	1-5-6 / 5-1-6
15				7	Yes	Yes	Yes	N/A	1-5-7 / 5-1-7
16		1-6	6-1	2	Yes	Yes	Yes	N/A	1-6-2 / 6-1-2
17				3	No				
18				4	Yes	Yes	Yes	N/A	1-6-4 / 6-1-4
19				5	Yes	Yes	Yes	N/A	1-6-5 / 6-1-5
20				7	Yes	Yes	Yes	N/A	1-6-7 / 6-1-7
21		1-7	7-1	2	Yes	Yes	Yes	N/A	1-7-2 / 7-1-2
22				3	No				
23				4	Yes	Yes	Yes	N/A	1-7-4 / 7-1-4
24				5	Yes	Yes	Yes	N/A	1-7-5 / 7-1-5
25				6	Yes	Yes	Yes	N/A	1-7-6 / 7-1-6
26		2-3	3-2	1	Yes	Yes	Yes	N/A	2-3-1 / 3-2-1
27				4	No				
28				5	No				
29				6	No				
30				7	No				

Three part assembly subsets generated for the product shown in the figure 5.5 are listed in table 5.6.

Table 5.6: List of three part subsets for transmission assembly

S. No.	Two part assembly subset		Generated three part assembly subset			S. No.	Two part assembly subset		Generated three part assembly subset	
1	A-C	C-A	A-C-D	C-A-D		35	B-G	G-B	B-G-H	G-B-H
2			A-C-E	C-A-E		36			B-G-J	G-B-J
3			A-C-G	C-A-G		37	B-H	H-B	B-H-C	H-B-C
4			A-C-K	C-A-K		38			B-H-E	H-B-E
5			A-C-L	C-A-L		39			B-H-J	H-B-J
6	A-D	D-A	A-D-G	D-A-G		40			B-H-K	H-B-K
7			A-D-K	D-A-K		41			B-H-L	H-B-L
8			A-D-L	D-A-L		42	B-J	J-B	B-J-C	J-B-C
9	A-G	G-A	A-G-B	G-A-B		43			B-J-E	J-B-E
10			A-G-C	G-A-C		44			B-J-L	J-B-L
11			A-G-D	G-A-D		45	C-D	D-C	C-D-B	D-C-B
12			A-G-H	G-A-H		46			C-D-E	D-C-E
13			A-G-K	G-A-K		47	C-E	E-C	C-E-B	E-C-B
14			A-G-L	G-A-L		48			C-E-F	E-C-F
15	A-K	K-A	A-K-C	K-A-C		49	E-F	F-E	E-F-C	F-E-C
16			A-K-D	K-A-D		50	G-H	H-G	G-H-A	H-G-A
17			A-K-G	K-A-G		51			G-H-B	H-G-B
18			A-K-H	K-A-H		52			G-H-J	H-G-J
19			A-K-L	A-K-L		53			G-H-K	H-G-K
20	A-L	L-A	A-L-C	L-A-C		54			G-H-L	H-G-L
21			A-L-D	L-A-D		55	H-J	J-H	H-J-B	J-H-B
22			A-L-J	L-A-J		56			H-J-G	J-H-G
23	B-C	C-B	B-C-A	C-B-A		57			H-J-K	J-H-K
24			B-C-D	C-B-D		58			H-J-L	J-H-L
25			B-C-E	C-B-E		59	H-K	K-H	H-K-B	K-H-B
26			B-C-G	C-B-G		60			H-K-G	K-H-G
27			B-C-H	C-B-H		61			H-K-J	K-H-J
28			B-C-J	C-B-J		62			H-K-L	K-H-L
29	B-E	E-B	B-E-F	E-B-F		63	H-L	L-H	H-L-A	L-H-A
30			B-E-G	E-B-G		64			H-L-B	L-H-B
31			B-E-H	E-B-H		65			H-L-G	L-H-G
32			B-E-J	E-B-J		66	J-L	L-J	J-L-A	L-J-A
33	B-G	G-B	B-G-C	G-B-C		67			J-L-B	L-J-B
34			B-G-E	G-B-E		68			J-L-H	L-J-H

For the assembly shown in Figure 5.5, 68 qualified 3-part assembly subsets have been generated as listed in Table 5.6. However due to presence of physical connectors, mechanical feasibility is tested for this assembly configuration. Mechanically unqualified assembly subsets are listed in Table 5.7.

Table 5.7: Mechanically infeasible assembly subsets for transmission assembly

S. No.	**Assembly**	**Assembly Subset**	**Non Existent Part**	**Assembly Predicate Testing**			
				Liaison	**Geometrical Feasibility**	**Stability**	**Mechanical Feasibility**
1	Figure 5.5	C-D	A	Yes	Yes	Yes	No
2		C-E	A	Yes	Yes	Yes	No
3		D-C	A	Yes	Yes	Yes	No
4		C-E	A	Yes	Yes	Yes	No
5		E-F	B	Yes	Yes	Yes	No
6		F-E	B	Yes	Yes	Yes	No

5.6 Set of all Feasible Assembly Sequences

The higher level assembly subset generation process iterates till the number of parts in the generated assembly subsets equals to the total number of parts in the product. A system with (i7-3770 CPU @ 3.40 GHz, 2.00 GB RAM) configuration settings is used to test the performance. The computational time to execute the code in order to result the all valid sequences has been presented in Table.5.8.

Table 5.8: Performance of part concatenation method and valid sequences.

Assembly (As indicated in figure)	**Number of parts**	**Computational time (sec)**	**Total number of valid assembly sequences**
Figure 5.2	4	1.25	1
Figure 5.3	4	1.02	2
Figure 5.4	7	3	64
Figure 5.5	11	406	1808

The computational time includes storing the data in an excel file. Time to store the results is more for the products shown in Figure 5.4 and Figure 5.5, due to more number of possible assembly sequences. Further selecting optimal assembly sequence for different criteria is discussed here.

5.7 Optimal Assembly Sequence Selection

From huge set of multiple feasible assembly sequences, each sequence offer its own benefits in terms of minimum assembly directional changes, minimum gripper changes, lowest assembly energy or combination of these. Finding out such optimal sequence need a defined fitness function, the following sections describes fitness function formulation for different objectives.

5.7.1 Minimal Assembly Reorientations

Change in part joining directional changes consumes more assembly efforts and time to do the assembly operation, Hence an assembly sequence with minimum number of assembly directional changes would be the choice of interest. In certain instances, it is possible to assemble a component in multiple feasible directions, the optimal direction is based on the direction of preceding component or succeeding component in the assembly sequence. An algorithm to detect the direction matrix is given below.

```
For each feasible assembly sequence
For each part in the sequence ( j=last part to part in second position)
    Get all possible feasible assembly directions for each part (j)
    If there exist a feasible direction of precedence part, assign it
            Else test with respect to succeeding part
End For
End for
```

An algorithm to detect the number of directional changes for a feasible sequences based on its direction matrix is stated below.

```
For each feasible assembly sequence
Change=0
For each part in the sequence ( j=1 to second part from last)
    If direction of part-j differs from its succeeding part
    Capture the change
End For
End for
```

The method is implemented on the products shown in Figure 5.4 and resulted directional matrix and number of directional changes are listed in Table 5.9.

Table 5.9: Direction Matrix and Number of reorientations for set of all feasible sequences.

S. No.	**Assembly Sequence**	**Direction**	**NDC**	**S. No.**	**Assembly Sequence**	**Direction**	**NDC**
1	1-2-3-4-5-6-7	y+,y+,y+,y+,y-,y+,y-	3	33	2-1-4-3-5-6-7	y+,y+,y+,y-,y-,y+,y-	3
2	1-2-3-4-5-7-6	y+,y+,y+,y+,y-,y-,y+	2	34	2-1-4-3-5-7-6	y+,y+,y+,y-,y-,y-,y+	2
3	1-2-3-4-6-5-7	y+,y+,y+,y+,y+,y-,y-	1	35	2-1-4-3-6-5-7	y+,y+,y+,y-,y+,y-,y-	3
4	1-2-3-5-4-6-7	y-,y-,y-,y-,y+,y+,y-	2	36	2-1-4-6-3-5-7	y+,y+,y+,y+,y-,y-,y-	1
5	1-2-3-5-4-7-6	y-,y-,y-,y-,y+,y-,y+	3	37	2-1-5-3-4-6-7	y-,y-,y-,y+,y+,y+,y-	2
6	1-2-3-5-7-4-6	y-,y-,y-,y-,y-,y+,y+	1	38	2-1-5-3-4-7-6	y-,y-,y-,y+,y+,y-,y+	3
7	1-2-4-3-5-6-7	y+,y+,y+,y-,y-,y+,y-	3	39	2-1-5-3-7-4-6	y-,y-,y-,y+,y-,y+,y+	3
8	1-2-4-3-5-7-6	y+,y+,y+,y-,y-,y-,y+	2	40	2-1-5-7-3-4-6	y-,y-,y-,y-,y+,y+,y+	1
9	1-2-4-3-6-5-7	y+,y+,y+,y-,y+,y-,y-	3	41	2-3-1-4-5-6-7	y+,y+,y+,y+,y-,y+,y-	3
10	1-2-4-6-3-5-7	y+,y+,y+,y+,y-,y-,y-	1	42	2-3-1-4-5-7-6	y+,y+,y+,y+,y-,y-,y+	2
11	1-2-5-3-4-6-7	y-,y-,y-,y+,y+,y+,y-	2	43	2-3-1-4-6-5-7	y+,y+,y+,y+,y+,y-,y-	1
12	1-2-5-3-4-7-6	y-,y-,y-,y+,y+,y-,y+	3	44	2-3-1-5-4-6-7	y-,y-,y-,y-,y+,y+,y-	2
13	1-2-5-3-7-4-6	y-,y-,y-,y+,y-,y+,y+	3	45	2-3-1-5-4-7-6	y-,y-,y-,y-,y+,y-,y+	3
14	1-2-5-7-3-4-6	y-,y-,y-,y-,y+,y+,y+	1	46	2-3-1-5-7-4-6	y-,y-,y-,y-,y-,y+,y+	1
15	1-4-2-3-5-6-7	y-,y-,y-,y-,y-,y+,y-	2	47	3-2-1-4-5-6-7	y+,y+,y+,y+,y-,y+,y-	3
16	1-4-2-3-5-7-6	y-,y-,y-,y-,y-,y-,y+	1	48	3-2-1-4-5-7-6	y+,y+,y+,y+,y-,y-,y+	2
17	1-4-2-3-6-5-7	y-,y-,y-,y-,y+,y-,y-	2	49	3-2-1-4-6-5-7	y+,y+,y+,y+,y+,y-,y-	1
18	1-4-2-6-3-5-7	y-,y-,y-,y+,y-,y-,y-	2	50	3-2-1-5-4-6-7	y-,y-,y-,y-,y+,y+,y-	2
19	1-4-6-2-3-5-7	y+,y+,y+,y-,y-,y-,y-	1	51	3-2-1-5-4-7-6	y-,y-,y-,y-,y+,y-,y+	3
20	1-5-2-3-4-6-7	y+,y+,y+,y+,y+,y+,y-	1	52	3-2-1-5-7-4-6	y-,y-,y-,y-,y-,y+,y+	1
21	1-5-2-3-4-7-6	y+,y+,y+,y+,y+,y-,y+	2	53	4-1-2-3-5-6-7	y-,y-,y-,y-,y-,y+,y-	2
22	1-5-2-3-7-4-6	y+,y+,y+,y+,y-,y+,y+	2	54	4-1-2-3-5-7-6	y-,y-,y-,y-,y-,y-,y+	1
23	1-5-2-7-3-4-6	y+,y+,y+,y-,y+,y+,y+	2	55	4-1-2-3-6-5-7	y-,y-,y-,y-,y+,y-,y-	2
24	1-5-7-2-3-4-6	y-,y-,y-,y+,y+,y+,y+	1	56	4-1-2-6-3-5-7	y-,y-,y-,y+,y-,y-,y-	2
25	1-6-4-2-3-5-7	y+,y+,y-,y-,y-,y-,y-	1	57	4-1-6-2-3-5-7	y+,y+,y+,y-,y-,y-,y-	1
26	1-7-5-2-3-4-6	y-,y-,y+,y+,y+,y+,y+	1	58	5-1-2-3-4-6-7	y+,y+,y+,y+,y+,y+,y-	1
27	2-1-3-4-5-6-7	y+,y+,y+,y+,y-,y+,y-	3	59	5-1-2-3-4-7-6	y+,y+,y+,y+,y+,y-,y+	2
28	2-1-3-4-5-7-6	y+,y+,y+,y+,y-,y-,y+	2	60	5-1-2-3-7-4-6	y+,y+,y+,y+,y-,y+,y+	2
29	2-1-3-4-6-5-7	y+,y+,y+,y+,y+,y-,y-	1	61	5-1-2-7-3-4-6	y+,y+,y+,y-,y+,y+,y+	2
30	2-1-3-5-4-6-7	y-,y-,y-,y-,y+,y+,y-	2	62	5-1-7-2-3-4-6	y-,y-,y-,y+,y+,y+,y+	1
31	2-1-3-5-4-7-6	y-,y-,y-,y-,y+,y-,y+	3	63	6-1-4-2-3-5-7	y-,y-,y-,y-,y-,y-,y-	0
32	2-1-3-5-7-4-6	y-,y-,y-,y-,y-,y+,y+	1	64	7-1-5-2-3-4-6	y+,y+,y+,y+,y+,y+,y+	0

Due to huge number of feasible assembly sequences for transmission assembly shown in Figure 5.5, assembly sequences with zero assembly direction changes and one assembly direction changes are listed in Table 5.10.

Table 5.10: Direction Matrix and Number of reorientations.

S. No	Assembly Sequence	Direction Matrix	NDC
1	L-J-H-G-K-A-B-C-D-E-F	y+,y+,y+,y+,y+,y+,y+,y+,y+,y+,y+	0
2	L-J-H-K-G-A-B-C-D-E-F	y+,y+,y+,y+,y+,y+,y+,y+,y+,y+,y+	0
3	A-G-H-J-K-L-B-C-D-E-F	y-,y-,y-,y-,y-,y-,y+,y+,y+,y+,y+	1
4	A-G-H-K-J-L-B-C-D-E-F	y-,y-,y-,y-,y-,y-,y+,y+,y+,y+,y+	1
5	A-G-K-H-J-L-B-C-D-E-F	y-,y-,y-,y-,y-,y-,y+,y+,y+,y+,y+	1
6	A-K-G-H-J-L-B-C-D-E-F	y-,y-,y-,y-,y-,y-,y+,y+,y+,y+,y+	1
7	C-B-A-G-H-J-K-L-D-E-F	y-,y-,y-,y-,y-,y-,y-,y-,y+,y+,y+	1
8	C-B-A-G-H-K-J-L-D-E-F	y-,y-,y-,y-,y-,y-,y-,y-,y+,y+,y+	1
9	C-B-A-G-K-H-J-L-D-E-F	y-,y-,y-,y-,y-,y-,y-,y-,y+,y+,y+	1
10	C-B-A-K-G-H-J-L-D-E-F	y-,y-,y-,y-,y-,y-,y-,y-,y+,y+,y+	1
11	G-A-B-C-D-E-F-H-J-K-L	y+,y+,y+,y+,y+,y+,y+,y-,y-,y-,y-	1
12	G-A-B-C-D-E-F-H-K-J-L	y+,y+,y+,y+,y+,y+,y+,y-,y-,y-,y-	1
13	G-A-B-C-D-E-F-K-H-J-L	y+,y+,y+,y+,y+,y+,y+,y-,y-,y-,y-	1
14	G-H-J-K-L-A-B-C-D-E-F	y-,y-,y-,y-,y-,y+,y+,y+,y+,y+,y+	1
15	G-H-J-L-K-A-B-C-D-E-F	y-,y-,y-,y-,y+,y+,y+,y+,y+,y+,y+	1
16	G-H-K-J-L-A-B-C-D-E-F	y-,y-,y-,y-,y-,y+,y+,y+,y+,y+,y+	1
17	H-G-A-B-C-D-E-F-J-K-L	y+,y+,y+,y+,y+,y+,y+,y+,y-,y-,y-	1
18	H-G-A-B-C-D-E-F-K-J-L	y+,y+,y+,y+,y+,y+,y+,y+,y-,y-,y-	1
19	H-G-K-A-B-C-D-E-F-J-L	y+,y+,y+,y+,y+,y+,y+,y+,y+,y-,y-	1
20	H-J-K-L-G-A-B-C-D-E-F	y-,y-,y-,y-,y+,y+,y+,y+,y+,y+,y+	1
21	H-J-L-G-K-A-B-C-D-E-F	y-,y-,y-,y+,y+,y+,y+,y+,y+,y+,y+	1
22	H-J-L-K-G-A-B-C-D-E-F	y-,y-,y-,y+,y+,y+,y+,y+,y+,y+,y+	1
23	H-K-G-A-B-C-D-E-F-J-L	y+,y+,y+,y+,y+,y+,y+,y+,y+,y-,y-	1
24	H-K-J-L-G-A-B-C-D-E-F	y-,y-,y-,y-,y+,y+,y+,y+,y+,y+,y+	1
25	J-H-G-A-B-C-D-E-F-K-L	y+,y+,y+,y+,y+,y+,y+,y+,y+,y-,y-	1
26	J-H-G-K-A-B-C-D-E-F-L	y+,y+,y+,y+,y+,y+,y+,y+,y+,y+,y-	1
27	J-H-K-G-A-B-C-D-E-F-L	y+,y+,y+,y+,y+,y+,y+,y+,y+,y+,y-	1
28	J-L-H-G-K-A-B-C-D-E-F	y-,y-,y+,y+,y+,y+,y+,y+,y+,y+,y+	1
29	J-L-H-K-G-A-B-C-D-E-F	y-,y-,y+,y+,y+,y+,y+,y+,y+,y+,y+	1
30	K-H-G-A-B-C-D-E-F-J-L	y+,y+,y+,y+,y+,y+,y+,y+,y+,y-,y-	1
31	K-H-J-L-G-A-B-C-D-E-F	y-,y-,y-,y-,y+,y+,y+,y+,y+,y+,y+	1

There exists many alternate possible assembly sequences for the objective of assembly directional changes. Out of 1808 feasible assembly sequences, number of assembly sequences with different number of assembly orientations are presented in Table 5.11 and figure 5.6.

Table 5.11: Number of sequences list with possible reorientations

No. of Directional Changes	**No. of assembly sequences**
0	2
1	29
2	117
3	273
4	446
5	443
6	316
7	148
8	31
9	3

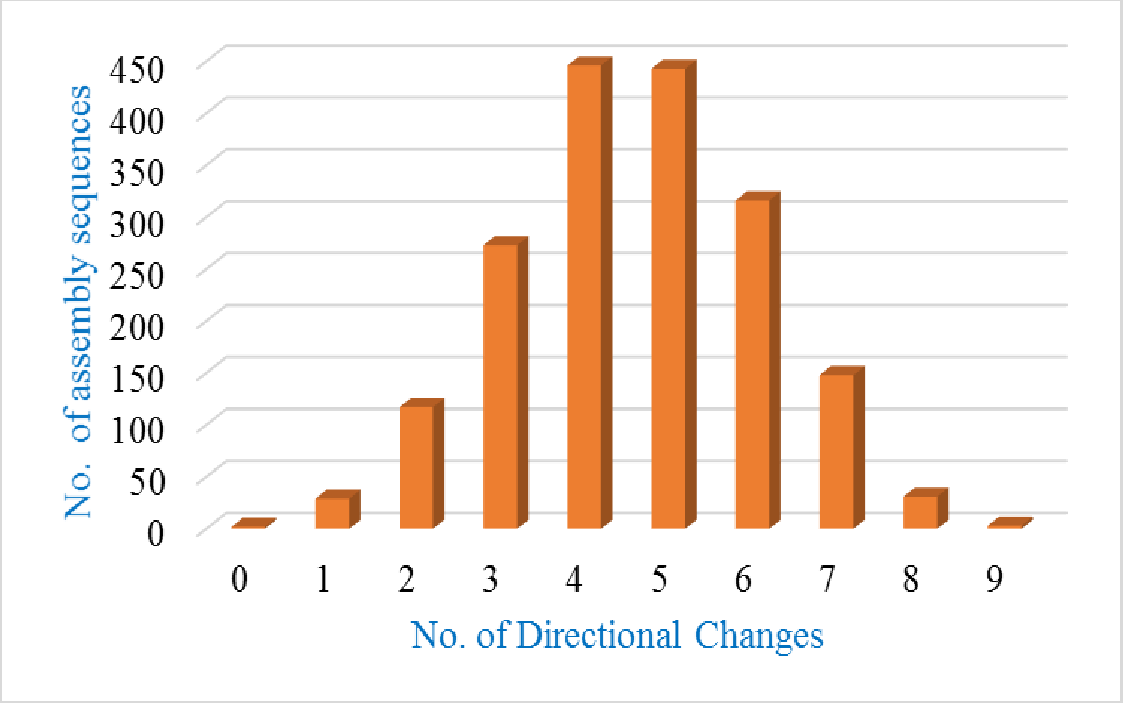


Figure 5.6: Number of sequences with possible reorientations

5.7.2 Minimal Gripper Changes

Due to change in part geometries, different assembly tools are required for holding and assembling operations. Change in grippers/tools increases the assembly time, hence an assembly sequence with minimum number of gripper changes offer minimized assembly time and efforts for gripper changes.

In certain instances, multiple components with similar geometrical configurations can be assembled with a same gripper, hence gripper selection is also dependent on the preceding component and succeeding component in the assembly sequence. An algorithm to detect the gripper matrix is given below.

```
For each feasible assembly sequence
    For each part in the sequence ( j=last part to part in second position)
        Get list of possible grippers  used for each part (j)
        If there exist a gripper similar to that of used for precedence part, assign it
                Else test with respect to succeeding part
    End For
End for
```

An algorithm to detect the number of gripper changes for a feasible sequences based on gripper matrix is stated below.

```
For each feasible assembly sequence
Change=0
For each part in the sequence ( j=1 to second part from last)
    If gripper number of part-j differs from its succeeding part
    Capture the change
End For
End for
```

The method is implemented on the product shown in Figure 5.4 and resulted gripper matrix and number of gripper changes are listed in Table 5.12.

Table 5.12: List of sequences with minimum gripper changes

S. No.	**Assembly Sequence**	**Gripper Matrix**	**NGC**	**S. No.**	**Assembly Sequence**	**Gripper Matrix**	**NGC**
1	1-2-3-4-5-6-7	1-2-3-4-4-6-6	4	33	1-2-3-4-6-5-7	1-2-3-4-6-5-7	6
2	1-2-3-4-5-7-6	1-2-3-4-4-7-7	4	34	1-2-3-5-7-4-6	1-2-3-5-7-4-6	6
3	1-2-3-5-4-6-7	1-2-3-5-5-6-6	4	35	1-2-4-3-6-5-7	1-2-4-3-6-5-7	6
4	1-2-3-5-4-7-6	1-2-3-5-5-7-7	4	36	1-2-4-6-3-5-7	1-2-4-6-3-5-7	6
5	2-1-3-4-5-6-7	2-1-3-4-4-6-6	4	37	1-2-5-3-7-4-6	1-2-5-3-7-4-6	6
6	2-1-3-4-5-7-6	2-1-3-4-4-7-7	4	38	1-2-5-7-3-4-6	1-2-5-7-3-4-6	6
7	2-1-3-5-4-6-7	2-1-3-5-5-6-6	4	39	1-4-2-3-6-5-7	1-4-2-3-6-5-7	6
8	2-1-3-5-4-7-6	2-1-3-5-5-7-7	4	40	1-4-2-6-3-5-7	1-4-2-6-3-5-7	6
9	2-3-1-4-5-6-7	2-3-1-4-4-6-6	4	41	1-4-6-2-3-5-7	1-4-6-2-3-5-7	6
10	2-3-1-4-5-7-6	2-3-1-4-4-7-7	4	42	1-5-2-3-7-4-6	1-5-2-3-7-4-6	6
11	2-3-1-5-4-6-7	2-3-1-5-5-6-6	4	43	1-5-2-7-3-4-6	1-5-2-7-3-4-6	6
12	2-3-1-5-4-7-6	2-3-1-5-5-7-7	4	44	1-5-7-2-3-4-6	1-5-7-2-3-4-6	6
13	3-2-1-4-5-6-7	3-2-1-4-4-6-6	4	45	1-6-4-2-3-5-7	1-6-4-2-3-5-7	6
14	3-2-1-4-5-7-6	3-2-1-4-4-7-7	4	46	1-7-5-2-3-4-6	1-7-5-2-3-4-6	6
15	3-2-1-5-4-6-7	3-2-1-5-5-6-6	4	47	2-1-3-4-6-5-7	2-1-3-4-6-5-7	6
16	3-2-1-5-4-7-6	3-2-1-5-5-7-7	4	48	2-1-3-5-7-4-6	2-1-3-5-7-4-6	6
17	1-2-4-3-5-6-7	1-2-4-3-5-6-6	5	49	2-1-4-3-6-5-7	2-1-4-3-6-5-7	6
18	1-2-4-3-5-7-6	1-2-4-3-5-7-7	5	50	2-1-4-6-3-5-7	2-1-4-6-3-5-7	6
19	1-2-5-3-4-6-7	1-2-5-3-4-6-6	5	51	2-1-5-3-7-4-6	2-1-5-3-7-4-6	6
20	1-2-5-3-4-7-6	1-2-5-3-4-7-7	5	52	2-1-5-7-3-4-6	2-1-5-7-3-4-6	6
21	1-4-2-3-5-6-7	1-4-2-3-5-6-6	5	53	2-3-1-4-6-5-7	2-3-1-4-6-5-7	6
22	1-4-2-3-5-7-6	1-4-2-3-5-7-7	5	54	2-3-1-5-7-4-6	2-3-1-5-7-4-6	6
23	1-5-2-3-4-6-7	1-5-2-3-4-6-6	5	55	3-2-1-4-6-5-7	3-2-1-4-6-5-7	6
24	1-5-2-3-4-7-6	1-5-2-3-4-7-7	5	56	3-2-1-5-7-4-6	3-2-1-5-7-4-6	6
25	2-1-4-3-5-6-7	2-1-4-3-5-6-6	5	57	4-1-2-3-6-5-7	4-1-2-3-6-5-7	6
26	2-1-4-3-5-7-6	2-1-4-3-5-7-7	5	58	4-1-2-6-3-5-7	4-1-2-6-3-5-7	6
27	2-1-5-3-4-6-7	2-1-5-3-4-6-6	5	59	4-1-6-2-3-5-7	4-1-6-2-3-5-7	6
28	2-1-5-3-4-7-6	2-1-5-3-4-7-7	5	60	5-1-2-3-7-4-6	5-1-2-3-7-4-6	6
29	4-1-2-3-5-6-7	4-1-2-3-5-6-6	5	61	5-1-2-7-3-4-6	5-1-2-7-3-4-6	6
30	4-1-2-3-5-7-6	4-1-2-3-5-7-7	5	62	5-1-7-2-3-4-6	5-1-7-2-3-4-6	6
31	5-1-2-3-4-6-7	5-1-2-3-4-6-6	5	63	6-1-4-2-3-5-7	6-1-4-2-3-5-7	6
32	5-1-2-3-4-7-6	5-1-2-3-4-7-7	5	64	7-1-5-2-3-4-6	7-1-5-2-3-4-6	6

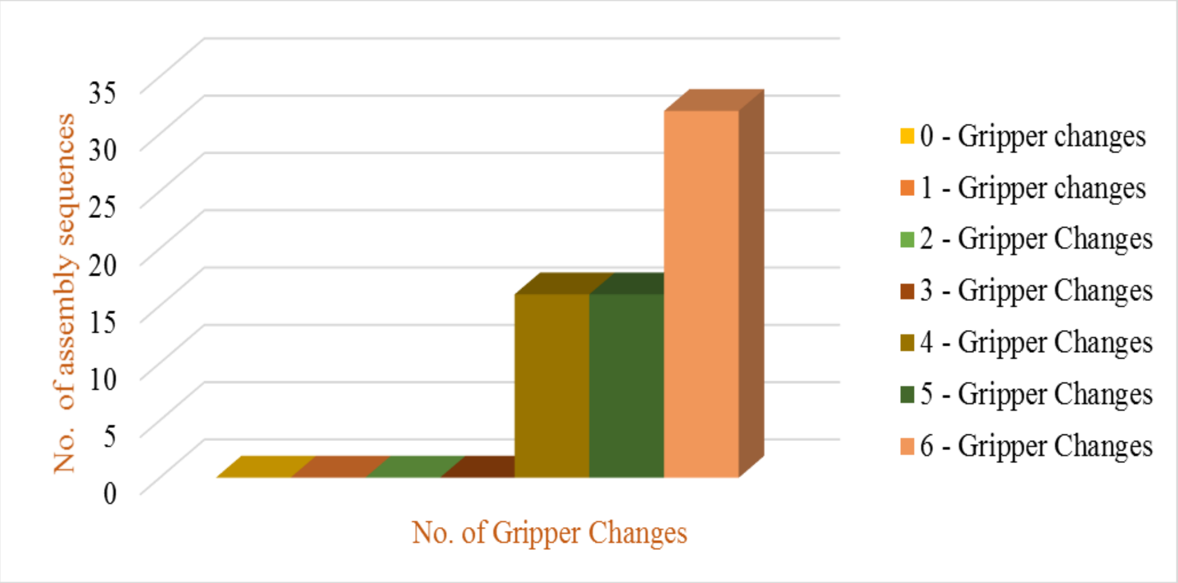


Figure 5.7: Number of sequences with possible gripper changes

A graph is drawn between minimum numbers of gripper changes against number of assembly sequences. It is observed from figure 5.7, there exist 6 number of sequences with minimum gripper changes. The method is implemented on the product shown in Figure 5.5 and all the sequences offer same gripper changes due to the reason, no gripper can be used for two different parts.

5.7.3 Minimum Assembly Energy

The distance travelled by each part to perform assembly raises the assembly efforts and assembly time. Hence an optimal assembly sequence should offer low assembly energy to economize the assembly process. Energy associated with the part movements is the summation of energy to assemble each part from second to last position in the assembly sequence along a feasible direction. Energy to assemble a component to an existed sub-assembly can be calculated using equation (1) using the upper and lower bounding corner points of the subassembly and the part to be assembled.

$$\delta = \sum_{i=2}^{n} m_i d_i \tag{1}$$

m_i – weight of the part-i; d_i – distance travelled by part-i along feasible direction. The units for assembly energy is “Joule”.

An Algorithm to compute the energy associated with each part movement is mentioned below.

```
For each feasible assembly sequence
Energy=0
  For each part in the sequence ( í=2 to second part from last)
    E=Compute energy to assembly $j^{th}$ part using bounding box coordinates and volume
    Energy=Energy + E
  End For
```

The method is implemented on the product shown in Figure 5.4 and resulted energy values are arranged descending order in Table 5.13.

Table 5.13: Set of all feasible assembly sequences with assembly energy

S. No.	**Assembly Sequence**	**Energy (j)**	**S. No.**	**Assembly Sequence**	**Energy (J)**
1	4-1-6-2-3-5-7	3.832679428	33	5-1-2-7-3-4-6	2.565932721
2	5-1-7-2-3-4-6	3.832679428	34	5-1-2-3-7-4-6	2.565932721
3	7-1-5-2-3-4-6	3.526166234	35	5-1-2-3-4-7-6	2.565932721
4	6-1-4-2-3-5-7	3.526166233	36	5-1-2-3-4-6-7	2.565932721
5	1-7-5-2-3-4-6	3.462285654	37	1-2-5-7-3-4-6	2.462222465
6	1-5-2-7-3-4-6	3.462285654	38	1-2-5-3-7-4-6	2.462222465
7	1-5-2-3-7-4-6	3.462285654	39	1-2-5-3-4-7-6	2.462222465
8	1-5-2-3-4-7-6	3.462285654	40	1-2-5-3-4-6-7	2.462222465
9	1-5-2-3-4-6-7	3.462285654	41	1-2-4-6-3-5-7	2.462222465
10	1-4-2-6-3-5-7	3.462285654	42	1-2-4-3-5-7-6	2.462222465
11	1-4-2-3-5-7-6	3.462285654	43	1-2-4-3-5-6-7	2.462222465
12	1-4-2-3-5-6-7	3.462285654	44	1-2-3-5-7-4-6	2.462222465
13	1-6-4-2-3-5-7	3.462285654	45	1-2-3-5-4-7-6	2.462222465
14	1-4-2-3-6-5-7	3.462285654	46	1-2-3-5-4-6-7	2.462222465
15	2-1-5-7-3-4-6	3.351167522	47	1-2-3-4-6-5-7	2.462222465
16	2-1-5-3-7-4-6	3.351167522	48	1-2-3-4-5-7-6	2.462222465
17	2-1-5-3-4-7-6	3.351167522	49	1-2-3-4-5-6-7	2.462222465
18	2-1-5-3-4-6-7	3.351167522	50	1-5-7-2-3-4-6	2.462222465
19	2-1-4-6-3-5-7	3.351167522	51	1-4-6-2-3-5-7	2.462222465
20	2-1-4-3-5-7-6	3.351167522	52	1-2-4-3-6-5-7	2.462222465
21	2-1-4-3-5-6-7	3.351167522	53	2-3-1-5-7-4-6	2.096106854
22	2-1-3-5-7-4-6	3.351167522	54	2-3-1-5-4-7-6	2.096106854
23	2-1-3-5-4-7-6	3.351167522	55	2-3-1-5-4-6-7	2.096106854
24	2-1-3-5-4-6-7	3.351167522	56	2-3-1-4-6-5-7	2.096106854
25	2-1-3-4-6-5-7	3.351167522	57	2-3-1-4-5-7-6	2.096106854
26	2-1-3-4-5-7-6	3.351167522	58	2-3-1-4-5-6-7	2.096106854
27	2-1-3-4-5-6-7	3.351167522	59	3-2-1-5-7-4-6	1.946443975
28	2-1-4-3-6-5-7	3.351167522	60	3-2-1-5-4-7-6	1.946443975
29	4-1-2-6-3-5-7	2.565932721	61	3-2-1-5-4-6-7	1.946443975
30	4-1-2-3-5-7-6	2.565932721	62	3-2-1-4-6-5-7	1.946443975
31	4-1-2-3-5-6-7	2.565932721	63	3-2-1-4-5-7-6	1.946443975
32	4-1-2-3-6-5-7	2.565932721	64	3-2-1-4-5-6-7	1.946443975

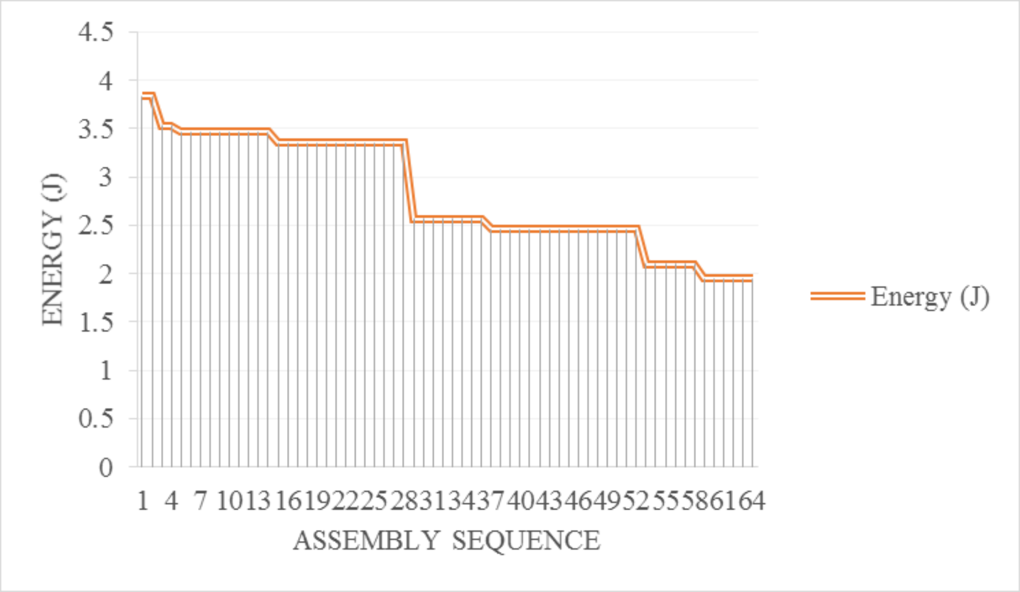


Figure 5.8: Energy representation for gear assembly

Energy values for set of all feasible assembly sequences are represented in Figure 5.8 for gear assembly shown in Figure 5.4, in which there exist six assembly sequences with minimum assembly energy.

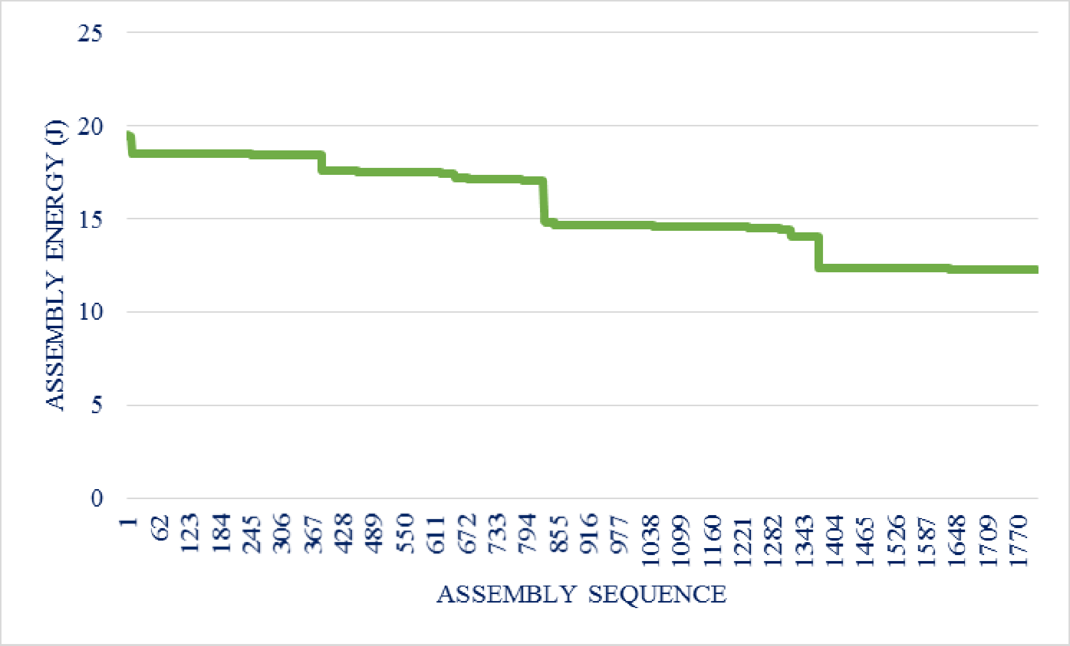


Figure 5.9: Assembly energy representation for transmission assembly

Energy values for set of all feasible assembly sequences are represented in Figure 5.9 for transmission assembly shown in Figure 5.5, in which there exist 174-assembly sequences with minimum assembly energy out of 1808-set of all feasible sequences.

5.7.4 Combined Objective

Assembly sequences with minimum assembly directional changes may not offer minimum assembly gripper changes and minimum assembly energy. Hence a combined objective function is required to define an optimality criteria with multiple benefits. However these weights must be defined based on the assembly facilities and requirements. A combined objective function with three weight factors is given below equation (2).

$$E = \sum_{i=1}^{3} w_i \delta_i \qquad (2)$$

δ_1 - *Energy associated with the part movements to create the assembly*
δ_2 - *Number of robot direction changes associated with assembly operations*
δ_3 - *Number of tool/gripper changes associated with assembly operations*
w_i - *weights associated with each segment*

The method is applied on products shown in figure 5.4 and Figure 5.5, and results obtained for different combinations of weight factors are listed in below table 5.14.

Table 5.14: List of optimal assembly sequences for combined objective function

Product	**Weights**	**Assembly sequence**	**Assembly Direction matrix**	**Gripper matrix**
Figure 5.4	$w_1=0.6;$ $w_2=0.2;$ $w_3=0.2;$	3-2-1-4-5-7-6	y+,y+,y+,y+,y-,y-,y+	3-2-1-4-4-7-7
		3-2-1-5-4-6-7	y-,y-,y-,y-,y+,y+,y-	3-2-1-5-5-6-6
	$w_1=0.25;$ $w_2=0.5;$ $w_3=0.25;$	6-1-4-2-3-5-7	y-,y-,y-,y-,y-,y-,y-	6-1-4-2-3-5-7
		7-1-5-2-3-4-6	y+,y+,y+,y+,y+,y+,y+	6-1-4-2-3-5-7
Figure 5.5	$w_1=0.25;$ $w_2=0.5;$ $w_3=0.25;$	a-g-j-h-i-k-b-c-d-e-f	y-,y-,y-,y-,y-,y-,y+,y,+y+,y+,y+	1-7-10-8-9-11-2-3-4-5-6
		a-j-g-h-i-k-b-c-d-e-f	y-,y-,y-,y-,y-,y-,y+,y,+y+,y+,y+	1-10-7-8-9-11-2-3-4-5-6
	$w_1=0.15;$ $w_2=0.7;$ $w_3=0.15;$	k-i-h-g-j-a-b-c-d-e-f	y+,y,+y+,y+,y+, y+,y,+y+,y+,y+,y+	11-9-8-7-10-1-2-3-4-5-6
		k-i-h-j-g-a-b-c-d-e-f	y+,y,+y+,y+,y+, y+,y,+y+,y+,y+,y+	11-9-8-10-7-1-2-3-4-5-6

5.8 Summary

This chapter illustrates the part concatenation method to build assembly subsets and feasible assembly sequences. Selection of Optimal feasible assembly sequences from set of all feasible sequences for single and/or combined objective function considering assembly direction change, assembly gripper change and assembly energy.

Chapter 6

DIRECT GENERATION OF OPTIMAL ASSEMBLY SEQUENCES

6.1 Overview

Generating set of all feasible sequences is highly time consuming due to presence of similar assembly subsets at each intermediate level of part concatenation method with altered sequence. Further selecting optimal sequences considering user defined weights rise the computational time. Eliminating such similar intermediate assembly subsets for a defined objective function save lots of computational time for products with huge number of parts. In this chapter, direct generation of optimal assembly sequences is discussed.

6.2 Assembly Indexing Method

In order to identify similar assembly subsets, assembly indexing method is developed. In this method, each component of the product is assigned with a numeric value based on the part number. Subsets with same components results equal assembly index value. The assembly index computation for different assembly subsets is presented in table.6.1.

Table 6.1: Assembly index computation

Assembly subset	**Assembly index computation**	**Assembly index**
1-2-3	$(10^1+10^2+10^3)/10$	111
2-3-1	$(10^2+10^3+10^1)/10$	111
1-4-7	$(10^1+10^4+10^7)/10$	1001001
1-6-7-4	$(10^1+10^6+10^7+10^4)/10$	1101001
1-6-4-7	$(10^1+10^6+10^4+10^7)/10$	1101001
1-4-7-6	$(10^1+10^4+10^7+10^6)/10$	1101001
1-4-6-7	$(10^1+10^4+10^6+10^7)/10$	1101001

There can be subsets with same assembly index with altered patterns. Four different sequences with same assembly index for four part assembly subsets are represented in Table.6.1, Furthermore in these sequences, few sequences may consume high fitness value and also there can be number of subsets with same fitness value for the assembling operations. Subsets with high fitness values must be deleted and no more helpful in finding optimal sequences. The subsets with equal energy level are redundant in nature and hence only one subset must be considered for the computation purpose. Pseudo code to calculate the assembly indices for the assembly subsets are presented below.

Pseudo code for assembly index computation

```
for i=1 to count
  ai(n,i)=0,  n is number of parts in the subset
  for j=1 to n
     ai(n,i)=ai(n,i)+(10 power asub(n,j)/10)
  end for
end for
```

Modified part concatenation method for direct optimal assembly sequence generation for user defined weights is presented in Figure 6.1

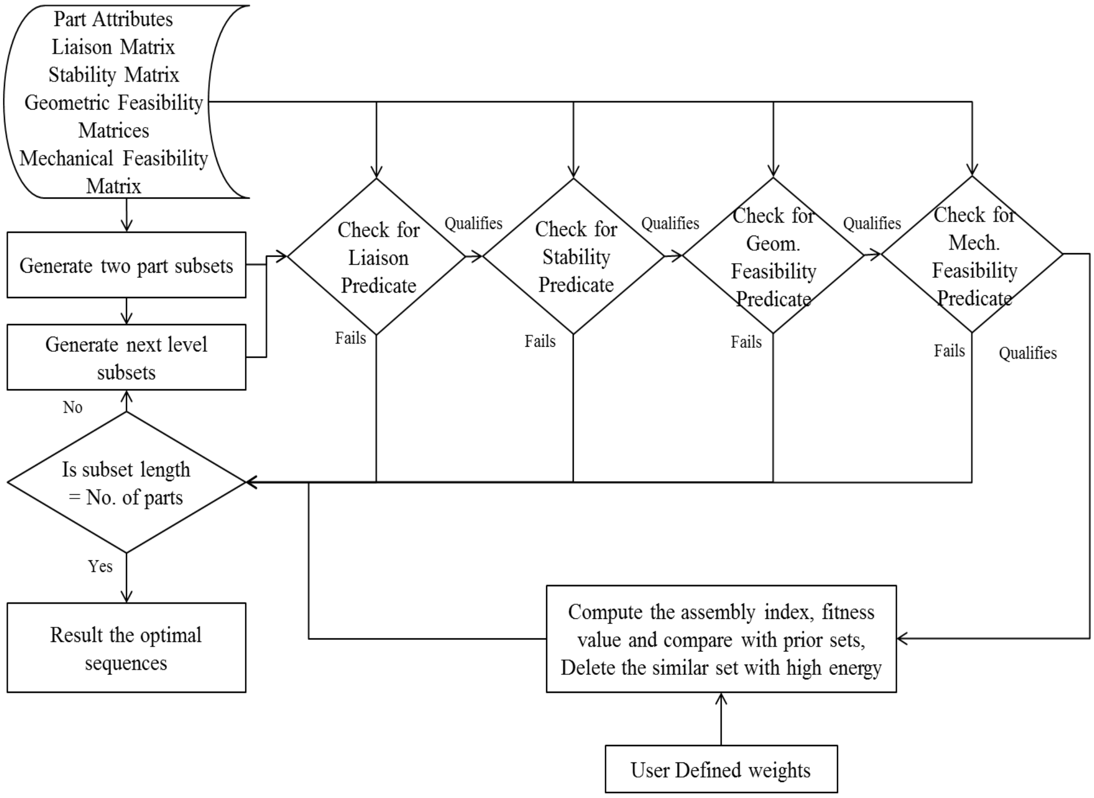


Figure 6.1: Direct method to obtain optimal assembly sequences

Considering minimum number of assembly directions as the objective function, the proposed assembly indexing is implemented through part concatenation method and the resulted assembly subsets, their assembly direction matrix and alternate possibilities are listed at all intermediate levels for 7-part gear assembly. Table 6.2 lists two part assembly subsets with unique assembly index value, their assembly direction matrix along with alternate possibilities.

Table 6.2: Two part assembly subsets with unique assembly index value

S. No.	Assembly Sequences	Assembly Index	Assembly direction	Alternate direction	Alternate sequence	Assembly direction	Alternate direction
1	1-2	11	4-4	3-3	2-1	4-4	3-3
2	2-3	110	4-4	3-3	3-2	4-4	3-3
3	1-4	1001	4-4	3-3	4-1	4-4	3-3
4	1-5	10001	4-4	3-3	5-1	4-4	3-3
5	1-6	100001	4-4		6-1	3-3	
6	1-7	1000001	3-3		7-1	4-4	

The filtered two part assembly subsets are used to generate three part assembly subsets through part concatenation method. Similar sets with high fitness values are eliminated by using assembly indexing method. Table 6.3 lists three part assembly subsets with unique assembly index value, their assembly direction matrix along with alternate possibilities.

Table 6.3: Three part assembly subsets with unique assembly index value

S. No.	Assembly Sequences	Assembly Index	Assembly direction	Alternate direction	Alternate sequence	Assembly direction	Alternate direction
1	1-2-3	111	4-4-4	3-3-3	2-3-1	4-4-4	3-3-3
2	1-2-4	1011	4-4-4		1-4-2	3-3-3	
3	1-2-5	10011	3-3-3		1-5-2	4-4-4	
4	1-4-5	11001	3-3-3		1-5-4	4-4-4	
5	1-2-6	100011	4-4-4		6-1-2	3-3-3	
6	1-4-6	101001	4-4-4		6-1-4	3-3-3	
7	1-5-6	110001	4-4-4		6-1-5	3-3-3	
8	1-2-7	1000011	3-3-3		7-1-2	4-4-4	
9	1-4-7	1001001	3-3-3		7-1-4	4-4-4	
10	1-5-7	1010001	3-3-3		7-1-5	4-4-4	
11	6-1-7	1100001	3-3-3		7-1-6	4-4-4	

As the number of parts in the subset increase, the possible combinations also raises exponentially however due to different fitness values similar subsets are deleted. Table 6.4 lists four part assembly subsets with unique assembly index value, their assembly direction matrix along with alternate possibilities.

Table 6.4: Four part assembly subsets with unique assembly index value

S. No.	**Assembly Sequences**	**Assembly Index**	**Assembly direction**	**Alternate sequence**	**Assembly direction**
1	1-2-3-4	1111	4-4-4-4	1-4-2-3	3-3-3-3
2	1-2-3-5	10111	3-3-3-3	1-5-2-3	4-4-4-4
3	1-4-2-5	11011	3-3-3-3	1-5-2-4	4-4-4-4
4	1-2-3-6	100111	4-4-4-4		
5	1-2-4-6	101011	4-4-4-4	6-1-4-2	3-3-3-3
6	1-5-2-6	110011	4-4-4-4		
7	6-1-4-5	111001	3-3-3-3		
8	1-2-3-7	1000111	3-3-3-3		
9	1-4-2-7	1001011	3-3-3-3		
10	1-2-5-7	1010011	3-3-3-3		
11	7-1-5-2	1010011	4-4-4-4		
12	7-1-5-4	1011001	4-4-4-4		
13	6-1-4-7	1101001	3-3-3-3		
14	7-1-5-6	1110001	4-4-4-4		

Table 6.5 lists five part assembly subsets with unique assembly index value, their assembly direction matrix along with alternate possibilities.

Table 6.5: Five part assembly subsets with unique assembly index value

S. No.	**Assembly Sequences**	**Assembly Index**	**Assembly direction**	**Alternate sequence**	**Assembly direction**
1	1-4-2-3-5	11111	3-3-3-3-3	1-5-2-3-4	4-4-4-4-4
2	1-2-3-4-6	101111	4-4-4-4-4	6-1-4-2-3	3-3-3-3-3
3	1-5-2-3-6	110111	4-4-4-4-4		
4	1-5-2-4-6	111011	4-4-4-4-4	6-1-4-2-5	3-3-3-3-3
5	1-4-2-3-7	1001111	3-3-3-3-3		
6	1-2-3-5-7	1010111	3-3-3-3-3	7-1-5-2-3	4-4-4-4-4
7	1-4-2-5-7	1011011	3-3-3-3-3	7-1-5-2-4	4-4-4-4-4
8	6-1-4-2-7	1101011	3-3-3-3-3		
9	7-1-5-2-6	1110011	4-4-4-4-4		

Table 6.6 lists six part assembly subsets with unique assembly index value, their assembly direction matrix along with alternate possibilities.

Table 6.6: Six part assembly subsets with unique assembly index value

S. No.	Assembly Sequences	Assembly Index	Assembly direction	Alternate sequence	Assembly direction
1	1-5-2-3-4-6	111111	4-4-4-4-4-4	6-1-4-2-3-5	3-3-3-3-3-3
2	1-4-2-3-5-7	1011111	3-3-3-3-3-3	7-1-5-2-3-4	4-4-4-4-4-4
3	6-1-4-2-3-7	1101111	3-3-3-3-3-3	6-1-4-2-7-3	3-3-3-3-3-3
4	7-1-5-2-3-6	1110111	4-4-4-4-4-4	7-1-5-2-6-3	4-4-4-4-4-4
5	6-1-4-2-5-7	1111011	3-3-3-3-3-3	7-1-5-2-4-6	4-4-4-4-4-4

The seven part subsets are the desired optimal assembly sequences with alternate possibilities. Table 6.7 lists optimal assembly sequence and its alternate possible solutions with assembly direction matrix.

Table 6.7: Optimal assembly sequences alternate possibilities

S. No.	Assembly Sequences	Assembly Index	Assembly direction	Alternate sequence	Assembly direction
1	6-1-4-2-3-5	1111111	3-3-3-3-3-3	7-1-5-2-3-4	4-4-4-4-4-4

Intermediate assembly subsets and computational time comparisons are listed in Table 6.8 while solving for set of all possible solutions while solving for gear assembly.

Table 6.8: Set of all possible intermediate assembly subsets vs optimal subsets

Assembly subsets	Set of all possible solutions	Optimal solutions
2-part subsets	12	12
3-part subsets	44	22
4-part subsets	112	18
5-part subsets	200	14
6-part subsets	208	10
7-part subsets	64	2

Graphical representation of optimal assembly subsets contrasted with set of all possible assembly sets are presented in Figure 6.2.

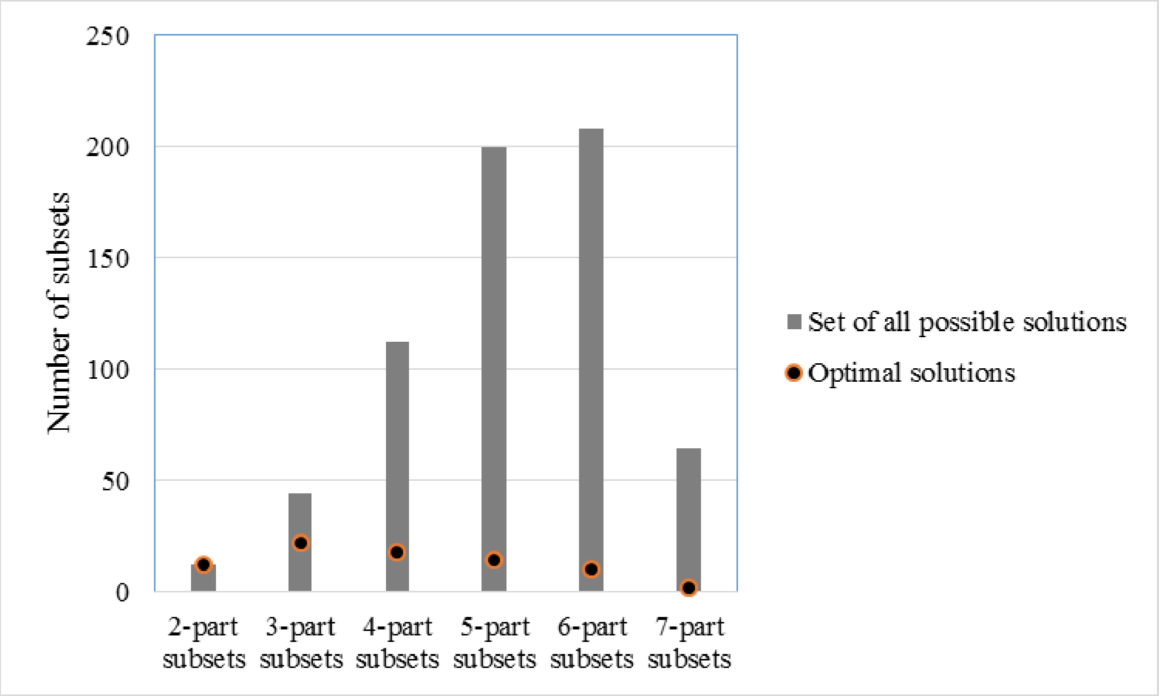


Figure 6.2: Set of all intermediate assembly subsets and optimal subsets for gear assembly

Achieving optimal assembly sequences for a defined objective function reduce the computational time due to reduced number of sets at the intermediate level, however eliminating the redundant sets slightly increase the computational time.

6.3 Summary

This chapter presents the concept of assembly indexing to eliminate the redundant assembly subsets with same assembly index and high fitness value to reduce the overall computational time while retrieving the optimal assembly sequences. The part concatenation method integrated with assembly indexing for a defined objective function improved the computational performance and resulted the outcomes accurately within less span of time.

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Overview

The various work carried out for completing this research have been presented elaborately in the previous chapters. Present chapter is dedicated to mention salient points and key findings of the entire work in the form of conclusions. The scope of future work to extend or to modify or to add some new concept to the work is also suggested in the present chapter.

7.2 Conclusions

This section, in nutshell, presents the achievements made through this research work, difficulties faced and key findings of the research work.

1. In order to address the complexities with the manual assembly information extraction, efficient computer aided methods have been developed by considering basic capabilities of solid modelling, assembly modelling and laws of physical equilibrium to extract assembly attributes. The automated methods are found to be efficient to extract information from CAD environment.
2. Efficient algorithms for assembly predicate testing considering assembly attributes are developed to test the quality and practical feasibility of an assembly subset.
3. Most of the part research literature did not consider all the assembly predicates for several reasons; in the present research work, Influence of assembly predicate consideration on computational time and quality of outcomes is illustrated for different assembly configurations.
4. A novel concatenation method is developed to generate set of all feasible assembly sequences considering all necessary predicates. The method is implemented on different products and is found to be successful and efficient in generating feasible assembly sequences. The procedure for selecting optimal sequences from set of all feasible sequences considering assembly directional changes, gripper changes and assembly energy is briefly illustrated.
5. The results obtained using concatenation method are proven global optimal with respect to the existing literature for the exampled assemblies.
6. A novel technique named assembly indexing is proposed to minimize the computational time by eliminating the possible similar assembly configurations with different patterns

at intermediate levels of the concatenation method. The proposed method is tested on products with large number of parts.

7. The major difficulty faced during the research work is interfacing with CAD environment to test partial stability between pair of parts using equilibrium of the bodies without the friction between the parts.

7.3 Contributions

The major contributions of the current work towards the assembly sequence generation problem are:

1 The effect of assembly predicate consideration on optimal ASG is provided for different assembly configuration to ensure appropriate result. The observations made by this study are further helpful in linear and parallel assembly systems.

2 The proposed stability representation eases the process of CAD-based ASG

3 The proposed concept of mechanical feasibility reduces the standard part count enormously and enhances the computational performance.

4 A novel and knowledge based method has been proposed to generate set of all feasible assembly sequences.

5 A novel assembly indexing method is proposed to reduce the computation time during optimal assembly sequence generation.

7.4 Future Research Scope

Although every consideration has been made for developing an efficient procedure for assembly sequence generation of mechanical parts for small and large products and the objectives of the research work has been fully achieved, there remain few scopes for further work of the present problem, some of these can be as follows.

- ✓ The proposed concatenation method to perform assembly sequence generation can also be extended to detect stable subassemblies for solving parallel assembly systems using the concept of partial and permanent stability.
- ✓ The proposed concatenation method is flexible to extend for solving complicated products with oblique assembly orientations (other than principal axes directions).
- ✓ By integrating the material database and functional requirement of parts and manufacturing testing procedures, the method can be used for DFA to reduce the number of parts of a mechanical product.

- ✓ The proposed method can also be extended for environmental and economic disassembly sequence detection for product end of life and repair and manufacturing processes.
- ✓ The proposed method can be extended to generate automated exploded view from CAD product

REFERENCES

ALFADHLANI, SAMADHI, T.A., MA'RUF, A.N.A.S. and TOHA, I.S., 2011. Automatic collision detection for assembly sequence planning using a three-dimensional solid model. Journal of Advanced Manufacturing Systems, 10(02), pp.277-291.

Anantha, R., Kramer, G.A. and Crawford, R.H., 1996. Assembly modelling by geometric constraint satisfaction. Computer-Aided Design, 28(9), pp.707-722.

Bai, Y.W., Chen, Z.N., Bin, H.Z. and Hun, J., 2005. An effective integration approach toward assembly sequence planning and evaluation. The International Journal of Advanced Manufacturing Technology, 27(1-2), pp.96-105.

Baldwin, D.F., Abell, T.E., Lui, M.C.M., De Fazio, T.L. and Whitney, D.E., 1991. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. Robotics and Automation, IEEE Transactions on, 7(1), pp.78-94.

Biswal, B.B., Deepak, B.B. and Rao, Y., 2013. Optimization of robotic assembly sequences using immune based technique. Journal of Manufacturing Technology Management, 24(3), pp.384-396.

Bonneville, F., Perrard, C. and Henrioud, J.M., 1995, October. A genetic algorithm to generate and evaluate assembly plans. In Emerging Technologies and Factory Automation, 1995. ETFA'95, Proceedings., 1995 INRIA/IEEE Symposium on (Vol. 2, pp. 231-239). IEEE.

Boothroyd, G., Dewhurst, P., & Knight, W. A. (2010). Product design for manufacture and assembly. CRC Press.

Bourjault A. Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des séquences opératoires. PhD Thesis, Universite de Franche-Comte, France, 1984.

Bullinger, H. J. and Ammer, E. D., (1984), Computer aided depicting of precedence diagrams-a step towards efficient planning in assembly, Computing and Industrial Engineering, Vol.18, No.3/4, pp165-169.

Cao, P.B. and Xiao, R.B., 2007. Assembly planning using a novel immune approach. The International Journal of Advanced Manufacturing Technology, 31(7-8), pp.770-782.

Chang, C.C., Tseng, H.E. and Meng, L.P., 2009. Artificial immune systems for assembly sequence planning exploration. Engineering Applications of Artificial Intelligence, 22(8), pp.1218-1232.

Chen, C.P. and Pao, Y.H., 1993. An integration of neural network and rule-based systems for design and planning of mechanical assemblies. Systems, Man and Cybernetics, IEEE Transactions on, 23(5), pp.1359-1371.

Chen, R.S., Lu, K.Y. and Yu, S.C., 2002. A hybrid genetic algorithm approach on multi-objective of assembly planning problem. Engineering Applications of Artificial Intelligence, 15(5), pp.447-457.

Chen, S.F. and Liu, Y.J., 2001. An adaptive genetic assembly-sequence planner. International Journal of Computer Integrated Manufacturing, 14(5), pp.489-500

Chen, W.C., Tai, P.H., Deng, W.J. and Hsieh, L.F., 2008. A three-stage integrated approach for assembly sequence planning using neural networks. Expert Systems with Applications, 34(3), pp.1777-1786.

Choi, Y.K., Lee, D.M. and Cho, Y.B., 2009. An approach to multi-criteria assembly sequence planning using genetic algorithms. The International Journal of Advanced Manufacturing Technology, 42(1-2), pp.180-188.

Ciszak, O., 2012. Computer aided determination of the assembly sequence of machine parts and sets. Advances in Engineering Software, 48, pp.17-26.

De Fazio, T.L. 2004. Mechanical assemblies: their design, manufacture, and role in product development. Vol. 1. Oxford University Press.

De Fazio, T.L. and Whitney, D.E., 1987. Simplified generation of all mechanical assembly sequences. Robotics and Automation, IEEE Journal of, 3(6), pp.640-658.

De Fazio, T.L., Edsall, A.C., Gustavson, R.E., Hernandez, J., Hutchins, P.M., Leung, H.W., Luby, S.C., Metzinger, R.W., Nevins, J.L., Tung, K. and Whitney, D.E., 1993. A prototype of feature-based design for assembly. Journal of Mechanical Design, 115(4), pp.723-734.

De Lit, P., Latinne, P., Rekiek, B. and Delchambre, A., 2001. Assembly planning with an ordering genetic algorithm. International Journal of Production Research, 39(16), pp.3623-3640.

De Mello, L.S.H. and Sanderson, A.C., 1990. AND/OR graph representation of assembly plans. Robotics and Automation, IEEE Transactions on, 6(2), pp.188-199.

De Mello, L.S.H. and Sanderson, A.C., 1991a. A correct and complete algorithm for the generation of mechanical assembly sequences. Robotics and Automation, IEEE Transactions on, 7(2), pp.228-240.

De Mello, L.S.H. and Sanderson, A.C., 1991b. Representations of mechanical assembly sequences. Robotics and Automation, IEEE Transactions on, 7(2), pp.211-227.

De Mello, L.S.H. and Sanderson, A.C., Automatic Generation of Mechanical Assembly Sequences. Report. Report No. CMU-FU-TR-88- 19, Camegie Mellon University, Dec 1988.

Del Valle, C., Gasca, R.M., Toro, M. and Camacho, E.F., 2003. A genetic algorithm for assembly sequence planning. In Artificial Neural Nets Problem Solving Methods (pp. 337-344). Springer Berlin Heidelberg.

Dini, G., Failli, F., Lazzerini, B. and Marcelloni, F., 1999. Generation of optimized assembly sequences using genetic algorithms. CIRP Annals-Manufacturing Technology, 48(1), pp.17-20.

Dong, T., Tong, R., Zhang, L. and Dong, J., 2007. A knowledge-based approach to assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 32(11-12), pp.1232-1244.

Eng, T.H., Ling, Z.K., Olson, W. and McLean, C., 1999. Feature-based assembly modeling and sequence generation. Computers & Industrial Engineering, 36(1), pp.17-33.

Failli, F. and Dini, G., 2000, June. Ant colony systems in assembly planning: a new approach to sequence detection and optimization. In Proceedings of the 2nd CIRP international seminar on intelligent computation in manufacturing engineering (pp. 227-232).

Floriani, L.D. and Nagy, G., 1989, May. A graph model for face-to-face assembly. In Robotics and Automation, 1989. Proceedings. 1989 IEEE International Conference on (pp. 75-78). IEEE.

Gao, L., Qian, W., Li, X. and Wang, J., 2010. Application of memetic algorithm in assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 49(9), pp.1175-1184.

Ghandi, S. and Masehian, E., 2015. A breakout local search (BLS) method for solving the assembly sequence planning problem. Engineering Applications of Artificial Intelligence, 39, pp.245-266.

Giri, R. and Kanthababu, M., 2015. Generating complete disassembly sequences by utilising two-dimensional views. International Journal of Production Research, (ahead-of-print), pp.1-21.

Gu, P. and Yan, X., 1995. CAD-directed automatic assembly sequence planning. International Journal of Production Research, 33(11), pp.3069-3100.

Guan, Q., Liu, J.H. and Zhong, Y.F., 2002. A concurrent hierarchical evolution approach to assembly process planning. International Journal of Production Research, 40(14), pp.3357-3374.

Guo, J., Tang, H., Sun, Z., Wang, S., Jia, X., Chen, H. and Zhang, Z., 2015. An Improved Shuffled Frog Leaping Algorithm for Assembly Sequence Planning of Remote Handling Maintenance in Radioactive Environment. Science and Technology of Nuclear Installations.

Hadj, R.B., Trigui, M. and Aifaoui, N., 2015. Toward an integrated CAD assembly sequence planning solution. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 229(16), pp.2987-3001.

Hoffman, R., 1989, May. Automated assembly in a CSG domain. In Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on (pp. 210-215). IEEE.

Hoffman, R., 1990, August. Automated assembly planning for B-rep products. In Systems Engineering, 1990., IEEE International Conference on (pp. 391-394). IEEE.

Hong, D.S. and Cho, H.S., 1993, July. Optimization of robotic assembly sequences using neural network. In Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on (Vol. 1, pp. 232-239). IEEE.

Hong, D.S. and Cho, H.S., 1995. A neural-network-based computational scheme for generating optimized robotic assembly sequences. Engineering applications of artificial intelligence, 8(2), pp.129-145.

Hong, D.S. and Cho, H.S., 1997. Generation of robotic assembly sequences with consideration of line balancing using simulated annealing. Robotica, 15(06), pp.663-673.

Hong, D.S. and Cho, H.S., 1999. A genetic-algorithm-based approach to the generation of robotic assembly sequences. Control Engineering Practice, 7(2), pp.151-159.

Hong, D.S. and Cho, H.S., 1999. Generation of robotic assembly sequences using a simulated annealing. In Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on (Vol. 2, pp. 1247-1252). IEEE.

Hsu, Y.Y., Tai, P.H., Wang, M.W. and Chen, W.C., 2011. A knowledge-based engineering system for assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 55(5-8), pp.763-782.

Huang, J., Du, P.A. and Liao, W.Z., 2007. Genetic algorithm for assembly sequences planning based on assembly constraint. COMPUTER INTEGRATED MANUFACTURING SYSTEMS-BEIJING-, 13(4), p.756.

Huang, Y.F. and Lee, C.G., 1991, April. A framework of knowledge-based assembly planning. In Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on (pp. 599-604). IEEE.

Hui, W., Dong, X. and Guanghong, D., 2008. A genetic algorithm for product disassembly sequence planning. Neurocomputing, 71(13), pp.2720-2726.

Ibrahim, I., Ibrahim, Z., Ahmad, H., Jusof, M.F.M., Yusof, Z.M., Nawawi, S.W. and Mubin, M., 2015. An assembly sequence planning approach with a rule-based multi-state gravitational search algorithm. The International Journal of Advanced Manufacturing Technology, 79(5-8), pp.1363-1376.

Kalpakjian, S. and Schmid, S.R. 2008. “Manufacturing processes for engineering materials” (Vol. 5). Pearson-Prentice Hall.

Kanai, S., Takahashi, H. and Makino, H., 1996. ASPEN: computer-aided assembly sequence planning and evaluation system based on predetermined time standard. CIRP Annals-Manufacturing Technology, 45(1), pp.35-39.

Kashkoush, M. and ElMaraghy, H., 2015. Knowledge-based model for constructing master assembly sequence. Journal of Manufacturing Systems, 34, pp.43-52.

Khosla, P.K. and Mattikali, R., 1989. Determining the assembly sequence from a 3-D model. Journal of Mechanical Working Technology, 20, pp.153-162.

Kim, M.G. and Wu, C.H., 1990, November. A formal part mating model for generating compliance control strategies of assembly operations. In Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on (pp. 611-616). IEEE.

Kuo T, Huang S, Zhang H (2002) Design for manufacture and design for ‘X’: concepts, applications, and perspectives. Computers & Industrial Engineering, 41(3), 241-260.

Lazzerini, B. and Marcelloni, F., 2000. A genetic algorithm for generating optimal assembly plans. Artificial Intelligence in Engineering, 14(4), pp.319-329.

Lee, H.R. and Gemmill, D.D., 2001. Improved methods of assembly sequence determination for automatic assembly systems. European Journal of Operational Research, 131(3), pp.611-621.

Lee, S. and Shin, Y.G., 1990. Assembly planning based on geometric reasoning. Computers & graphics, 14(2), pp.237-250.

Li, M., Wu, B., Hu, Y., Jin, C. and Shi, T., 2013. A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation. The International Journal of Advanced Manufacturing Technology, 68(1-4), pp.617-630.

Li, Y.L., Gao, J.M., Shi, L. and Wang, S., 2012. Assembly modeling for 3d component based on polychromatic sets. In Advanced Materials Research, Vol. 411, pp. 388-392.

Ling, Z.K. and Narayan, S., 1996. Heuristic feature recognition from a CAD model. Integrated Computer-Aided Engineering, 3(3), pp.212-224.

Linn, R.J. and Liu, H., 1999. An automatic assembly liaison extraction method and assembly liaison model. IIE transactions, 31(4), pp.353-363.

Liu, Y. and Popplestone, R.J., 1989, May. Planning for assembly from solid models. In Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on (pp. 222-227). IEEE.

Lu, C. and Yang, Z., 2015. Integrated assembly sequence planning and assembly line balancing with ant colony optimization approach. The International Journal of Advanced Manufacturing Technology, pp.1-14.

Lui, M.C.M., 1988. Generation and evaluation of mechanical assembly sequences using the liaison-sequence method (Doctoral dissertation, Massachusetts Institute of Technology).

Lv, H. and Lu, C., 2010. An assembly sequence planning approach with a discrete particle swarm optimization algorithm. The international journal of advanced manufacturing technology, 50(5-8), pp.761-770.

Marian, R.M., Luong, L.H. and Abhary, K., 2003. Assembly sequence planning and optimisation using genetic algorithms: part I. Automatic generation of feasible assembly sequences. Applied Soft Computing, 2(3), pp.223-253.

Marian, R.M., Luong, L.H. and Abhary, K., 2006. A genetic algorithm for the optimisation of assembly sequences. Computers & Industrial Engineering, 50(4), pp.503-527.

Mathew, A. and Rao, C.S.P., 2010. A CAD system for extraction of mating features in an assembly. Assembly Automation, 30(2), pp.142-146.

Mathew, A.T. and Rao, C.S.P., 2010. A novel method of using API to generate liaison relationships from an assembly. Journal of Software Engineering and Applications, 3(02), p.167.

McGovern, S.M. and Gupta, S.M., 2006. Ant colony optimization for disassembly sequencing with multiple objectives. The International Journal of Advanced Manufacturing Technology, 30(5-6), pp.481-496.

Milner, J.M., Graves, S.C. and Whitney, D.E., 1994, May. Using simulated annealing to select least-cost assembly sequences. In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on (pp. 2058-2063). IEEE.

Mok, S.M., Ong, K. and Wu, C.H., 2001. Automatic generation of assembly instructions using STEP. In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on (Vol. 1, pp. 313-318). IEEE.

Motavalli, S. and Islam, A.U., 1997. Multi-criteria assembly sequencing. Computers & industrial engineering, 32(4), pp.743-751.

Ning, L.H. and Gu, T.L., 2007. Hybrid algorithm for assembly sequence planning. COMPUTER INTEGRATED MANUFACTURING SYSTEMS-BEIJING-, 13(4), p.762.

Ou, L.M. and Xu, X., 2013. Relationship matrix based automatic assembly sequence generation from a CAD model. Computer-Aided Design, 45(7), pp.1053-1067.

Pan, C., Smith, S.S.F. and Smith, G.C., 2005. Determining interference between parts in CAD STEP files for automatic assembly planning. Journal of Computing and Information Science in Engineering, 5(1), pp.56-62.

Prabhakar, S. and Henderson, M.R., 1992. Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models. Computer-Aided Design, 24(7), pp.381-393.

Prenting T, Battaglin R (1964) The precedence diagram: A tool for analysis in assembly line balancing. Journal of Industrial Engineering. 15(4): 208-213.

Rashid, M.F.F., Hutabarat, W. and Tiwari, A., 2012. A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. The International Journal of Advanced Manufacturing Technology, 59(1-4), pp.335-349.

Sanderson, A.C., De Mello, L.S.H. and Zhang, H., 1990. Assembly sequence planning. AI Magazine, 11(1), p.62.

Senin, N., Groppetti, R. and Wallace, D.R., 2000. Concurrent assembly planning with genetic algorithms. Robotics and Computer-Integrated Manufacturing, 16(1), pp.65-72.

Shah, J.J. and Rogers, M.T., 1993. Assembly modeling as an extension of feature-based design. Research in Engineering Design, 5(3-4), pp.218-237.

Shan, H., Li, S., Gong, D. and Lou, P., 2006, November. Genetic simulated annealing algorithm-based assembly sequence planning. In Technology and Innovation Conference, 2006. ITIC 2006. International (pp. 1573-1579). IET.

Shan, H., Zhou, S. and Sun, Z., 2009. Research on assembly sequence planning based on genetic simulated annealing algorithm and ant colony optimization algorithm. Assembly Automation, 29(3), pp.249-256.

Sharma, S., Biswal, B.B., Dash, P. and Choudhury, B.B., 2008, August. Generation of optimized robotic assembly sequence using ant colony optimization. In Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on (pp. 894-899). IEEE.

Shi, S.C., Li, R., Fu, Y.L. and Ma, Y.L., 2010. Assembly sequence planning based on improved ant colony algorithm. Computer Integrated Manufacturing Systems, 16(6), pp.1189-1194.

Shpitalni, M., Elber, G. and Lenz, E., 1989. Automatic assembly of three-dimensional structures via connectivity graphs. CIRP Annals-Manufacturing Technology, 38(1), pp.25-28.

Shuang, B., Chen, J. and Li, Z., 2008. Microrobot based micro-assembly sequence planning with hybrid ant colony algorithm. The International Journal of Advanced Manufacturing Technology, 38(11-12), pp.1227-1235.

Sinanoglu, C. and Riza Börklü, H., 2005. An assembly sequence-planning system for mechanical parts using neural network. Assembly Automation, 25(1), pp.38-52.

Smith, G.C. and Smith, S.S.F., 2002. An enhanced genetic algorithm for automated assembly planning. Robotics and Computer-Integrated Manufacturing, 18(5), pp.355-364.

Smith, S.S.F. and Liu, Y.J., 2001. The application of multi-level genetic algorithms in assembly planning. Journal of Industrial Technology, 17(4), p.1-4.

Smith, S.S.F., 2004. Using multiple genetic operators to reduce premature convergence in genetic assembly planning. Computers in Industry, 54(1), pp.35-49.

Smith, S.S.F., Smith, G.C. and Liao, X., 2001. Automatic stable assembly sequence generation and evaluation. Journal of Manufacturing Systems, 20(4), pp.225-235.

Su, Q., 2009. A hierarchical approach on assembly sequence planning and optimal sequences analyzing. Robotics and Computer-Integrated Manufacturing, 25(1), pp.224-234.

Sung, R.C., Corney, J.R. and Clark, D.E., 2001. Automatic assembly feature recognition and disassembly sequence generation. Journal of Computing and Information Science in Engineering, 1(4), pp.291-299.

Tseng, H.E. and Li, R.K., 1999. A novel means of generating assembly sequences using the connector concept. Journal of Intelligent Manufacturing,10(5), pp.423-435.

Tseng, H.E., Chen, M.H., Chang, C.C. and Wang, W.P., 2008. Hybrid evolutionary multi-objective algorithms for integrating assembly sequence planning and assembly line balancing. International Journal of Production Research, 46(21), pp.5951-5977.

Tseng, H.E., Li, J.D. and Chang, Y.H., 2004. Connector-based approach to assembly planning using a genetic algorithm. International Journal of Production Research, 42(11), pp.2243-2261

Tseng, H.E., Wang, W.P. and Shih, H.Y., 2007. Using memetic algorithms with guided local search to solve assembly sequence planning. Expert Systems with Applications, 33(2), pp.451-467.

Tseng, Y.J., Chen, J.Y. and Huang, F.Y., 2010. A multi-plant assembly sequence planning model with integrated assembly sequence planning and plant assignment using GA. The International Journal of Advanced Manufacturing Technology, 48(1-4), pp.333-345.

Tseng, Y.J., Yu, F.Y. and Huang, F.Y., 2011. A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method. The International Journal of Advanced Manufacturing Technology, 57(9), pp.1183-1197.

Van Holland, W. and Bronsvoort, W.F., 2000. Assembly features in modeling and planning. Robotics and computer-integrated manufacturing, 16(4), pp.277-294.

Viganò, R. and Gomez, G.O, 2012. Assembly planning with automated retrieval of assembly sequences from CAD model information. Assembly Automation, 32(4), pp.347-360.

Viganò, R. and Gómez, G.O., 2013. Automatic assembly sequence exploration without precedence definition. International Journal on Interactive Design and Manufacturing (IJIDeM), 7(2), pp.79-89.

Wang, H., Rong, Y. and Xiang, D., 2014. Mechanical assembly planning using ant colony optimization. Computer-Aided Design, 47, pp.59-71.

Wang, J.F., Liu, J.H. and Zhong, Y.F., 2005. A novel ant colony algorithm for assembly sequence planning. The international journal of advanced manufacturing technology, 25(11-12), pp.1137-1143.

Wang, J.F., Liu, J.H., Li, S.Q. and Zhong, Y.F., 2003. Intelligent selective disassembly using the ant colony algorithm. AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 17(04), pp.325-333.

Wang, Y. and Liu, J., 2013. Subassembly identification for assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 68(1-4), pp.781-793.

Wang, Y. and Liu, J.H., 2010. Chaotic particle swarm optimization for assembly sequence planning. Robotics and Computer-Integrated Manufacturing, 26(2), pp.212-222.

Wilson, R.H. and Latombe, J.C., 1994. Geometric reasoning about mechanical assembly. Artificial Intelligence, 71(2), pp.371-396.

Wolter, J.D., 1989. On the Automatic Generation of Assembly Plans. .In Proceedings IEEE International Conference on Robotics and Automation (pp. 62-68). IEEE.

Wong, H. and Leu, M.C., 1993. Adaptive genetic algorithm for optimal printed circuit board assembly planning. CIRP Annals-Manufacturing Technology, 42(1), pp.17-20.

Wu, C.H. and Kim, M.G., 1994. Modeling of part-mating strategies for automating assembly operations for robots. Systems, Man and Cybernetics, IEEE Transactions on, 24(7), pp.1065-1074.

Yin, Z., Ding, H., Li, H. and Xiong, Y., 2003. A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. Computer-Aided Design, 35(1), pp.37-56.

YU, H., WANG, C.E., YU, J.P. and YUAN, H., 2010. Assembly Sequence Planning Based on Particle Swarm Optimization Algorithm for Complex Product [J]. Journal of Northeastern University (Natural Science), 2, p.028.

Yu, J. and Wang, C., 2013. A max–min ant colony system for assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 67(9-12), pp.2819-2835.

Yu, J. and Wang, C., 2013. Method for discriminating geometric feasibility in assembly planning based on extended and turning interference matrix. The International Journal of Advanced Manufacturing Technology, 67(5-8), pp.1867-1882.

Yu, J., Da Xu, L., Bi, Z. and Wang, C., 2014. Extended interference matrices for exploded view of assembly planning. Automation Science and Engineering, IEEE Transactions on, 11(1), pp.279-286.

Zeng, C., Gu, T., Zhong, Y. and Cai, G., 2011. A multi-agent evolutionary algorithm for connector-based assembly sequence planning. Procedia Engineering, 15, pp.3689-3693.

Zha, X.F., Lim, S.Y. and Fok, S.C., 1998. Integrated knowledge-based assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 14(1), pp.50-64.

Zhang, H., Liu, H. and Li, L., 2014. Research on a kind of assembly sequence planning based on immune algorithm and particle swarm optimization algorithm. The International Journal of Advanced Manufacturing Technology, 71(5-8), pp.795-808.

Zhou, W., Yan, J., Li, Y., Xia, C. and Zheng, J., 2013. Imperialist competitive algorithm for assembly sequence planning. The International Journal of Advanced Manufacturing Technology, 67(9-12), pp.2207-2216.

Zhou, W., Zheng, J.R., Yan, J.J. and Wang, J.F., 2011. A novel hybrid algorithm for assembly sequence planning combining bacterial chemotaxis with genetic algorithm. The International Journal of Advanced Manufacturing Technology, 52(5-8), pp.715-724.

DISSEMINATION

Journal Articles

1. **Bahubalendruni, M. R.**, & Biswal, B. B. (2016). A review on assembly sequence generation and its automation. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(5), 824-838.
2. **Bahubalendruni, M. R.**, & Biswal, B. B. (2016). Liaison Concatenation - A Method To Obtain Feasible Assembly Sequences From 3d-Cad Product. *Sadhana*, 41(1),67-74.
3. **Bahubalendruni, M. R.**, Deepak, B.B & Biswal, B. B. (2016). An Advanced Immune Based Strategy to Obtain an Optimal Feasible Assembly Sequence. *Assembly Automation*, *36*(2),127-137.
4. **Bahubalendruni, M. R.**, Biswal, B. B. & Deepak, B.B, (2016). Optimal Robotic Assembly Sequence Generation Using Particle Swarm Optimization. *Journal of Automation and Control Engineering*, *4*(2),89-95.
5. **Bahubalendruni, M. R.**, Biswal, B. B., Kumar, M., & Nayak, R. (2015). Influence of assembly predicate consideration on optimal assembly sequence generation. *Assembly Automation*, *35*(4), 309-316.
6. **Bahubalendruni, M. R.**, & Biswal, B. B. (2015). A novel concatenation method for generating optimal robotic assembly sequences. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 0954406215623813.
7. **Bahubalendruni, M. V. A.**, Biswal, B. B., & Khanolkar, G. R. (2015). A Review on Graphical Assembly Sequence Representation Methods and Their Advancements. *Journal of Mechatronics and Automation*, *1*(2), 16-26.
8. **Bahubalendruni, M. V. A.**, & Biswal, B. B. (2014). An algorithm to test feasibility predicate for robotic assemblies. *Trends in Mechanical Engineering & Technology*, *4*(2), 11-16.

Conference Presentations

1. **Bahubalendruni, M. R.**, Biswal, B. B., Kumar, M., & Deepak, B. B. V. L. (2016). A Note on Mechanical Feasibility Predicate for Robotic Assembly Sequence

Generation. In *CAD/CAM, Robotics and Factories of the Future* (pp. 397-404). Springer India.

2. Nayak, R., **Bahubalendruni, M. R.**, Biswal, B. B., & Kumar, M. (2015) Comparison of Liaison Concatenation Method with Simulated Annealing for Assembly Sequence Generation Problems. In Next Generation Computing Technologies (NGCT-2015).IEEE.
3. **Bahubalendruni, M. R.**, & Biswal, B. B. (2015). An Intelligent Method to Test Feasibility Predicate for Robotic Assembly Sequence Generation. In *Intelligent Computing, Communication and Devices* (pp. 277-283). Springer India.
4. **Bahubalendruni, M. V. A.**, & Biswal, B. B. (2014, January). Computer aid for automatic liaisons extraction from cad based robotic assembly. In Intelligent Systems and Control (ISCO), 2014 IEEE 8th International Conference on (pp. 42-45). IEEE.
5. **Bahubalendruni, M. R.**, Biswal, B. B., & Upadhyaya, V. (2014) Assembly Sequence Generation and Automation. *In Proceeding of International Symposium on Engineering and Technology, pp185-192.*

Under Communication:

1. **Bahubalendruni, M. R.**, & Biswal, B. B. Computer Aid for Stability Testing Between Parts towards Automatic Assembly Sequence Generation. S*adhana.* (Under Review)
2. **Bahubalendruni, M. R.**, & Biswal, B. B. An Efficient Stable Subassembly Identification Method towards Assembly Sequence Generation. National Academy Science Letters. (Under Review)
3. **Bahubalendruni, M. R.**, & Biswal, B. B. An Efficient Concatenation Approach towards Optimal Robotic Assembly Sequence Generation. *Assembly Automation* (Awaiting Reviewer Scores)
4. **Bahubalendruni, M. R.**, & Biswal, B. B. Computer Aided Assembly Attributes Retrieval Methods for Automated Assembly Sequence Generation. *International Journal of Computer Integrated Manufacturing* (Under Review)

List of Patents

1. **Bahubalendruni, M. R.**, & Biswal, B. B. KNOWLEDGE BASED METHOD FOR OPTIMAL ASSEMBLY SEQUENCE GENERATION.
2. **Bahubalendruni, M. R.**, & Biswal, B. B. METHOD TO TEST STABILITY OF PARTS DURING ASSEMBLY SEQUENCE GENERATION.

APPENDICES

PROGRAM: 1

To generate liaison matrix for a product opened in CATIA V5 environment.

```
Sub CATMain()
Inpu = 5
Set Prod=CATIA.ActiveDocument.Product
Set PPss=Prod.Products
nop1= PPss.count
Set sel=CATIA.ActiveDocument.Selection
temp=0
For i=1 to nop1
  pnam=PPss.Item(i).Name
  If (Left(UCase(pnam), 3) = "PIN" OR Left(UCase(pnam), 3) = "NUT" OR
Left(UCase(pnam), 4) = "BOLT" OR Left(UCase(pnam), 5) = "SCREW" OR
Left(UCase(pnam), 5) = "RIVET") Then
          set par=PPss.Item(i)
          sel.add par
          temp=temp+1
  End If
Next
set vis= sel.VisProperties
vis.SetShow 1
sel.clear
nop=nop1-temp
Set xlApp = CreateObject("EXCEL.Application")
          xlApp.Visible = True
          xlApp.Workbooks.Add
          xlApp.Range("i5:j5").Font.Name = "Century Gothic"
          xlApp.Range("i5:j5").HorizontalAlignment = 3
          xlApp.Range("i5:j5").Font.Size =24
          xlApp.Sheets.item(1).Cells(5,9).Value = "Liaison Matrix"
          xlApp.Range("G8:XFD8").Font.Name = "Rockwell Condensed"
          xlApp.Range("h8:XFD500").HorizontalAlignment = 3
          xlApp.Range("G8:G500").Font.Name = "Rockwell Condensed"
          xlApp.Range("G8:XFD8").Font.Size =12
          xlApp.Range("G8:G500").Font.Size =12
          xlApp.Range("A8:XFD8").ColumnWidth = 15
          Set sel1=CATIA.ActiveDocument.Selection
          filename=PPss.name
          Set cClashes =
CATIA.ActiveDocument.Product.GetTechnologicalObject("Clashes")
          Set oClash = cClashes.AddFromSel
          oClash.ComputationType = 0
          oClash.InterferenceType = 1
          oClash.Clearance=Inpu
```

```
        oClash.Compute
        Set cConflicts = oClash.Conflicts
        for i=1 to cConflicts.count
                Set oConflict = cConflicts.Item(i)
                CV=oConflict.Value
                xlApp.Sheets.item(1).Cells(5+i,2).Value=i
                xlApp.Sheets.item(1).Cells(5+i,3).Value=oConflict.FirstProduct.Name

        xlApp.Sheets.item(1).Cells(5+i,4).Value=oConflict.SecondProduct.Name
                xlApp.Sheets.item(1).Cells(5+i,5).Value=CV
        Next
        for i=1 to nop
          xlApp.Sheets.item(1).Cells(8,7+i).Value=PPss.item(i).name
          xlApp.Sheets.item(1).Cells(8+i,7).Value=PPss.item(i).name
        Next
        for i=1 to nop
          for j=1 to nop
                xlApp.Sheets.item(1).Cells(8+i,7+j).Value=0
          Next
        Next
        for i=1 to cConflicts.count
                Set oConflict = cConflicts.Item(i)
                If (oConflict.Value > -0.01 and oConflict.Value < 0.01 ) Then
       for j=1 to nop
                If (PPss.item(j).name=oConflict.FirstProduct.Name) Then
                                        rr=j
                                         j=nop
                                End If
                  Next
       for k=1 to nop
                        If(PPss.item(k).name=oConflict.SecondProduct.Name) Then
                                        cc=k
                                         k=nop
                                End If
                  Next
                  xlApp.Sheets.item(1).Cells(8+rr,7+cc).Value=1
                  xlApp.Sheets.item(1).Cells(8+cc,7+rr).Value=1
        End If
        Next
Msgbox "The Liaison matrix is created"
End Sub
```

PROGRAM: 2

To generate interference free matrix along “X-” direction for a product opened in CATIA V5 environment.

```
Sub CATMain()
Dim movedist(11)
Dim pa(50,8)  'Part Attributes (COG, Volume & Density )
Dim fmyp(50,50)
Set Prod=CATIA.ActiveDocument.Product
Set PPss=Prod.Products
nop1= PPss.count
Set sel=CATIA.ActiveDocument.Selection
temp=0
For i=1 to nop1
  pnam=PPss.Item(i).Name
  If (Left(UCase(pnam), 3) = "PIN" OR Left(UCase(pnam), 3) = "NUT" OR
Left(UCase(pnam), 4) = "BOLT" OR Left(UCase(pnam), 5) = "SCREW" OR
Left(UCase(pnam), 5) = "RIVET") Then
        set par=PPss.Item(i)
        sel.add par
        temp=temp+1
  End If
Next
set vis= sel.VisProperties
vis.SetShow 1
sel.clear
nop=nop1-temp
fpnam=Inputbox("Enter file name(Full Name including Path) ", "Location of the file
and File name" ,"C:\Users\hp\Desktop\Imp Programs\11P1.xls")
Set XL = CreateObject("Excel.Application")
Set objWorkbook = XL.Workbooks.Open(fpnam)
'XL.Application.Visible = True
For i=1 to nop
  For k=1 to 8
    pa(i,k)=XL.Sheets.item(1).Cells(2+i,k+1).Value
  Next
Next
For i=1 to nop
 For j=1 to nop
  If i=j then
   fmyp(i,j)=0
  End If
  dist=pa(j,4)-pa(i,1)
  If dist <=0 then
   fmyp(i,j)=1
  End If
 If dist>0 then
  If Not i=j  then
   For k=1 to nop
     set par=PPss.Item(k)
```

```
        sel.add par
    Next
    set vis= sel.VisProperties
    vis.SetShow 1
    sel.clear
    sel.add PPss.Item(i)
    sel.add PPss.Item(j)
    set vis= sel.VisProperties
    vis.SetShow 0
    sel.clear
   'msgbox i&" - "&j&" - "&dist
    For dst = 1 to dist
        Set prod1 = PPss.Item(i)
        Set mv1 = prod1.Move
        Set mv1 = mv1.MovableObject
        movedist(0) = 1.000000
        movedist(1) = 0.000000
        movedist(2) = 0.000000
        movedist(3) = 0.000000
        movedist(4) = 1.000000
        movedist(5) = 0.000000
        movedist(6) = 0.000000
        movedist(7) = 0.000000
        movedist(8) = 1.000000
        movedist(9) = 1
        movedist(10) = 0.00
        movedist(11) = 0.000000
        mv1.Apply movedist
        Set sel1=CATIA.ActiveDocument.Selection
        Set cClashes =
CATIA.ActiveDocument.Product.GetTechnologicalObject("Clashes")
        Set oClash = cClashes.AddFromSel
        oClash.ComputationType = 0
        oClash.InterferenceType = 1
        oClash.Clearance=5
        oClash.Compute
        Set cConflicts = oClash.Conflicts
        If cConflicts.count > 0 then
        Set oConflict = cConflicts.Item(1)
               CV=oConflict.Value
               If dst=dist AND CV >= 0 then
         fmyp(i,j)=1
                movedist(9) = -dst
                mv1.Apply movedist
             End If
               If CV<0 then
              fmyp(i,j)=0
                movedist(9) = -dst
                mv1.Apply movedist
```

```
                    dst=dist
                End If

           End If
           If cConflicts.count = 0 then
           If dst=dist then
             fmyp(i,j)=1
                    movedist(9) = -dst
                    mv1.Apply movedist
                End If
           End If
      Next
    End If
  End If
   Next
 Next

 msgbox "Its working Fine"
 Set xlApp = CreateObject("EXCEL.Application")
 xlApp.Visible = True
 xlApp.Workbooks.Add
 for i=1 to nop
 for j=1 to nop
 xlApp.Sheets.item(1).Cells(i,j).Value=fmyp(i,j)
 Next
 Next
 End Sub
```

PROGRAM: 3

To generate interference free matrix along “X+” direction for a product opened in CATIA V5 environment.

```
Sub CATMain()

Dim movedist(11)
Dim pa(50,8)   'Part Attributes (COG, Volume & Density )
Dim fmxm(50,50)
Set Prod=CATIA.ActiveDocument.Product
Set PPss=Prod.Products
nop1= PPss.count
Set sel=CATIA.ActiveDocument.Selection
temp=0

For i=1 to nop1
   pnam=PPss.Item(i).Name
   If (Left(UCase(pnam), 3) = "PIN" OR Left(UCase(pnam), 3) = "NUT" OR
Left(UCase(pnam), 4) = "BOLT" OR Left(UCase(pnam), 5) = "SCREW" OR
Left(UCase(pnam), 5) = "RIVET") Then
            set par=PPss.Item(i)
            sel.add par
            temp=temp+1
   End If
Next

set vis= sel.VisProperties
vis.SetShow 1
sel.clear
nop=nop1-temp



fpnam=Inputbox("Enter file name(Full Name including Path) ", "Location of the file
and File name" ,"C:\Users\hp\Desktop\Imp Programs\11P1.xls")
Set XL = CreateObject("Excel.Application")
Set objWorkbook = XL.Workbooks.Open(fpnam)
'XL.Application.Visible = True

For i=1 to nop
   For k=1 to 8
      pa(i,k)=XL.Sheets.item(1).Cells(2+i,k+1).Value
   Next
Next

For i=1 to nop
  For j=1 to nop
   If i=j then
     fmxm(i,j)=0
   End If
   dist=pa(i,4)-pa(j,1)
```

```
   If dist <=0 then
     fmxm(i,j)=1
   End If
 If dist>0 then
   If Not i=j  then
     For k=1 to nop
        set par=PPss.Item(k)
            sel.add par
     Next
     set vis= sel.VisProperties
     vis.SetShow 1
     sel.clear
     sel.add PPss.Item(i)
     sel.add PPss.Item(j)
     set vis= sel.VisProperties
     vis.SetShow 0
     sel.clear
    'msgbox i&" - "&j&" - "&dist
     For dst = 1 to dist
            Set prod1 = PPss.Item(i)
            Set mv1 = prod1.Move
            Set mv1 = mv1.MovableObject
            movedist(0) = 1.000000
            movedist(1) = 0.000000
            movedist(2) = 0.000000
            movedist(3) = 0.000000
            movedist(4) = 1.000000
            movedist(5) = 0.000000
            movedist(6) = 0.000000
            movedist(7) = 0.000000
            movedist(8) = 1.000000
            movedist(9) = -1
            movedist(10) = 0.00
            movedist(11) = 0.000000

            mv1.Apply movedist

            Set sel1=CATIA.ActiveDocument.Selection
            Set cClashes =
CATIA.ActiveDocument.Product.GetTechnologicalObject("Clashes")
            Set oClash = cClashes.AddFromSel
            oClash.ComputationType = 0
            oClash.InterferenceType = 1
            oClash.Clearance=5
            oClash.Compute
            Set cConflicts = oClash.Conflicts
            If cConflicts.count > 0 then
            Set oConflict = cConflicts.Item(1)
                        CV=oConflict.Value
```

```
                        If dst=dist AND CV >= 0 then
              fmxm(i,j)=1
                          movedist(9) = dst
                          mv1.Apply movedist
                    End If
                        If CV<0 then
                      fmxm(i,j)=0
                          movedist(9) = dst
                          mv1.Apply movedist
                          dst=dist
                    End If

            End If
            If cConflicts.count = 0 then
            If dst=dist then
              fmxm(i,j)=1
                          movedist(9) = dst
                          mv1.Apply movedist
                    End If
            End If
     Next
   End If
 End If
  Next
Next

msgbox "Its working Fine"


Set xlApp = CreateObject("EXCEL.Application")
xlApp.Visible = True
xlApp.Workbooks.Add
for i=1 to nop
for j=1 to nop
xlApp.Sheets.item(1).Cells(i,j).Value=fmxm(i,j)
Next
Next

End Sub
```

PROGRAM: 4

To generate partial stability matrix for a product opened in CATIA V5 environment.

```
Dim cog(2)

Sub CATMain()

Dim pa(50,8)  'Part Attributes (COG, Volume & Density )
Dim fmxm(50,50)
Dim lm(50,50)
Dim stb(50,50)
Dim gfm(6,50,50)

Set Prod=CATIA.ActiveDocument.Product
Set PPss=Prod.Products
nop1= PPss.count
Set sel=CATIA.ActiveDocument.Selection
temp=0

For i=1 to nop1
  pnam=PPss.Item(i).Name
  If (Left(UCase(pnam), 3) = "PIN" OR Left(UCase(pnam), 3) = "NUT" OR
Left(UCase(pnam), 4) = "BOLT" OR Left(UCase(pnam), 5) = "SCREW" OR
Left(UCase(pnam), 5) = "RIVET") Then
        set par=PPss.Item(i)
        sel.add par
        temp=temp+1
  End If
Next

set vis= sel.VisProperties
vis.SetShow 1
sel.clear
nop=nop1-temp


fpnam=Inputbox("Enter file name(Full Name including Path) ", "Location of the file
and File name" ,"C:\Users\hp\Desktop\Imp Programs\7p.xls")
Set XL = CreateObject("Excel.Application")
Set objWorkbook = XL.Workbooks.Open(fpnam)
'XL.Application.Visible = True

For i=1 to nop
  For k=1 to nop
    lm(i,k)=XL.Sheets.item(2).Cells(i,k).Value
    gfm(1,i,k)=XL.Sheets.item(4).Cells(i,k).Value
    gfm(2,i,k)=XL.Sheets.item(5).Cells(i,k).Value
    gfm(3,i,k)=XL.Sheets.item(6).Cells(i,k).Value
    gfm(4,i,k)=XL.Sheets.item(7).Cells(i,k).Value
    gfm(5,i,k)=XL.Sheets.item(8).Cells(i,k).Value
```

```
      gfm(6,i,k)=XL.Sheets.item(9).Cells(i,k).Value
   Next
Next


Set Doc=CATIA.ActiveDocument.Product
Set PP = Doc.Products


For j=1 to nop

For i=1 to nop
     stb (i,j) = 0
     For hp=1 to nop

         set par=PPss.Item(hp)
             sel.add par

     Next
     set vis= sel.VisProperties
     vis.SetShow 1
     sel.clear

      set par=PPss.Item(i)
      sel.add par
      set par=PPss.Item(j)
      sel.add par
      set vis= sel.VisProperties
      vis.SetShow 0
      sel.clear


     If lm(j,i)=1 and gfm(6,i,j)=0 Then
            Set FP=PP.Item(i).ReferenceProduct.Parent.Part
            PP.Item(i).ReferenceProduct.Analyze.GetGravityCenter cog
            'msgbox i&"-"&cog(0)&"-"&cog(1)&"-"&cog(2)
            Set H2= FP.HybridBodies.Add
            H2.Name="Lines"
            set l= FP.HybridBodies.item("Lines")
            set m=FP.HybridShapeFactory
            set oo= m.AddNewPointCoord(cog(0),cog(1),cog(2))
            l.AppendHybridShape oo
            FP.update
            set oo= m.AddNewPointCoord(cog(0)+50,cog(1),cog(2))
            l.AppendHybridShape oo
            FP.update
            Set poi1= l.HybridShapes.item(1)
            Set poi2= l.HybridShapes.item(2)
```

```
        Set Ref1 = FP.CreateReferenceFromObject (poi1)
        Set Ref2 = FP.CreateReferenceFromObject (poi2)

        set m=FP.HybridShapeFactory
        set oo= m.AddNewLinePtPt(Ref1,Ref2)
        l.AppendHybridShape oo
        FP.update

        Set lin1 =  l.HybridShapes.item(2)
        Set SFy = FP.ShapeFactory
        Set Ref3 = FP.CreateReferenceFromObject(lin1)
        Set rot1 = SFy.AddNewRotate2(Ref3, 1)
        Set hybrot1 = rot1.HybridShape
        hybrot1.RotationType = 0
        hybrot1.Axis = Ref3
        FP.InWorkObject = hybrot1
        FP.Update


      Set sel1=CATIA.ActiveDocument.Selection
        Set cClashes =
CATIA.ActiveDocument.Product.GetTechnologicalObject("Clashes")
        Set oClash = cClashes.AddFromSel
        oClash.ComputationType = 0
        oClash.InterferenceType = 1
        oClash.Clearance=5
        oClash.Compute
        Set cConflicts = oClash.Conflicts
        If cConflicts.count > 0 then
          Set oConflict = cConflicts.Item(1)
                CV1=oConflict.Value
        End If



        Set FP=PP.Item(i).ReferenceProduct.Parent.Part
        set l= FP.HybridBodies.item("Lines")
        Set lin1 =  l.HybridShapes.item(2)
        Set SFy = FP.ShapeFactory
        Set Ref3 = FP.CreateReferenceFromObject(lin1)
        Set rot1 = SFy.AddNewRotate2(Ref3, -2)
        Set hybrot1 = rot1.HybridShape
        hybrot1.RotationType = 0
        hybrot1.Axis = Ref3
        FP.InWorkObject = hybrot1
        FP.Update

        Set sel1=CATIA.ActiveDocument.Selection
```

```
          Set cClashes =
CATIA.ActiveDocument.Product.GetTechnologicalObject("Clashes")
          Set oClash = cClashes.AddFromSel
          oClash.ComputationType = 0
          oClash.InterferenceType = 1
          oClash.Clearance=5
          oClash.Compute
          Set cConflicts = oClash.Conflicts
          If cConflicts.count > 0 then
            Set oConflict = cConflicts.Item(1)
                  CV2=oConflict.Value
          End If

          If CV1<0 and CV2<0 then
           stb(i,j)=1
       End If

          'Msgbox "j= " &j &"i= " &i &"CV1= " &CV1 &"CV2= " &CV2

          Set FP=PP.Item(i).ReferenceProduct.Parent.Part
          set l= FP.HybridBodies.item("Lines")
          Set lin1 =  l.HybridShapes.item(2)
          Set SFy = FP.ShapeFactory
          Set Ref3 = FP.CreateReferenceFromObject(lin1)
          Set rot1 = SFy.AddNewRotate2(Ref3, 1)
          Set hybrot1 = rot1.HybridShape
          hybrot1.RotationType = 0
          hybrot1.Axis = Ref3
          FP.InWorkObject = hybrot1
          FP.Update



          'Msgbox "its back"

    End If


     For hp=1 to nop
          set par=PPss.Item(i)
          sel.add par
     Next
     set vis= sel.VisProperties
     vis.SetShow 0
     sel.clear
Next
Next
```

```
    For hp=1 to nop
        set par=PPss.Item(hp)
            sel.add par
    Next
    set vis= sel.VisProperties
    vis.SetShow 0
    sel.clear

msgbox "Its working Fine"

Set xlApp = CreateObject("EXCEL.Application")
xlApp.Visible = True
xlApp.Workbooks.Add
for i=1 to nop
for j=1 to nop
  xlApp.Sheets.item(1).Cells(i,j).Value=stb(i,j)
Next
Next


End Sub
```

PROGRAM: 5

To generate permanent stability (due to part mating features) matrix for a product opened in CATIA V5 environment.

```
Dim cog(2)

Sub CATMain()

Dim pa(50,8)  'Part Attributes (COG, Volume & Density )
Dim fmxm(50,50)
Dim lm(50,50)
Dim stb(50,50)
Dim gfm(6,50,50)
Set Prod=CATIA.ActiveDocument.Product
Set PPss=Prod.Products
nop1= PPss.count
Set sel=CATIA.ActiveDocument.Selection
temp=0
For i=1 to nop1
  pnam=PPss.Item(i).Name
  If (Left(UCase(pnam), 3) = "PIN" OR Left(UCase(pnam), 3) = "NUT" OR
Left(UCase(pnam), 4) = "BOLT" OR Left(UCase(pnam), 5) = "SCREW" OR
Left(UCase(pnam), 5) = "RIVET") Then
        set par=PPss.Item(i)
        sel.add par
        temp=temp+1
  End If
Next

set vis= sel.VisProperties
vis.SetShow 1
sel.clear
nop=nop1-temp

fpnam=Inputbox("Enter file name(Full Name including Path) ", "Location of the file
and File name" ,"C:\Users\hp\Desktop\Imp Programs\7P.xls")
Set XL = CreateObject("Excel.Application")
Set objWorkbook = XL.Workbooks.Open(fpnam)
'XL.Application.Visible = True

For i=1 to nop
  For k=1 to nop
    lm(i,k)=XL.Sheets.item(2).Cells(i,k).Value
    gfm(1,i,k)=XL.Sheets.item(4).Cells(i,k).Value
    gfm(2,i,k)=XL.Sheets.item(5).Cells(i,k).Value
    gfm(3,i,k)=XL.Sheets.item(6).Cells(i,k).Value
    gfm(4,i,k)=XL.Sheets.item(7).Cells(i,k).Value
    gfm(5,i,k)=XL.Sheets.item(8).Cells(i,k).Value
    gfm(6,i,k)=XL.Sheets.item(9).Cells(i,k).Value
  Next
```

```
Next

Set Doc=CATIA.ActiveDocument.Product
Set PP = Doc.Products


For i=1 to nop

For j=1 to nop
    stb (i,j) = 0
    par= gfm(1,i,j)+gfm(2,i,j)+gfm(3,i,j)+gfm(4,i,j)+gfm(5,i,j)+gfm(6,i,j)
    aa=0
    bb=0
    If lm(j,i)=1 and par=1 Then
        Set FP=PP.Item(i).ReferenceProduct.Parent.Part
        Set Shp = FP.Bodies.Item(1).Shapes
        For z=1 to shp.Count
                If (Left(UCase(Shp.item(z).Name), 6) = "THREAD" ) Then
                aa=1
                    z= shp.Count
                End If
        Next
        Set FP=PP.Item(j).ReferenceProduct.Parent.Part
        Set Shp = FP.Bodies.Item(1).Shapes
        For z=1 to shp.Count
                If (Left(UCase(Shp.item(z).Name), 6) = "THREAD" ) Then
                bb=1
                    z= shp.Count
                End If
        Next
    End If
    If (aa=1 and bb=1) Then
      stb(i,j)=2
    End If
Next
Next
msgbox "Its working Fine"
Set xlApp = CreateObject("EXCEL.Application")
xlApp.Visible = True
xlApp.Workbooks.Add
for i=1 to nop
for j=1 to nop
  xlApp.Sheets.item(1).Cells(i,j).Value=stb(i,j)
Next
Next

End Sub
```

PROGRAM: 6

To generate permanent stability (due to part mating features) matrix for a product opened in CATIA V5 environment.

```
Sub CATMain()

Dim movedist(11)
Dim pa(50,8)   'Part Attributes (COG, Volume & Density )
Dim fmyp(50,50)
Dim stb(50,50)
Dim tempa(5)
Set Prod=CATIA.ActiveDocument.Product
Set PPss=Prod.Products
nop1= PPss.count
Set sel=CATIA.ActiveDocument.Selection
temp=0

For i=1 to nop1
   pnam=PPss.Item(i).Name
   If (Left(UCase(pnam), 3) = "PIN" OR Left(UCase(pnam), 3) = "NUT" OR
Left(UCase(pnam), 4) = "BOLT" OR Left(UCase(pnam), 5) = "SCREW" OR
Left(UCase(pnam), 5) = "RIVET") Then
            'set par=PPss.Item(i)
            'sel.add par
            temp=temp+1
   End If
Next

'set vis= sel.VisProperties
'vis.SetShow 1
'sel.clear
nop=nop1-temp

for i=1 to nop
for j=1 to nop

stb(i,j)=0

Next
Next



For i=nop+1 to nop1
     For j=nop+1 to nop1
        set par=PPss.Item(j)
            sel.add par
     Next
     set vis= sel.VisProperties
     vis.SetShow 1
```

```
    sel.clear
    sel.add PPss.Item(i)
    set vis= sel.VisProperties
    vis.SetShow 0
    sel.clear

        Set sel1=CATIA.ActiveDocument.Selection
        Set cClashes =
CATIA.ActiveDocument.Product.GetTechnologicalObject("Clashes")
        Set oClash = cClashes.AddFromSel
        oClash.ComputationType = 0
        oClash.InterferenceType = 1
        oClash.Clearance=5
        oClash.Compute
        Set cConflicts = oClash.Conflicts

        aa=1
        For ll=1 to cConflicts.count
                Set oConflict = cConflicts.Item(ll)
                CV=oConflict.Value

                If CV < 0.01 and CV > -0.01 then

                        If(PPss.item(i).name=oConflict.FirstProduct.Name) Then
                  For jj=1 to nop
                            If(PPss.item(jj).name=oConflict.SecondProduct.Name) Then
                                tempa(aa)=jj
                                aa=aa+1
                                jj=nop
                            End If
                        Next
              ' msgbox PPss.item(i).name &"-" &PPss.item(tempa(aa)).name
                   End If

                        If(PPss.item(i).name=oConflict.SecondProduct.Name) Then
                  For jj=1 to nop
                            If(PPss.item(jj).name=oConflict.FirstProduct.Name) Then
                                tempa(aa)=jj
                                aa=aa+1
                                jj=nop
                            End If
                        Next
                                'msgbox PPss.item(i).name &"-"
&PPss.item(tempa(aa)).name
                   End If
                End If
        Next
        If tempa(1)>=1 and tempa(2)>=1 then
                stb(tempa(1),tempa(2))=3
```

```
                    stb(tempa(2),tempa(1))=3
          End If

Next

msgbox "Its working Fine"


Set xlApp = CreateObject("EXCEL.Application")
xlApp.Visible = True
xlApp.Workbooks.Add

for i=1 to nop
for j=1 to nop

xlApp.Sheets.item(1).Cells(i,j).Value=stb(i,j)

Next
Next


End Sub
```

PROGRAM: 7

Part Concatenation method to generate set of all feasible assembly sequences along with assembly directional changes, Tool/gripper changes and assembly energy.

```
'Give xl file along with path.. and run macro

Sub CATMain()

'Maximum numbre of Parts 50
'Maximum sequences 50000
Dim asq(12,200000,12)
Dim alen(15)
Dim nop               'Total number of parts
Dim pa(50,8)  'Part Attributes (COG, Volume & Density )
Dim lm(50,50)'Liaison Matrix
Dim st(50,50)  'Stability Matrix
Dim fm(6,50,50)        'Interference free matrix along all six direction
Dim mfm(50,50,50)  'Mechanical feasibilty matrix
Dim grpp(50,50) 'Gripper matrix
Dim grp(5000,50)

Dim asd(3000,25) 'Assembly Direction matrix
Dim tdr(6)
Dim lbl



fpnam=Inputbox("Enter file name(Full Name including Path) ", "Location of the file
and File name" ,"C:\Users\hp\Desktop\Imp Programs\7P.xls")
Set XL = CreateObject("Excel.Application")
Set objWorkbook = XL.Workbooks.Open(fpnam)
'XL.Application.Visible = True

nop=XL.Sheets.item(1).Cells(1,2).Value

'msgbox nop
'exit sub

For i=1 to nop
  For k=1 to 8
    pa(i,k)=XL.Sheets.item(1).Cells(2+i,k+1).Value
  Next
  For j=1 to nop
    lm(i,j)=XL.Sheets.item(2).Cells(i,j).Value
    st(i,j)=XL.Sheets.item(3).Cells(i,j).Value
    grpp(i,j)=XL.Sheets.item(1).Cells(2+i,10+j).Value
    For k=1 to 6
        fm(k,i,j)=XL.Sheets.item(3+k).Cells(i,j).Value
    Next
```

```
      For ki=1 to nop
            mfm(i,j,ki)=XL.Sheets.item(10).Cells(i+(nop*(ki-1)),j).Value
      Next
   Next
Next

XL.ActiveWorkBook.Close
XL.Quit


'msgbox mfm(1,2,4)

''''''''''''''''''Generation of 2set assembly matrix''''''''''''''''
alen(2)=0
for i=1 to nop
  for j=1 to nop
            If lm(i,j)=1 And st(j,i)>0  then
             alen(2)= alen(2)+1
         asq(2,alen(2),1)=i  : asq(2,alen(2),2)=j
        End if
   next
next


''''''''''''''''''End of  2set assembly matrix generation ''''''''''''''''

''''''''''''''Number of parts
n=nop


'''''''''''''Generation of higher level assembly sets'''''''

For i=2 to n-1

 alen(i+1)=0
 For j=1 to alen(i)

  For k=1 to n

        temp=0
            for l=1 to i
         if asq(i,j,l)=k then
          temp=temp+1
         end if
        next

            if temp=0 then
            'check for liaison
```

```
            sum1=0
            for l=1 to i
               sum1=sum1+lm(asq(i,j,l),k)
            Next

'check for stability

            sum2=0
            for l=1 to i
               sum2=sum2+st(k,asq(i,j,l))
            Next

'check for geometric feasibility
            sum3=0
            for pp=1 to 6
               sum4=0
               for l=1 to i
                    sum4=sum4+fm(pp,k,asq(i,j,l))
               Next
               If sum4=i then
                    sum3=1
                    pp=6
       End If
            Next

'check for mechanical feasibility
            sum5=0
            For l=1 to i
               If st(k,asq(i,j,l))>=2 Then
                        For chi=1 to i
                          If mfm(k,asq(i,j,l),asq(i,j,chi))>0 then
                           sum5=1
                                    'msgbox k&asq(i,j,l)&chi&asq(i,j,chi)
                           chi=i
                           l=i
                          End If
                        Next
               End If
            Next

'Append Part
            if (sum1>0) And (sum2>=1) And (sum3>0) And (sum5<1)  then
               alen(i+1)=alen(i+1)+1

               for l=1 to i
              asq(i+1,alen(i+1),l)=asq(i,j,l)
   next
           asq(i+1,alen(i+1),i+1)=k
```

```
                       End If

      end if
  Next

 Next
 'msgbox i&"-"&alen(i+1)
Next

''''''''''Énd of higher level assembly sets generation''''''

msgbox alen(n)
'msgbox "its working fine"
'exit sub

''''''''''''Number of parts
n=nop


''''''''''''Direction matrix calculation

For i=1 to alen(n)
   For j=n to 2 Step -1
      For dir=1 to 6
        sum4=0
        tdr(dir)=0
            For l=1 to j-1
          sum4=sum4+fm(dir,asq(n,i,j),asq(n,i,l))
        Next
        If sum4 = j-1 then
            tdr(dir)=1
            End If
      Next
      If j=n then
        For lbl=1 to 6
          If tdr(lbl)=1 then
              asd(i,j)=lbl
              lbl=6
          End If
        Next
      End If
      If j<n then
        For lbl=1 to 6
          If tdr(lbl)=1  And lbl=asd(i,j+1) Then
              asd(i,j)=lbl
              lbl=6
          End If
        Next
        If Not asd(i,j)=asd(i,j+1) then
```

```
        For lbl=1 to 6
          If tdr(lbl)=1 Then
             asd(i,j)=lbl
             lbl=6
          End If
        Next
         End If
    End If
  Next
  asd(i,1)=asd(i,2)
Next


"""""""Verification for sequenciality
For i=1 to alen(n)
  For j=3 to n
   If Not asd(i,j-1)= asd(i,j)   Then
        sum4=0
        for l=1 to j-1
         sum4=sum4+fm(asd(i,j-1),asq(n,i,j),asq(n,i,l))
        Next
        If sum4=j-1 Then
          asd(i,j)=asd(i,j-1)
        End If
   End If
  Next
Next


"""""""No. of directional changes
Dim noch(2000)

For i=1 to alen(n)
  chg=0
  For j=1 to n-1
   diff=asd(i,j)-asd(i,j+1)
   diff=Abs(diff)
   tempd=1
   If diff=0 then
        tempd=0
   End If
   chg=chg+tempd
 Next
 noch(i)=chg
Next


"""""""Gripper matrix calculation
```

```
For i=1 to alen(n)
  For j=1 to n
    For l=1 to n
        If asq(n,i,j)=l then
       If j=1 then
         grp(i,j)=grpp(l,1)
           l=n
          End If
          If j>1 then
            For k=1 to n
             asdf=1
             If grp(i,j-1)=grpp(l,k) Then
            grp(i,j)=grpp(l,k)
                 k=n
            asdf=0
            l=n
             End If
         Next
            If asdf=1 then
           grp(i,j)=grpp(l,1)
                l=n
            End If
       End If
      End If
    Next
  Next
Next


'''''''''''''No. of gripper changes
Dim nocg(2000)

For i=1 to alen(n)
  chg=0
  For j=1 to n-1
   diff=grp(i,j)-grp(i,j+1)
   diff=Abs(diff)
   tempd=1
   If diff=0 then
        tempd=0
   End If
   chg=chg+tempd
  Next
  nocg(i)=chg
Next

'''''''''''''Energy estimation
Dim aener(5000)
For i=1 to alen(n)
```

```
aener(i) = 0
  For j=n to 2 step -1
     pp=asq(n,i,j)
     If asd(i,j)=1 then
          ppd=pa(pp,1)
          apd=pa(asq(n,i,1),4)
          For k=2 to j-1
            ap=asq(n,i,k)
            high=pa(ap,4)
                        If apd<high then
                apd=high
            End If
          Next
          distt=abs(apd-ppd)
    End if
    If asd(i,j)=2 then
          ppd=pa(pp,4)
          apd=pa(asq(n,i,1),1)
          For k=2 to j-1
            ap=asq(n,i,k)
            low=pa(ap,1)
                        If apd>low then
                apd=low
            End If
          Next
          distt=abs(ppd-apd)
    End if
     If asd(i,j)=3 then
          ppd=pa(pp,2)
          apd=pa(asq(n,i,1),5)
          For k=2 to j-1
            ap=asq(n,i,k)
            high=pa(ap,5)
                        If apd<high then
                apd=high
            End If
          Next
          distt=abs(apd-ppd)
    End if
    If asd(i,j)=4 then
          ppd=pa(pp,5)
          apd=pa(asq(n,i,1),2)
          For k=2 to j-1
            ap=asq(n,i,k)
            low=pa(ap,2)
                        If apd>low then
                apd=low
            End If
          Next
```

```
          distt=abs(ppd-apd)
    End if
    If asd(i,j)=5 then
          ppd=pa(pp,3)
          apd=pa(asq(n,i,1),6)
          For k=2 to j-1
            ap=asq(n,i,k)
            high=pa(ap,6)
                        If apd<high then
                apd=high
            End If
          Next
          distt=abs(apd-ppd)
    End if
    If asd(i,j)=6 then
          ppd=pa(pp,6)
          apd=pa(asq(n,i,1),3)
          For k=2 to j-1
            ap=asq(n,i,k)
            low=pa(ap,3)
                        If apd>low then
                apd=low
            End If
          Next
          distt=abs(ppd-apd)
    End if
    ener=pa(pp,7)*pa(pp,8)*distt/1E11
     aener(i)=aener(i)+ener
  Next
Next
"""""Open an excel file and add workbook
Set xlApp = CreateObject("EXCEL.Application")
xlApp.Visible = True
xlApp.Workbooks.Add

For i=1 to alen(n)
  For j=1 to n
    xlApp.Sheets.item(1).Cells(i,j).Value=asq(n,i,j)
    xlApp.Sheets.item(1).Cells(i,n+j+1).Value=asd(i,j)
    xlApp.Sheets.item(1).Cells(i,n+n+j+4).Value=grp(i,j)
  Next
  xlApp.Sheets.item(1).Cells(i,(n*2)+3).Value=noch(i)
  xlApp.Sheets.item(1).Cells(i,(n*3)+7).Value=nocg(i)
  xlApp.Sheets.item(1).Cells(i,(n*3)+9).Value=aener(i)
Next
msgbox "Its working fine"
End Sub
```

INDEX