GLOBAL SEQUENCE GENERATION AND MINIMIZING THE TOTAL UTILITY WORK FOR MIXED MODEL ASSEMBLY LINES

AMJAD BAIG
MS-76 (Mechanical Engineering)
(NUST201362454MCEME35113F)

Supervisor

Assistant Prof. Dr. Sajid Ullah Butt

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

JULY, 2015

GLOBAL SEQUENCE GENERATION AND MINIMIZING THE TOTAL UTILITY WORK FOR MIXED MODEL ASSEMBLY LINES

Amjad Baig
NUST201362454MCEME35113F

A thesis submitted in partial fulfillment of the requirements for the degree of MS Mechanical Engineering

Thesis Supervisor

Assistant Professor Dr. Sajid Ullah Butt

Thesis Supervisor's Signature:____

DEPARTMENT OF MECHANICAL ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD
JULY, 2015

Declaration

I certify that this research work titled "*Global Sequence Generation and Minimizing the Total Utility Work for Mixed Model Assembly Lines*" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student
Amjad Baig
NUST201362454MCEME35113F

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student
Amjad Baig
NUST201362454MCEME35113F

Signature of Supervisor

Copyright Statement

Acknowledgement

First and foremost, I would like to thank my Almighty Allah for all His blessings and benevolence which helped me complete this incredible task in terms of integrity and completeness.

Moreover, I would like to express my special appreciation and gratitude to my advisor and co-advisor, Assistant Professor Dr. Sajid Ullah Butt and Associate Professor Dr. Aamer Ahmed Baqai for their extended support throughout my thesis work. Their patience and knowledge helped pave way when I was stuck and allowed to me to work in a manner which was both productive and un-stressful. Their advices have hence been priceless.

I would also like to thank Assistant Professor Dr. Imran Akhtar, and Assistant Professor Dr. Hasan Aftab Saeed, for serving as my Guidance and Evaluation committee even at hardship. I am really obliged for letting my defense be an enjoyable moment, and for their brilliant comments and suggestions.

Also, I am heavily indebted to the knowledge transfer related to genetic algorithms (GAs) by Lecturer Uzair Khaleeq uz Zaman. It surely helped clarify a lot of issues pertaining to the creation of an algorithm for GA.

Conclusively, I would like to thank the entire honorable faculty of Mechanical Department of the College of Electrical and Mechanical Engineering, whose professional approach and vision groomed me as a sound person both technically and morally.

Amjad Baig
May, 2015

To my beloved parents and family….

Abstract

The recent developments in the field of industrial automation, especially related to mass customization, have increased the demand for mixed model assembly lines which involve customized production following a particular ‘product mix’, i.e., number of models of a base product are jointly processed on a single line, in an increased quantity, quality and conducive environment. Various factors relate to the optimal operating sequence of the operations such as total setup cost, smooth consumption of parts’ usage, total utility work, etc. In order to provide corrective measures in each case, mixed model assembly lines require the services of evolutionary algorithms.

Genetic Algorithm (binary encoding/decoding, two point crossover and uniform mutation) has been used in this study to address a global problem, i.e., total utility work, by working on associated local problems of manufacturing facility such as part assembly, quality control and supporting staff activities. A methodology has been developed to test and analyze the impact of local problems on the concerned global objectives and defense industry-oriented problems were presented to test the algorithm in real world conditions. The results were critically examined and respective improvement measures were stated along with graphical interpretations.

Key Words: *Mixed Model Assembly Line Sequencing, Genetic Algorithms, Total Utility Work, Global Sequence Generation*

Contents

List of Tables

List of Figures

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
AI	Artificial Intelligence
AL	Assembly Line
ALB	Assembly Line Balancing
ALBP	Assembly Line Balancing Problem
ALSP	Assembly Line Sequencing Problem
COP	Combinatorial Optimization Problems
EC	Evolutionary Computation
EA	Evolutionary Algorithm
EGCM	Extended Goal Chasing Method
FALB	Flexible Assembly Line Balancing
GALB	General Assembly Line Balancing
GA	Genetic Algorithm
GCM – I	Goal Chasing Method – I
GCM – II	Goal Chasing Method – II
HH	Hyundai's Heuristic Procedure
ICSP	Industrial Car Sequencing Problem
JIT	Just-in-Time
LS	Linear Scaling
MILP	Mixed Integer Linear Programming
MMAL	Mixed Model Assembly Line
MMALS	Mixed Model Assembly Line Sequencing
MMALS-P	Mixed Model Assembly Line Sequencing-Problem
MPS	Minimum Part Set
MSH – I	Miltenburg and Sinnamon's Heuristic – I
MSH – II	Miltenburg and Sinnamon's Heuristic – II
NP-hard	Non-deterministic Polynomial-time hard
PSO	Particle Swarm Optimization
RS	Ranking Selection

RWS	Roulette Wheel Selection
SA	Simulated Annealing
SALB-P	Single Model Assembly Line Balancing-Problem
SP	Selective Pressure
STS	Sigma Truncation Scaling
SWO	Small World Optimization
TS	Tournament Selection
TUW	Total Utility Work
UW	Utility Work

CHAPTER 1: INTRODUCTION

Earlier 20th century is marked as dawn of industrial revolution, when Henry Ford in Highland Park, Michigan (Cochrane, 1999), presented his famous model T. He brought down the cycle time from twelve laborious hours to less than two hours (Tidd & Bessant, 2009). It was all possible due to the concept of a moving assembly line, which later on played an important role in the economic uplift of the industrial units. The concept of mass production and industrialization allowed manufacturing productivity to rise high and consequently prices of the products dropped down. But with the passage of time the large variations in customer demands and market fluctuations prompted the major changes in process flows.

As a result of above scenario, there was a prominent shift from 'mass production' to 'mass customization'. Thus the manufacturers tried to meet the diversified customer demands as per market fluctuations on the basis of a certain base design (Hu et. al 2011). This transition from 'mass production' to 'mass customization' demanded the scientific decomposition of work process in to a system, so that high quantity and customized products can be produced. In this concept, work load is distributed between different stations in an ordered sequence satisfying the precedence relationship (Pallavi et.al., 2012). This really increases the line efficiency by maximizing the ratio between cost and industry throughput.

Assembly lines are basically flow oriented production system where the work stations (known as productive units) performing different operations are aligned in serial or parallel manner. The assembly line can be designed on any of the three main categories; single model, multi model or mixed model assembly lines (Scholl, 1999). In first category, a single product is assembled with no variants on a single assembly line. All operations are identical on all stations which produces the identical products. In second category, different products that require different assembly processes and 'station setup' needs to be changed for the start of assembly of another product. As a result assembly is carried out in batches to reduce the changeover cost (Torenli, 2009). In third category, assembly of variants of a single/base product is carried out, which is referred as customized production following a particular 'product mix', which means variant models of the base product are jointly processed/assembled on a line, in an increased quantity, quality and productive environment. The base product is customizable by the (de-

)selection of optional features out of a certain set of options (Boysen et al., 2007), and the variants of base product then undergo similar processes with different operation times for different models. So the Mixed Model Assembly Lines (MMAL) are more flexible and cost effective in high volume, mass production systems (Duplaga & Bragg, 1998).

Designing MMALs invites many issues and their effective utilization involves the solution of couple of problems in a sequential manner; one, designing and balancing the MMAL and two, to find the viable (optimized) production sequence for different models (R. Tavakkoli et. al 2007). In order to calculate the launch sequence of the models, MMALs have broadly worked on either, the leveling of workloads on each work station on the assembly line, or keeping, parts usage rate constant on the assembly line. Both of the mentioned objectives have been addressed either individually or combined in literature.

The applications of MMALs today range from consumer electronics to automotive industries wherein diversified demands from the market are catered for by not only cutting off large end product inventories but also reducing significant setup times (J.cano et.al 2008). Hence, mass customization is being processed on a mass production cost and MMALs are producing high quantity standardized / customized commodities in low to medium volume production setups (Pallavi et. al 2012).

To find out the efficiency of different production sequences, researchers have worked on further sub-objectives such as minimizing total utility work, keeping a constant rate of part usage, minimizing total setup cost, minimizing the risk of stopping a conveyor, minimizing the overall line length, leveling workloads, etc (S. Tunali et. al 2010). Each of these sub-objectives when combined with one or more of the factors such as resources (quality control and support staff), assembly line layout, efficiency and cost, invite the concept of Multi-objective problem (MOP). MOPs have been addressed in literature by heuristics (e.g., Kotani et.al, 2004), metaheuristics (e.g., Tunali, 2010) and mixed integer linear programming (MILP) models (e.g., Giard and Jeunet, 2010). MILP, Branch & Bound (B&B) Algorithm and Constraint Program, are part of ‘Exact Approaches’ while ‘Heuristic Methods’ include Greedy Approach, Local Search Approach, Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm

Optimization (PSO), commonly referred to as the evolutionary algorithms (EAs) (Durmusoglu, 2011).

The EAs follow an iterative and stochastic process and copies natural evolution by working on set of individuals which is called 'population'. Every individual reflects a potential solution to the problem at hand, which is being solved. This solution is a result which is obtained with the help of an encoding/decoding mechanism (Sivanandam & Deepa, 2008). As mentioned earlier the researchers have worked with number of search techniques and Enumerative Techniques (Branch & Bound and Back tracking), to solve Non-deterministic Polynomial-time hard (NP-hard) problems, but the GAs have been recognized as the most extended group of methods which works on successive generations of better and better individuals by employing very simple techniques (operators) such as crossover and mutation. Each individual is assigned a fitness value, called chromosome, with the help of fitness function and finally an objective function is optimized. Following figure shows the search techniques used to-date:

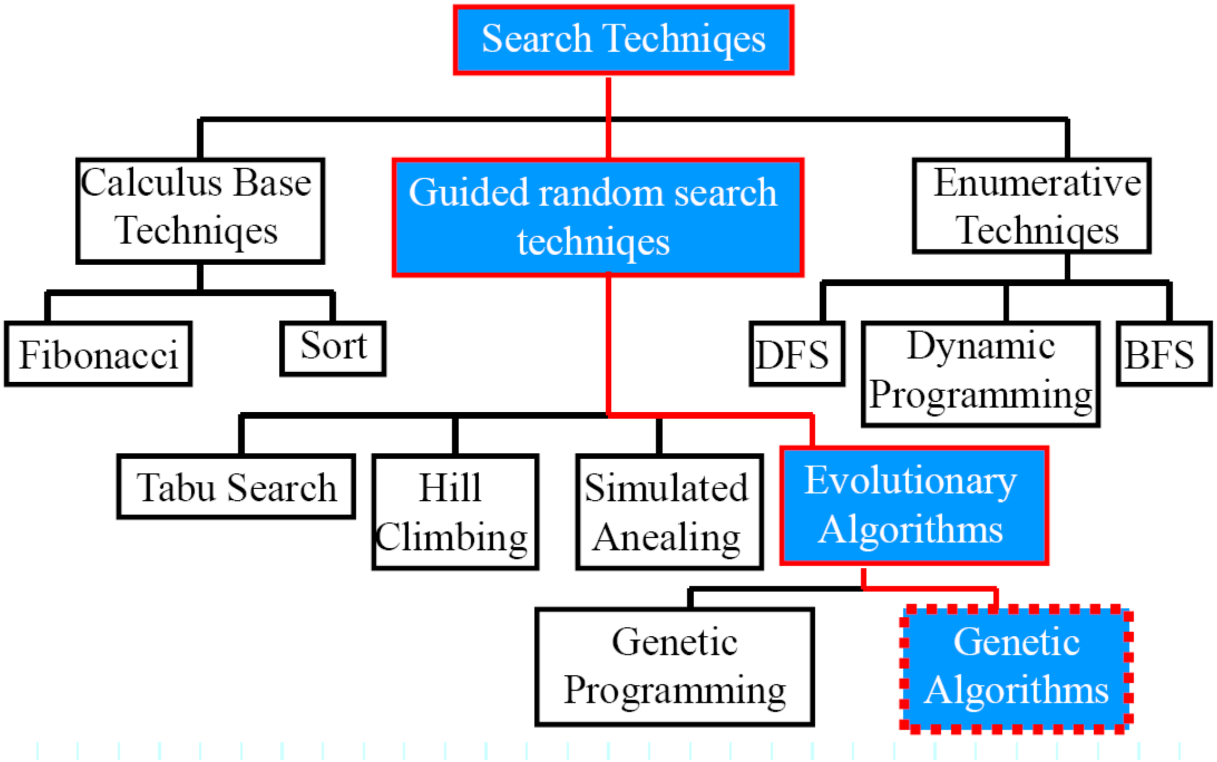


Figure 1. Search Techniques

This study has focused on the mixed model assembly line, which is paced and asynchronous, with the aim of minimizing total utility work and generating a global sequencing using Genetic Algorithm. As a case study, a Defense Industry problem has been considered to test the algorithm by implementation of the obtained results on ground. Finally the results have been critically examined and summarized. The current study consists of following sections:

1. Assembly Line Design & Evolutionary Methods
2. Balancing & Sequencing Related to Assembly Lines
3. Conventional and Evolutionary Optimization Techniques
4. Proposed Approach
5. Results, Analysis and Future work

CHAPTER 2: ASSEMBLY LINE DESIGN & EVOLUTIONARY METHODS

Literature survey shows that a lot of research has been carried out on balancing and sequencing of mixed model assembly lines (MMAL) and more emphasis has been given on balancing rather than sequencing of MMALs. The literature review has been divided into following areas to discuss the Concept of assembly line design, its configurations and issues related to assembly lines and finally the heuristics involved in finding the solutions of these issues in detail:

1. Concept of assembly line design
2. Configuration of assembly lines
3. Definition / notations used in assembly lines’ design

2.1 CONCEPT OF ASSEMBLY LINE DESIGN

Assembly lines (ALs) consist of a series of stations, which are connected with the help of conveyor and it can perform the set of tasks on the products which are passing through them. These assembly lines have a complex structure due to variety of components (e.g. material handling facility, line efficiency, reliability, cost, imbalance, stations space, tooling, etc.).

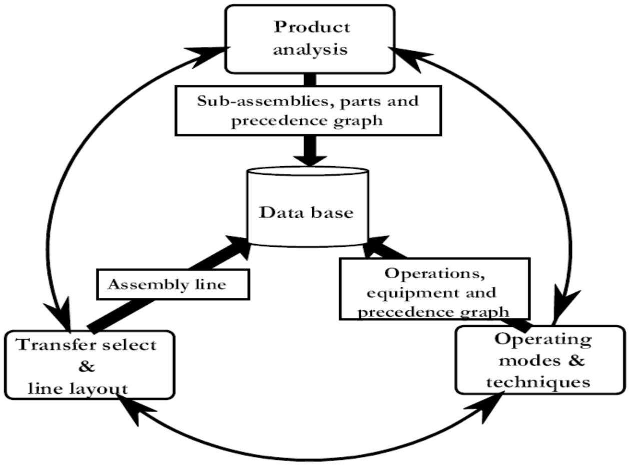


Figure 2. Methodology and Information Flow Of The ALD (F. Pellichero et. al 1999)

The assembly lines can be categorized as synchronous and asynchronous. In synchronous assembly lines there is a coordinated work movement and all work pieces move to the next workstation simultaneously. No buffer is needed here and the number of work pieces remains constant. While in asynchronous assembly lines we don't have coordinated movement. The operator or machine starts the operation as soon as a work piece becomes available. When an operation completes, the work piece moves to the next station subject to the availability of space. Then there are paced and un paced assembly lines. The paced lines are quite simple. In figure 3 the number of workstations is four. These have been connected with help of a conveyor and these stations can performs one or more tasks, on the partially finished product which are in front of it. Robots and trained workers complete these tasks. The conveyor on AL moves, by the interval of cycle time (C). This ensures the positioning each product at the next workstation in the line. Finished product leaves the assembly line at last work station. Now the precedence constraints are involved among different tasks / operations.

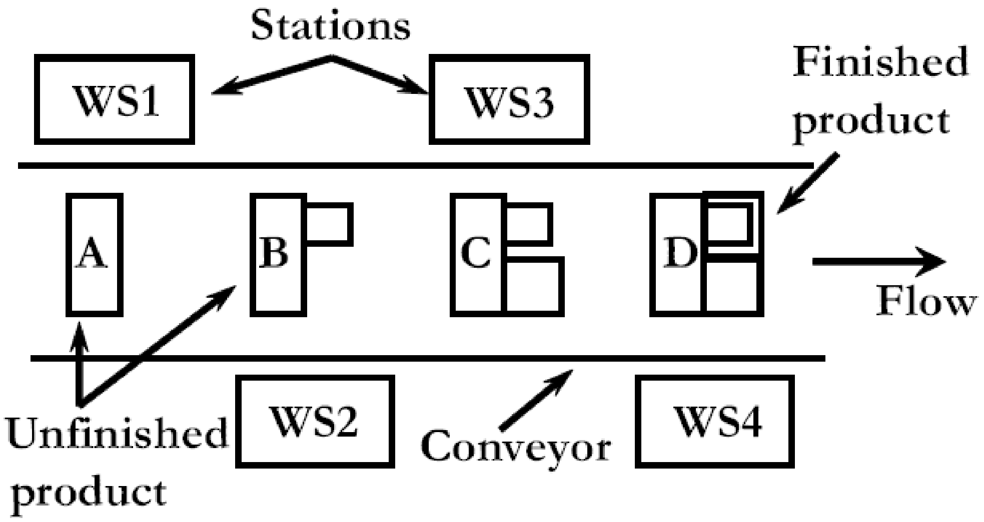


Figure 3. Assembly Line Concept

2.2 CONFIGURATIONS OF ASSEMBLY LINES

Assembly lines can be designed on several configurations, like serial, parallel, and U shaped assembly lines etc., which have been discussed in details in succeeding paragraphs.

2.2.1 Serial Assembly Lines

In this category single workstations are designed along a straight line which has some conveyer system as shown in Figure 4. Every workstation act as simple system of a complex unit as it performs tasks on the partially finished product. (I. Baybar 1986).

Figure 4. Serial Assembly Line Configuration

2.2.2 U-shaped Assembly Lines

'Just in time' principle of production, suggest that U shaped assembly lines have several advantages over the over the traditional configuration as shown in Figure 5. The workers are employed in the center of U and they can easily monitor the progress of each other and can coordinate easily whenever it is required (J. Wijngaard et.al 1994). As a result, workers highly motivated, which results in high quality of the products and flexibility is also increased.

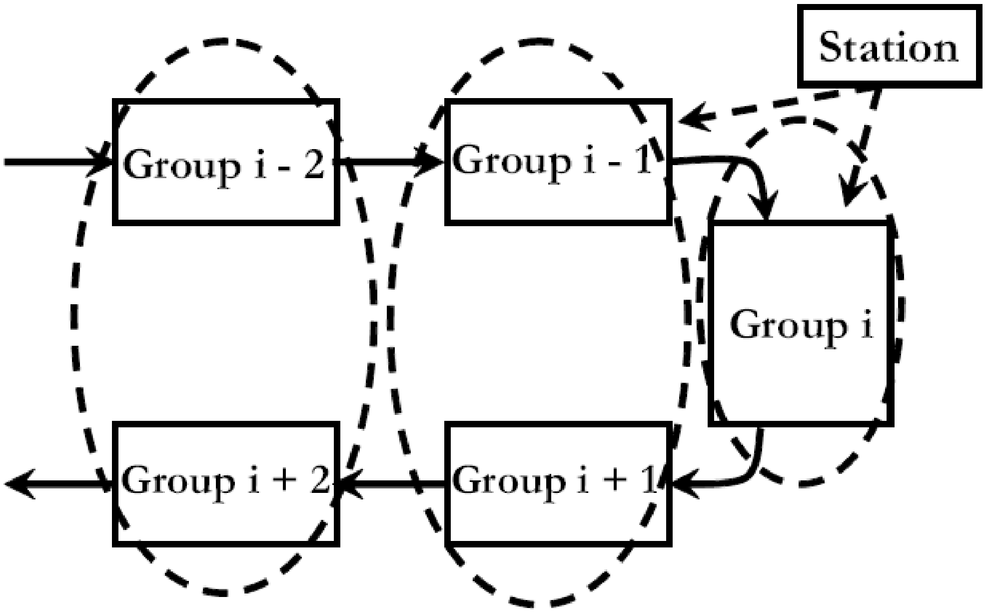


Figure 5. U-shaped line configuration

2.2.3 Parallel Assembly Lines

In case of high demand, entire assembly line can be copied like a parallel system. It definitely has several advantages including the shortening of the overall assembly line. But in this case more tooling and equipment is required. In case of failure happens at some workstation, the other lines can keep on working. Figure 6illustrate the working of parallel lines.

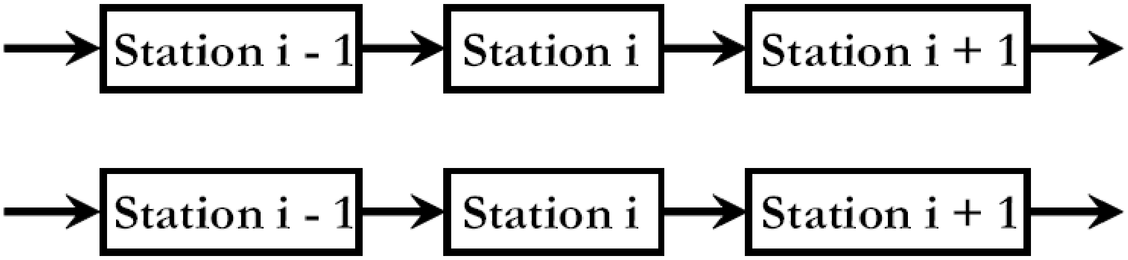


Figure 6. Parallel assembly lines

2.2.4 Parallel Stations

With high production rates it happens sometime that the largest task time exceeds a specified cycle time. As a common measure workstations are created as parallel or serial posts. So as result more number of workers can do similar jobs. This procedure minimizes average value of the ‘task duration’ which is proportional to the number of workers on other workstations as shown in Figure 7.

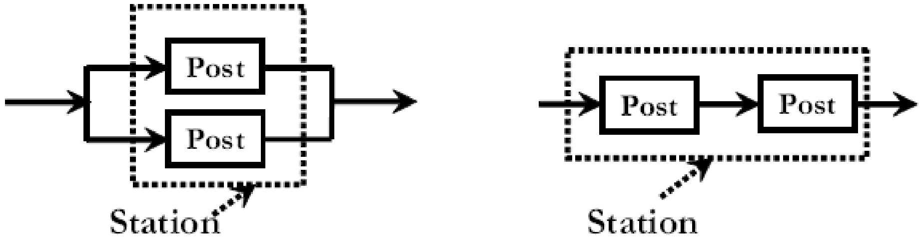


Figure 7. Parallel and serial stations

2.3 DEFINITIONS/NOTATIONS USED IN ASSEMBLY LINES' DESIGN (Rekiek et al. 2006)

2.3.1 Assembly

In this process various parts are fitted together to produce a finished product. These parts are further segregated into components and sub-assemblies.

2.3.2 Assembly Line

It is a flow oriented production system and consists of a number of workstations. The jobs are launched on this system consecutively. These jobs are moved from station to station.

2.3.3 Task

Assembly line process consists of the portion of the total work content. This work content is called Task. The time to perform a task is called the task process time. These tasks can not be further divided or split in to smaller work elements.

2.3.4 Precedence Constraints

The tasks are performed in some order, which are the precedence constraints. The Ordering of tasks can be explained with the help of a precedence graph (E.D. Sacerdot 1977). Nodes in figure 8 represent the Tasks the directed arrows shows the precedence relationships between these tasks. The Figure 8 also shows that task 5 is preceded by tasks 3 and 4.

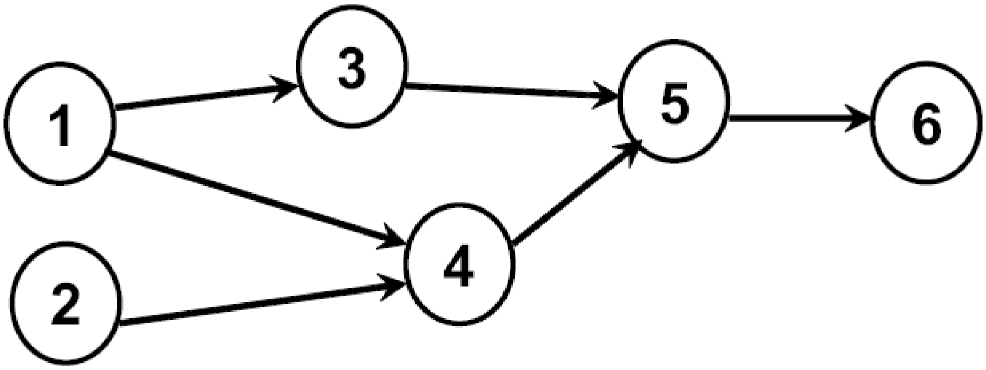


Figure 8. Precedence Graph

2.3.5 Cycle Time (C)

It is the time between the exits of two products, consecutively, from the assembly line. Each station has maximum amount of work process and some required cycle time. Normally the planning department is looking for this cycle time. And the effective cycle time is the actual cycle time on which the assembly line will operate.

2.3.6 Capacity Supply (CS)

It is CS = nC i.e. cycle capacity supply is equal to N times cycle time. It can also be defined as the total time which is available to assemble a product.

Makespan

Maximum completion time which is required to process all the tasks of a prescribed set of products is known as makespan.

2.3.7 Maximum Peak Time

It is related to multi model assembly lines. So there a variant process times. It is never exceeded by a variant process time at any workstation.

2.3.8 Imbalance

The process time of a workstation depends on the variation in the products. So the imbalance is calculated by the difference between total duration of tasks and cycle time on a workstation.

2.3.9 Work Content

The sum of process time (Ti) of all tasks is called:

$$WC = \sum_{i=1..n} T_i$$

2.3.10 Station Time

Total process time at a station is called station time. And when all the station times are added it gives the total assembly time.

2.3.11 Line Efficiency (E)

It is defined as E = WC/CS i.e. efficiency is equal to work content divided by capacity supply. Balance delay time is defined as BD = CS − WC.

2.3.12 Station Idle Time

The difference of cycle time and the workstation time gives station idle time. Addition of the workstations idle time can be calculated by I = CS−WC.

2.3.13 Smoothness Index (SX)

There is distribution of work among different work stations and its standard deviation is calculated by smoothness index and is calculated by the following formula:

$$SX = \sqrt{\sum_{i=1}^{i=n} (C - ST(i))^2}$$

In early nineteenth century, Henry Ford introduced his Model T (Cochrane, 1999) which is a good example of early time assembly line of a single product. It provided a best comprehension towards product modularity (Kamal U., & Lastra, 2011). As the single product passes through number of productive units, the flow of assembly line required balancing of work load between different stations. So people started working on assembly line balancing problem (ALBP). In mid 50's, Salveson (1955) suggested a linear programming solution (Kriengkorakot & Pianthong, 2007). And in 1960's Bowman (1960), Supnik and Solinger (1960), and Hu (1961) worked in the same area (Erel & Sarin, 1998). Mass production was central idea of these early researchers. Focusing on the Single models they tried to familiarize with the assembly line concept and its implications in relation to the then industry. Afterwards, a lot of research took place focusing single model assembly lines and different algorithms were proposed to give the solution to these balancing problems.

In today’s era the customer needs diversity in the product range, with short product life and lead times. But the mass production of a single product works on the concept of labor specialization and leveling the work load between work stations. As a result it reduces assembly costs but low work satisfaction comes out due to the employee turnover. So the solution to this problem is ‘Mixed model assembly lines’, as these focus on low to medium volume production by addressing present era customer demands; which is a step towards mass customization instead of mass production on single model assembly line (Kamal U., & Lastra, 2011).

Balancing and Sequencing related to mixed model assembly lines were the first approaches used to handle product variety using a given ‘mix model’ between the work stations (Bukchin et. al 2002). The ‘mix model’ is defined as the number of units produced during specific shift duration. The mixed model line balancing or sequencing procedure is the same as that for single model assembly line, but the tasks are assigned on the basis of shift duration rather than station cycle times.

Thomopoulas (1967) and Macaskill (1972) worked on combined precedence diagrams but a research gap was left by them. These diagrams can only be applied if the method of combining the precedence diagrams was acyclic. The research gap left by Thomopoulas and Macaskill was considered by Ahmadi and Wurgaft (1994) and they tried to divide the model mix into smaller subsets. They came out with successful solution but lacked conviction when large processing times were required at the same workstation which was scheduled after each other. Later on this problem was optimally solved by Erel and Gokcen (1999), according to Bukchin et. al (2002). They came with the idea to push the models in the line in such a way that the starvation and blockage at individual stations can be minimized, which means to optimize mixed model assembly line sequencing problem.

The two domains requires special attention towards the effective utilization of a mixed model assembly line; first, allocation of tasks to work stations and second, different models sequencing on the mix model assembly line (Duplaga and Dragg 1998). The former domain has been answered comprehensively in the above mentioned paragraph and the second domain requires attention now. Toyota’s Just-in-time (JIT) production system using mixed integer programming was described by Monden (1983). It was a good algorithm but its use was restricted to small problems only. Later on three sequencing procedures were developed by Miltenburg (1989), for a product that requires tentatively the same number of parts.

Sumichrast and Russell (1990) carried out work on the Miltenburg's heuristics and mentioned that the third algorithm of Miltenburg, was using heuristic 2 (M-A3H2) and it was the most suited heuristic. This algorithm is used to minimize the average absolute deviation from the uniform production of every finished product Duplaga and Dragg (1998).

The mix model product may require different components and different quantity for its assembly, for example a standard car may be different from its other variant, in terms of certain optional features. Therefore, the component parts usage rates are important in sequencing. So the Toyota Motor Corporation catered for the usage rates, by use of two sequencing heuristics (Monden 1983). In addition to this, Duplaga and Dragg (1998) further worked in this area and developed a heuristic and also included the report of three sequencing procedures which were proposed by Miltenburg and Sinnamon (1989).

As there was a shift from mass production to mass customization, there was also changing in the assembly lines configurations i.e. parallel assembly lines, U-shaped assembly lines, and serial and parallel work stations. The other factors like diversified product ranges, larger problem sizes, varying product structures, many model mixes, etc., gave the idea of shifting the focus from General assembly line balancing (GALB) to Flexible assembly line balancing (FALB) such that a lot many extensions can be dealt in a combined manner (Boysen et. al, 2006). Various approaches of designing and sequencing an mixed model assembly line in presence of high part variety including Liaison graph, and Adjacency matrix has been mentioned by Hu et al. (2011).

CHAPTER 3: BALANCING & SEQUENCING RELATED TO ASSEMBLY LINES AND HEURISTIC METHODS

In the previous section, concept of assembly line design, its configurations and terms related to AL was discussed in detail. The equipment selection, complexity of the operational tasks, operation criteria of the assembly line, constraints, sequencing, work station allocation, cost and reliability of the system, inventory control, buffers allocation, and variety in the product are some of the reasons which curtail the problems associated with AL design. So the selection of assembly line design and its configurations depends on above mentioned factors. Depending upon the selected assembly line design and its configurations, the balancing and sequencing problems vary as well. It is now evident that single model assembly lines have no change in facility layout or tool change etc., whereas mixed model assembly lines have many variations of a base product. Thus, AL balancing and Sequencing have its significance in above scenario. In this study I will briefly explain balancing of assembly line while sequencing will be discussed in detail because my research focus is on sequencing of assembly line.

3.1 BALANCING

Precedence constraints, capacity constraints and zoning constraints etc. are the major design constraints related to assembly line balancing. These have been mentioned by Vilarinho & Simaria (2006). Precedence constraints have also been mentioned by Elfner (2003). He stated that the optimization of cycle time and minimization of idle time, can done through the allocation of tasks to different workstations.

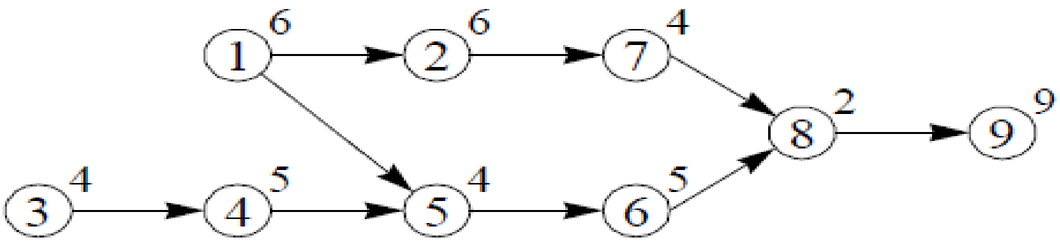


Figure 9. Precedence Diagram 1 – 9 Tasks

The above diagram (Figure 9) can be understood by understanding the basic problem of Assembly Line Balancing (ALB). It involves number of work stations ranging from 1 to m, (k = 1,…., m). These are usually aligned along a conveyor belt or a similar material handling system. It requires certain amount of workers and machines to perform a task (task-j) which requires time t_j. The sum of task times, $t_{sum.}$ gives total work load necessary for assembling a work piece. The above mentioned precedence diagram shows 1 to 9 assembly station having one task each. The task times vary from 2 to 9 units and t_{sum} is equal to 45. The other terms like station load, work content, line efficiency, station time, demand rate, etc. (already mentioned in previous section) are also related to AL balancing problem. Following steps are involved:

- Allocation of tasks to workstations as per precedence constraint:

  $$\max_{i=1,\ldots,N} t_i \leq \max_{j=1,\ldots,K} s_j \leq c \leq 1/D$$

  Where D = demand rate, K = number of work stations and N = number of tasks in the problem.
- Calculate station load / work content
- Calculate station time

  $$s_j = \sum_{i \in S_k} t_i$$

- Calculate feasibility of line balance [if *t(Sk)*< *C* , where C is the cycle time]
- Calculate line efficiency.

After going through the brief but important crux of the AL balancing procedures mentioned above, it is necessary to shift our focus from balancing to AL Sequencing. As mentioned earlier, calculus based techniques were used to solve the balancing and sequencing problems which was a time consuming process. Afterwards it was realized that assembly line problems fall into the class of NP-hard Combinatorial Optimization Problems (COPs) which can be solved best via heuristics (the techniques based on common sense and logic rather than mathematical proofs), for both balancing and sequencing (Grzecha, 2011).

Some of the well known balancing heuristics are:

- Ranked Positional Weight
- Number of Followers
- Number of Immediate Followers
- Number of Predecessors
- Work Element Time.

The next section will focus on Assembly line sequencing and the heuristics associated with it.

3.2 SEQUENCING

Leveling of work load among different work stations is related to ‘balancing of a line’ where the precedence constraints are also kept under consideration. However, if several models of a product are jointly processed / assembled on a line, a need arises to determine the operating schedule/sequence of the operations. This extended version of assembly line variant design is broadly termed as ‘Assembly Line Sequencing’. As per Tunal et. al (2010), various customer demands and intense competition have forced the manufacturing arena to provide quick customized solutions in a cost effective manner. Grzecha, (2011) mentions that factors like varying assembly plans (mixed/batch/single model), differing operating stations (manual/robotic/hybrid), assembly line layouts (straight/parallel/U-shaped) and varying transport mechanisms (conveyor/pallet based) fall in the domain of ALSP. Although, sequencing can be applicable to all the lines technically, but Mixed Model Assembly lines provide exactly the same production environment as required in the statement given by Tunal (2010) above and they don’t require large inventories as in mass production (Rabbani et. al, 2011). The focus of literature survey from now onwards will hence remain on Mixed Model Assembly Line Sequencing (MMALS). Mixed model assembly lines are ‘generally’ used in the cases where; (Rakiek et.al 2006):

- Sub model is a variant of the base model
- Assembly process require the same resources
- There should be more than one model of the products
- Cycle time is normally greater than a minute

- Final product may / may not be delivered in a short time
- Assembly line set up time is short.

MMALs are associated with two goals; First, levelling the workload on every work station on the assembly line, and Second, keep a constant rate of usage of every part which is being used by the assembly line. The initial research was carried out by Yano and Bolat (1989) and Ghosh and Gagnon (1989) while second goal was initially addressed by Monden (1983) when he described Toyota's Just-in-Time (JIT) Production System which was limited to small problems only.

Later on, many sequencing heuristics were devised as per the second goal (which is referred as 'the General Case');

- Toyota's Goal Chasing Algorithms (Monden, 1983)
- Miltenburg Heuristics (Miltenburg, 1989, 1991, 1992)
- Quick and Dirty Sequencing Method (Bulfin, 1992)
- Time Spread Method (Sumichrast et. al, 1992)

3.2.1 Heuristics related to sequencing

The above mentioned six heuristics have been termed as the building blocks of MMALS-P related to the 'General Case'; these building blocks were then designed to smoothen the component part usage in sequencing problems. Duplaga & Bragg, (1998) has briefly explained each heuristic which is given below:

Goal Chasing Method – I (GCM – I)

This method reduces the deviation between the expected and actual usage of components / parts. For a certain position *k* in the sequence, overall measure of deviation for product, *i*, is given by:

$$D_{ik} = \sqrt{\sum_{h=1}^{C}[\frac{k \times N_h}{Q} - \left(X_{h(k-1)} + b_{hi}\right)]^2}$$

Where, C = The number of variety parts used by the line

Q = The total quantity of all products to be assembled

N_h = The total quantity of component part *h* required to assemble all *Q* items in the final assembly sequence

$X_{h(k-1)}$ = The quantity of component part *h* required to assemble the first *(k-1)* items in the sequence

b_{hi} = The quantity of component part *h* required to assemble one unit of the product *i*

However this method cannot guarantee for the overall minimization of the entire sequence of the variety of products.

Goal Chasing Method - II (GCM – II)

The method decreases the computations required to determine a sequence by assigning the product that requires more production to position *k* in the sequence. This position is assigned as per the largest value attained by the following formula:

$$E_{ik} = \sum_{h \in C_i} \left(\frac{k \times N_h}{Q} - X_{h(k-1)} \right)$$

Where, C_i = The set of constituent parts for product *i*.

Hyundai's Heuristic Procedure (HH)

This method was developed for the approximation of the result produced by GCM – I by reducing the number of computations. It follows the mechanism of selecting the product which has the smallest value of D_{ik} by deducing the fact that if usage rate of one sub-product is smoothened, then the others will automatically follow the same pattern, which is unlike GCM – I, which computes usage rate over all the components.

Miltenburg and Sinnamon's Heuristic – I (MSH – I)

This method aims at smoothing the production at all levels of mixed model and multi-level production systems. Four (04) level production system is followed by this method; The level 1 was product, second level was sub-assemblies, third level was manufactured items, and fourth level was raw materials. Meaning that level one requires level two which are made from level three which in turn are derived from level four.

It selects the product *i* for the position *k* as per the smallest value derived by the following formula:

$$H_{ik} = w_1\left(X_{i1(k-1)} - kr_{i1}\right) + 0.5\sum_{j=2}^{4} B_{ijk}$$

where, B_{ijk} is:

$$B_{ijk} = w_j \sum_{h=1}^{n_j} [\left(X_{hj(k-1)} + b_{hij}\right) - \left(XT_{j(k-1)} + bt_{ij}\right)r_{hj}]^2$$

Miltenburg and Sinnamon's Heuristic – II (MSH – II)

This method focused on an area upon which none of MSH – I or GCMs focused, i.e., effect of the decision made at position *k* on the following positions (e.g., *k+1*). The measure of the deviation for product *i* scheduled in position *k* is given by:

$$V1_i = w_1 [(X_{11(k-1)} - kr_{11})^2 + (X_{21(k-1)} - kr_{21})^2 + \cdots + ((X_{i1(k-1)} - kr_{i1})^2 + \cdots + \left(X_{n_1 1(k-1)} - kr_{n_1 1}\right)^2] + \sum_{j=2}^{4} B_{ijk}$$

This method helped to calculate the deviation on all levels if product *i* is scheduled in the k^{th} position.

Extended Goal Chasing Method (EGCM)

In EGCM, Toyota's GCMs were extended to consider all levels in a multi-level production system and the smallest measure of deviation for product *i* scheduled at position *k* was calculated by the following formula:

$$ED_{ik} = \sum_{j=1}^{4} \sum_{h=1}^{n_j} w_j [(X_{hj(k-1)} + b_{hij}) - \frac{k \times N_{hj}}{Q}]^2$$

The six heuristics mentioned above may be used to solve the problems of sequencing via mathematical techniques.

Duplaga & Dragg (1998) also carried out their research on the change in the relative ranking of each of these six heuristics methods, subject to Problem Size (which involve the quantity of all the products to be sequenced), the Product Range (Ratio of number of different products to problem size), the product structure (number of components per product), and the Product Mix. After the discussion of different heuristics involved in the second goal, the first goal which is related to leveling of the workload on each station on the assembly line, will be discussed in detail in succeeding paragraphs.

As per Tunal et. al (2010), several formulations have been proposed in the literature to check the efficiency of different production sequences, which includes the minimization of the total utility work, to keep a constant rate of part usage, minimization of the total setup cost, minimization of the risk of stopping a conveyor, minimization of the overall line length and to level workloads among different work stations, etc. However, in real world conditions, to solve a single objective problem seems meaningful in practice but it does not happen all the time. Multiple objective problems need to be considered simultaneously due to their conflict with each other. As a result the sequencing problem becomes a Multi Objective problem. These Multiple objective problems, as per Cevikcan (2011), have been solved in literature by heuristics (e.g., Kotani et.al, 2004), metaheuristics (e.g., Tunail, 2010) and mixed integer linear programming (MILP) models (e.g., Giard and Jeunet, 2010).

‘Exact Approaches’ involve MILP, Constraint Program and Branch & Bound Algorithm, whereas ‘Heuristic Methods’ involve Greedy Approach, Local Search Approach, Genetic Algorithm, Ant Colony Optimization and Particle Swarm Optimization (Gang et. al, 2011) [Refer to Figure 1].

Some of the other MMALS heuristics include Cycle Time Based Sequencing Algorithm (CTBSA) and Demand Based Sequencing Algorithm (DBSA). The former works on the positioning of models with longer and shorter processing times consecutively with the aim to decrease the utility time and hence facilitate workload leveling among stations. The later technique takes into account the demand of each model and works out the leveling of workload without including time-based parameters.

The literature further shows that the Computational Complexity of MMALS-Ps increases exponentially with the passage of time. So, ‘Heuristic Methods’ have been considered the most effective of all the techniques to solve Assembly lines sequencing problems. Evolutionary algorithms, As per Gagne (2011), provide the most reliable solutions when traversing a large ‘Search Space’ to generate an approximation of the Pareto Optimal front in a single optimization step. In Pareto optimality, resources are assigned in such a manner that it is not possible to make one individual better without making another individual worse.

As per Melanie (1999) the mechanism of evolution suites the most pressing of the computational problems in many fields e.g. scenarios require searching through a large number of possibilities for solutions. However, they evolve from the Darwin’s ideas of ‘adaptation’ and ‘evolution’ who specified. ‘Natural Selection’ as the primary driving mechanism of evolution in 1859 (Charbonneau, 2002). The concept of natural selection revolves around the idea of ‘survival of the fittest’ where individuals survive and leave behind more off springs than their less apt colleagues. Two ingredients are necessary for a natural selection to lead to evolution Charbonneau (2002).

Sivandam and Deepa (2008) emphasized that the advantages of evolutionary algorithms are numerous. First of all, they are conceptually very simple as shown in the flowchart below:

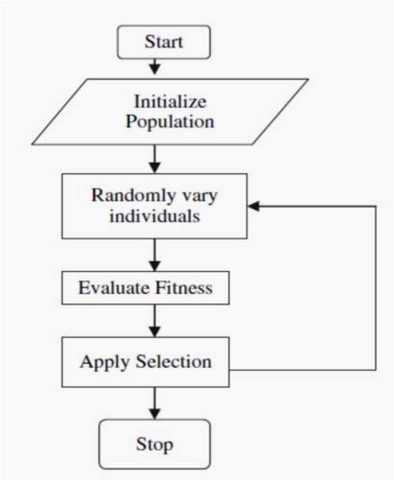


Figure 10. Flowchart for an Evolutionary Algorithm (Sivandam and Deepa, p-10)

Secondly, they have a broad applicability as they can be applied to any kind of a problem which falls in the domain of ‘function optimization’. For instance, a small/large change in parents will trigger a small/large change in off springs, subsequently giving the evolutionary algorithms the ability to tune themselves. Third, they have the capacity to hybridize with other methods. In addition, EAs have a high ability to respond to changes in the environment making them dynamically Robust. And lastly, they have the mentioned the advantage of solving problems that have no solution at all, as per human expertise. For example, Artificial Intelligence (AI) is way at the lead in terms of computational speed as compared to human intelligence. But it can be said about AI that, “They solve problems, but they do not solve the problem of how to solve problems” (Fogel 1966). This is where evolutionary algorithms step in and solve what is not solvable.

CHAPTER 4: CONVENTIONAL AND EVOLUTIONARY OPTIMIZATION TECHNIQUES

Some of the conventional and evolutionary optimization techniques will be discussed under the following headings:

4.1 GRADIENT-BASED LOCAL OPTIMIZATION METHOD

This method is employed when the ‘Objective Function’ is smooth and the goal is an efficient local optimization. Moreover, the performance of such kind of methods varies with each iteration progressed (Sivandam and Deepa, 2008). Overview of gradient based technique involves having an objective function such as:

$$f(x): R^n \rightarrow R$$

So, if a good optimal solution is required, the gradient methods must be hybridized with other search techniques (Sivandam and Deepa, 2008). Three distinct methods include;

- **Newton’s method** (an iterative numerical method to find roots of equations). If computation power required for second derivatives is less, then Newton’s method will give the best results.
- **Conjugate gradient method** (Ax = b, it’s an iterative method to solve large systems of linear equations).
- **Secant method** (it incorporates a succession of roots of secant lines for better approximation of a root of a function and requires two initial guesses for the root). The secant method is faster than conjugate gradient method but has memory problems associated with it.

4.2 RANDOM SEARCH

It is a very basic method which is rarely used by researchers in the past to randomly select solutions and evaluate their fitness. The qualities associated with it involve its ability to be improved, never get stuck up in any point such as local minima, and the guarantee to reach optimal solution (for finite search space). Moreover, finite space is very less starting point to look into and hence, this search technique loses its importance, if the finite space is very less for the starting point (Sivandam and Deepa, 2008).

4.3 STOCHASTIC HILL CLIMBING

This method is a ‘stochastic’ and ‘local’ optimization algorithm. This is a direct search technique that does not require derivatives of search space as were needed in the gradient based techniques (Brownlee, 2012). These require a ‘gradient’ to guide the direction of search (Sivandam and Deepa 2008). The technique converges for continuous fitness function, towards an optimal solution if, first, the fitness function is continuous and second, the fitness function has only one peak, so it is unimodel technique. This adds a constraint in the effectiveness of the algorithm as it tends to stop on any first peak found regardless of it being the highest or smallest. Hence, the hill climbing procedure is unable to continue as the first peak found becomes the local optimum.

As the hill climbing algorithm starts by randomly selecting a single point in the search space, numerous iterations are done to find many local minima with a goal to reach global minimum or optimum. In a nut shell, it works fine if there are not many local minima in the search space and it has a non-noisy fitness (i.e. having lesser peaks). Brownlee (2012) has further discussed the famous working domain of hill climbing algorithms as ‘discrete’ with explicit neighbors. However, they can be also applied to continuous domains by using a step size to define candidate solution neighbors. Last but not least, as stochastic hill climbing is a local search technique, it can be best utilized by refining an existing solution already worked on by an evolutionary algorithm.

4.4 SIMULATED ANNEALING (SA)

It was developed in 1983, to deal with nonlinear problems. This technique uses the analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (called annealing) and looks for a minimum on a ‘global’ scale (Prajapati et.al, 2010).

In a natural annealing process, a metal is slowly heated and cooled under self-controlled conditions to increase the size of crystals within the material to reduce/eliminate defects. It in turn increases the strength and durability of the subject material. The same happens in a SA technique with each configuration of a solution in the search space representing a different internal energy of the system with heating resulting in the relaxation of the acceptance criteria of samples chosen from search space and cooling doing the vice versa. That is, to focus on picking the best samples. Once the whole system cools down completely, the resulting configuration is or nearly is close to a global optimum (Brownlee, 2012). Usually, SA technique starts from a higher temperature and then cools exponentially. The probability of reaching the resulting configuration is given by Gibbs Law as follows (Sivandam and Deepa, 2008):

$$p = e^{\frac{E}{k^T}}$$

Where E = energy, k = Boltzmann constant, and T = temperature.

The SA technique searches for a new solution in the neighborhood of the actual solution but with the possibility of going downhill to avoid being stuck up at local minima (Sivandam and Deepa 2008). When the fitness function is saturated, the new solution is kept with the following probability:

$$p = e^{\frac{-(f(y)-f(x))}{k^T}}$$

Where, f(y) – f(x) shows the difference between the new and the old solutions.

The SA technique including other search techniques produces excellent results, which makes SA a good competitor of Genetic Algorithm (GA). Sivandam and Deepa (2008) also mention the comparison between SA and GA as following:

- Both the techniques have been derived from analogy, i.e., Natural system evolutions
- Same kind of optimization problems can be dealt with both techniques
- GA uses a population based selection which makes it more efficient, while SA deals with one individual at each iteration
- GA covers a larger area of search space at each iteration which increases its reliability, while SA iterations are much simpler which make it much faster.

- GA uses recombination operators while SA uses simple techniques (again much faster)

4.5 ANT COLONY OPTIMIZATION (ACO)

The ACO method is a stochastic population based technique which was developed initially by Dorigo and Caro to copy the behavior of ants which is based on ‘pheromone’ trail laying behavior. Pheromone is a substance that real ants put on the path on the ground they have already passed. The substance is then smelled by the following ants which lead them to follow the previous ants. When the number of ants passing a particular path increases, then concentration of pheromone dropped on the path also increases. It should however be noted that this substance is evaporated and doesn’t remain on ground for a longer period of time (Solimanpur et. al, 2010). Hence, it can be deduced that the amount of pheromone on shorter path will be greater than that on the longer path and will have a greater chance to be selected.

The probability for an ant to choose the shorter path is given by:

$$p_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h}$$

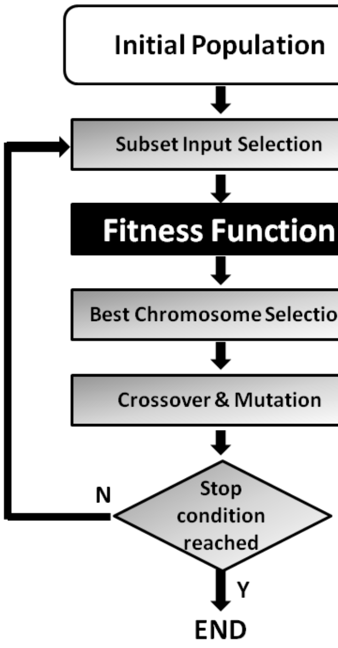
where, m_1 = the number of Ants that use the shorter path
m_2 = the number of Ants that use the larger path
k,h = fitted to the experimental data

However, the ACO technique is a versatile Algorithm which can be applied to the same versions of a problem. For example, it can be used to solve, the TSP, Sequential Ordering Problem, Network Routing etc.

4.6 GENETIC ALGORITHM (GA)

Genetic Algorithm (GA) is the most popular method employed in Evolutionary Computations (ECs). It has been recognized as simplified computational model of biological evolution (Tunal et. al, 2010). Although, it is often regarded as a ‘function optimizer’, but the range of problems it can cover/solve is quite broad (Whitley, 1994). The current study also uses the same technique.

Almost every evolutionary Algorithm has been successful, as far as reaching a solution is concerned. However, the domain to focus should contain all those Algorithms which produce the best results on a consistent basis and GA is one of them. It was first created by John Holland and his students in 1960s and 1970s at University of Michigan. Box (1957), Friedman (1959), Bledsoe (1961), Bremermann (1962), and Reed, Toombs, and Baricelli (1967) all contributed in the area of GAs as well (Melanie, 1999). These early authors focused more on moving from one population to another new population of chromosomes/bits/strings after combining the ‘natural selection’ and genetics (operators) (crossover/mutation/inversion). Actually it was Holland who carried out the documentation of the entire science behind the GAs in the form of ‘schemas’ in 1975 (Melanie, 1999). These document forms the basis of all the research being carried out around the globe on GAs today, Especially in the field of ‘mathematical technology transfer’ where in few models/methods depict a huge domain, solve various problems (Sivandam and Deepa, 2008).



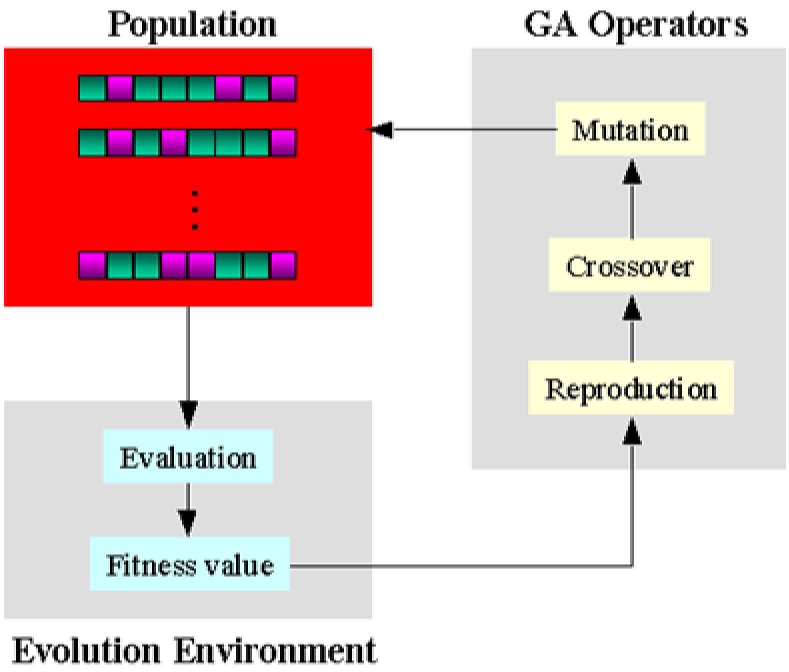


Figure 11. Genetic Algorithm Evolution Flow

Different Terminologies associated with GA

Cell	Cell is the building block of life. There are numerous small factories working together in a human/animal cell and the center of a cell is called nucleus which contains all the genetic information.
Chromosomes	These are present within the nucleus containing all the information of genetics. Each of the chromosomes is built from Dioxy Ribo Nucleic Acid (DNA).
Genes	Every chromosome is split to several parts called genes. For example, a gene can be the ‘eye color’ of a human.
Alleles	The maximum possibilities that can stem from one gene are called alleles. In the example of eye color as a gene, the eyes shades such as black, brown, blue, gray, etc. are the alleles.
Gene Pool	The set of all possible alleles in a specific population makes a gene pool. The pool has the capability to calculate all the different possibilities of variations in the future generations. The diversity of individuals in population hence, has a strong link with the size of the subject pool.
Genome	It consists of set of all genes of a particular specie.
Locus	The unique position a gene holds on a genome is called locus.

Genotype	A complete set of genes of an individual is called genotype. Any two individuals having same set of genes have the same genotype.
Phenotype	When a genotype is decoded, it produces phenotype, i.e., it contains mental and physical characteristics of an individual like eye colour, height, intelligence, brain size, etc.
Diploid	The organisms whose chromosomes are arrayed in pairs are called diploids.
Haploid	The organisms whose chromosomes are unpaired are called haploids. They are easy to construct and store only one set of each gene.
Fitness	It is the probability with which the organism will live long enough to reproduce.

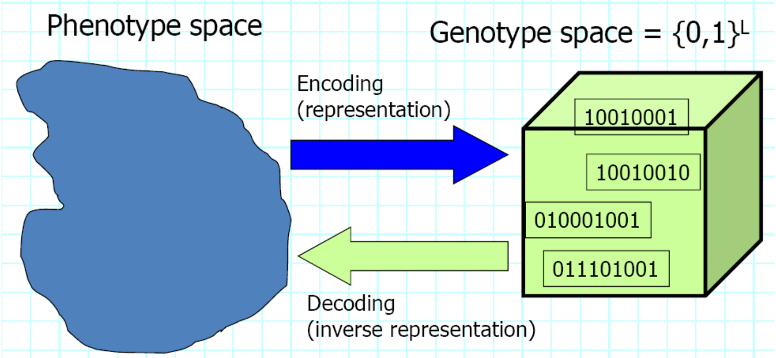


Figure 12. Phenotype and Genotype space

Chromosome is referred to as the candidate solution to a problem having several **genes** (Melanie, 1999). **Fitness** score is assigned to each chromosome (solution) and the individual with the optimal fitness is searched for in a course of several generations.

Figure 13 shows a pictorial form of a **population**, which is a space where GAs always finds the relevant matches:

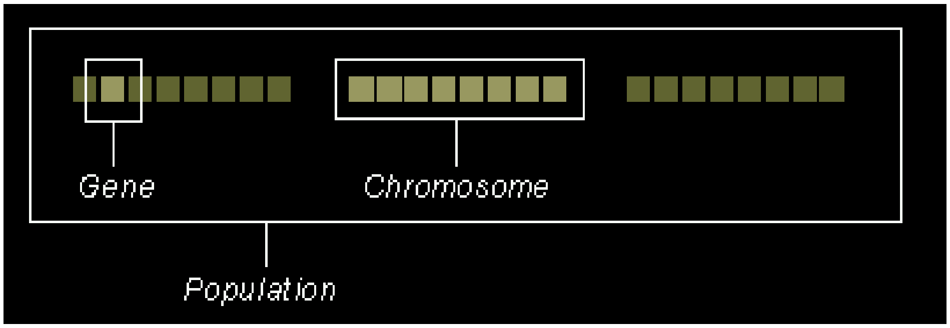


Figure 13. A Population (Simplified)

The Search **Space** (Figure 14) consists of all the feasible solutions or it is the area which is explored for feasible solutions. Normally an extreme value (min or max value) is sought for (Obtiko, 1998). Genetic Algorithm is used to find this extreme value:

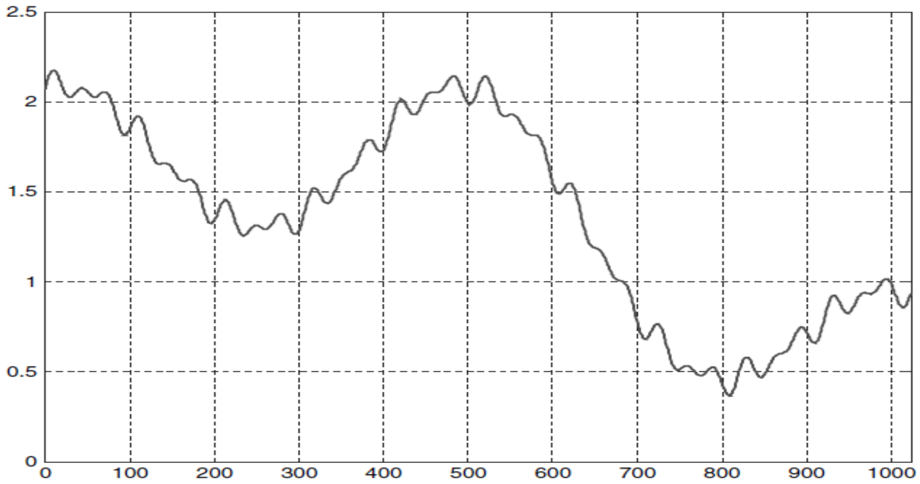


Figure 14. A Search Space

Reproduction or recombinations are often used to represent Crossover and Mutation, respectively in Natural selections and in genetic algorithms. Another famous GA operator is Inversion. The genetic operators don't introduce any new solutions into the search space but rather form an intermediate population. They allow GA to discover the required fit and low order schemata over time (Rekiek et. al, 2006). Details of these genetic operators and other terms (mentioned below) have been given in the succeeding paragraphs:

- Representation
- Types of encoding
- Sampling mechanism
- Genetic operators
- Fitness value
- Population
- Stopping criteria
- Advantages ,limitations and application of GA

4.7 REPRESENTATION

Three procedures involve in the representation i.e. encoding, infeasible solution and chromosomes & solution spaces. It is necessary to devise a suitable representation before attempting to formulate a GA (Rekiek et. al 2006).

4.7.1 Encoding

Coding must be carried out as per the search problem. This should be able to provide an easy but strong mapping between the solution space and search space.

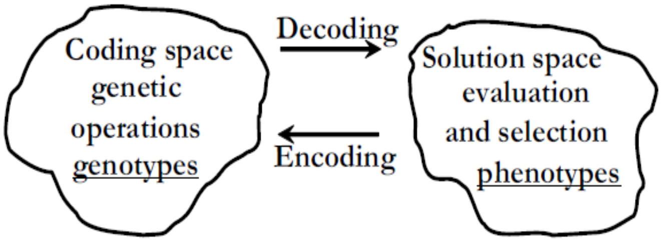


Figure 15. Encoding and Decoding of a GA (Rekiek, page-32)

4.7.2 Infeasible Solutions

GAs apply 4 basic strategies when dealing with infeasible solutions. First, Rejection (discards all the infeasible individuals), Second Repair (i.e. Recreates the feasible solutions), Third, Modification of genetic operator (To create feasible solutions), and Fourth, Assignment of penalties (of infeasible solutions recombine to produce feasible solutions).

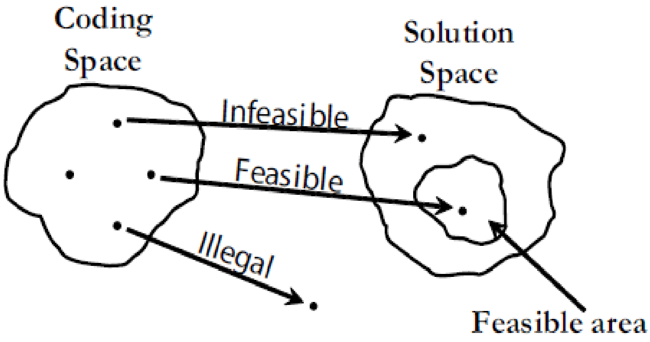


Figure 24. Feasibility of solutions (Rekiek, page-34)

4.7.3 Chromosomes and Solution Spaces

It is thought normally that in a GA, one to one mapping is best, in which each chromosome is represented by one solution and each chromosome decodes exactly each solution, but this kind of coding is superfluous which results in dropping the GA efficiency considerably. And ‘n’ to one mapping on the other hand, is another way of mapping but it also, lacks details. So, a correct mapping is a key to the success in a GA.

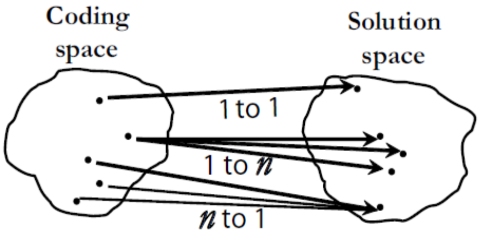


Figure 17. Mapping from Encoding to solutions (Rekiek, page-35)

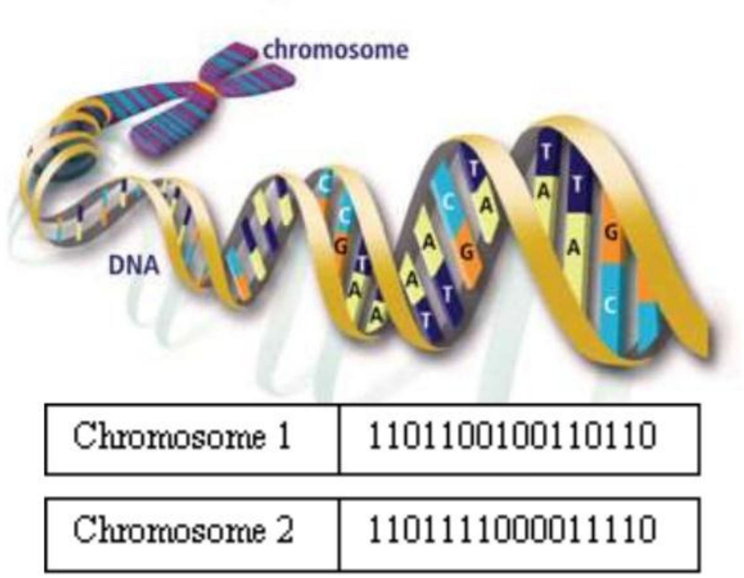
At times it may be tedious to calculate the optimal representation, which not only caters for the search space structure but also caters for the genetic operators as per the properties of the problem at hand (Sivandam and Deepa 2008).

4.8 TYPES OF ENCODING

It includes the following types of encoding (Obtiko, 1998):

4.8.1 Binary Encoding

In this category a chromosome is coded as a string of bits of 0 and 1.



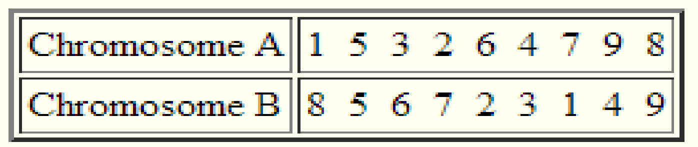
Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

Figure 18. Chromosomes with Binary Encoding

‘Knapsack problem’ is an example related to the application of binary encodings where the objective is to select values that maximize the value and do not exceed the knapsack capacity.

4.8.2 Permutation Encoding

In this category a chromosome is coded as a string of numbers which represents a number in sequence. Example of this type of encoding is travelling problems.

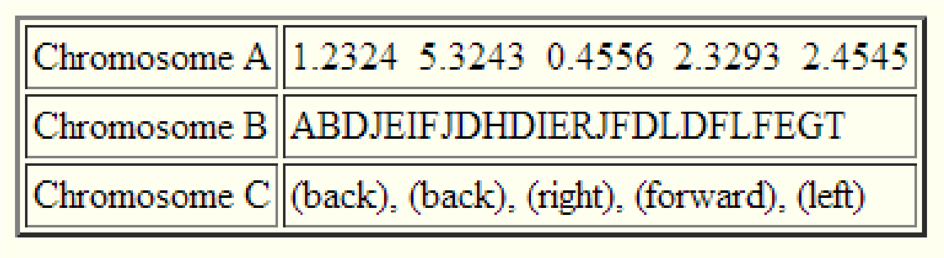


Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Figure 19. Permutation Encoding

4.8.3 Value Encoding

In this type of encoding a chromosome is coded as a string of values (numbers, characters, etc.).Value encoding is used when many variables are used in the problem, i.e., complex situations. Neural Network is the example of this type.



Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

Figure 20. Value Encoding

4.8.4 Tree Encoding

In this case a chromosome is coded as a tree of objects like functions and commands in a programming language. This type of encoding is used for genetic programming. E.g. chromosomes represent the functions:

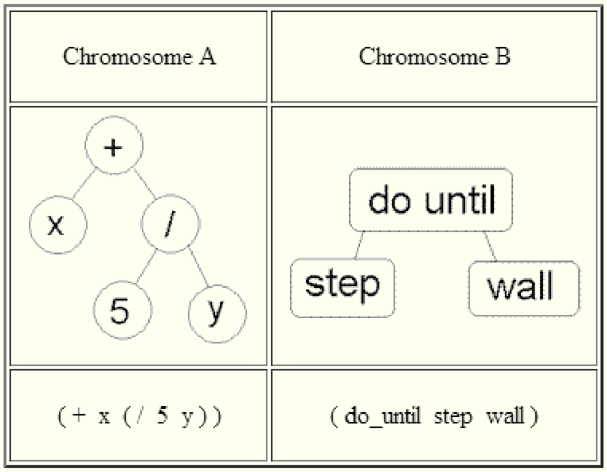


Figure 21. Tree Encoding

4.9 SAMPLING MECHANISM

Evolutionary Algorithms revolve around the technique of ‘survival of the fittest’ and therefore, selection plays a vital role in it. It compares fitness value of each individual via ‘fitness function’ in the population. Actually, this phase involve the selection of chromosomes from the population as parents for either of the genetic operators discussed above. Each shares a Terminology of the sampling mechanisms has been summarized below (Obtiko, 1998):

• **Selective Pressure(SP)**	It is the probability of the best individual compared to the average probability of selection of all the individuals. SP = 2-10 is successfully applied to various GAs. SP > 4 or 5 is considered strong.
• **Bias**	It is the difference between an individual’s normalized fitness and its expected reproduction probability.
• **Spread**	This constitutes the limit of possible values against the offspring number of an individual.

- **Loss of diversity** It is the proportion of population individuals which is not selected during the selection phase.

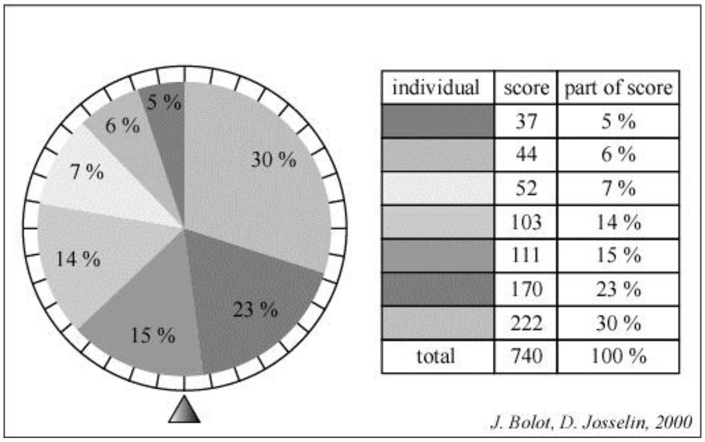
- **Selection Intensity** It is the expected average population fitness value that is obtained after application of a selection method to the normalized Gaussian distribution.

- **Selection Variance** It is the expected fitness distribution variance of the population after application of a selection method to the normalized Gaussian distribution.

The most commonly used mechanisms are as follows:

4.9.1 Roulette Wheel Selection (RWS)

In this technique, single slot of the wheel is paired with population individual and the size of this single slot is proportional to the corresponding fitness of the individual. Number of times the roulette wheel is spun is equivalent to the size of the population.



individual	score	part of score
▨	37	5 %
▨	44	6 %
▨	52	7 %
▨	103	14 %
▨	111	15 %
▨	170	23 %
▨	222	30 %
total	740	100 %

J. Bolot, D. Josselin, 2000

Figure 22. Roulette Wheel Selection

4.9.2 Ranking Selection (RS)

It comes into play when the solutions are selected proportionally to their rank instead of their evaluation (Rekiek et.al, 2006). Population in this case is sorted according to the objective values (Pohlheim 2006). However, it is famous for resolving issues pertaining to stagnation due to too small selection pressure or premature convergence. So, uniform scaling and control of selective pressure are the mainstream attributes of ranking selection.

4.9.3 Tournament Selection (TS)

Several tournaments among a few individuals are run. These individuals are chosen randomly from the population. Following three steps are included in TS:

- Randomly select a group of members from the current population
- Fitness is compared among the chosen members
- Select the fittest individuals from the group to propagate to the next generation

When the tournament size is higher, weak individuals tend to have a smaller chance of getting selected, i.e., SP increases. Pohlheim (2006) has pointed out the parameter for TS as ‘Tour’, which takes the values ranging from 2 to 9. The following table shows the relation between ‘tournament size’ and ‘selection intensity’:

Tournament Size	**1**	**3**	**5**	**7**	**15**	**32**
Selection Intensity	0	0.561	0.852	1.153	1.538	2.045

Table 1. Tournament Selection vs Selection Intensity (Pohlheim, 2006)

The Selection intensity varies and is given by the following formula:

$$SelInt(Tour) \approx \sqrt{2(\ln(Tour) - \ln(\sqrt{4.14\ln(Tour)}\,)}$$

The formula for the Loss of Diversity is given below:

$$LossDiv(Tour) = Tour^{\frac{-1}{Tour-1}} - Tour^{\frac{-Tour}{Tour-1}}$$

And, finally the Selection variance is given as follows:

$$SelVar(Tour) \approx \frac{0.918}{\ln(1.186 + 1.328\,Tour)}$$

4.9.4 Truncation Selection

It is an artificial method used by breeders for larger populations (Pohlheim 2006). The sorting is carried out according to individual fitness and the best individuals are selected as parents. Its parameter is Truncation Threshold, which represents the proportion of population to be selected and it lies in the range [upto 60%]. Table 2 shows the relationship between them:

Truncation Threshold	**5%**	**15%**	**25%**	**45%**	**60%**	**90%**
Selection Intensity	2.662	1.764	1.232	0.971	0.801	0.343

Table 2. Truncation selection vs Selection Intensity (Pohlheim, 2006).

4.10 GENETIC OPERATORS

There are three kinds of genetic operators like **inversion, crossover and mutation.** Before application of these operators, the sampling mechanism tries to 'copy' some of the solutions to make an intermediate population after that these operators are applied. So no new solution is generated or brings forth new solutions from the search space. Hence the concept of 'Recombination' steps in via three distinguished genetic operators (Rekiek et.al, 2006). Following paragraphs explains the above mentioned operators:

4.10.1 Crossover

It is a mechanism in which two of the individuals are selected and the operator ‘cross over’ is applied. It has two parameters, first crossover probability and second crossover operator. Members selected as per the sampling mechanism either go to a cross over operator or are passed unchanged into the next generation as per a crossover probability in the range 0-1, where probability of 0 turns off the crossover and 1 ensures that the crossover is always applied. The aim of this operator is to find new solutions in a search space. The types of crossover are as following:

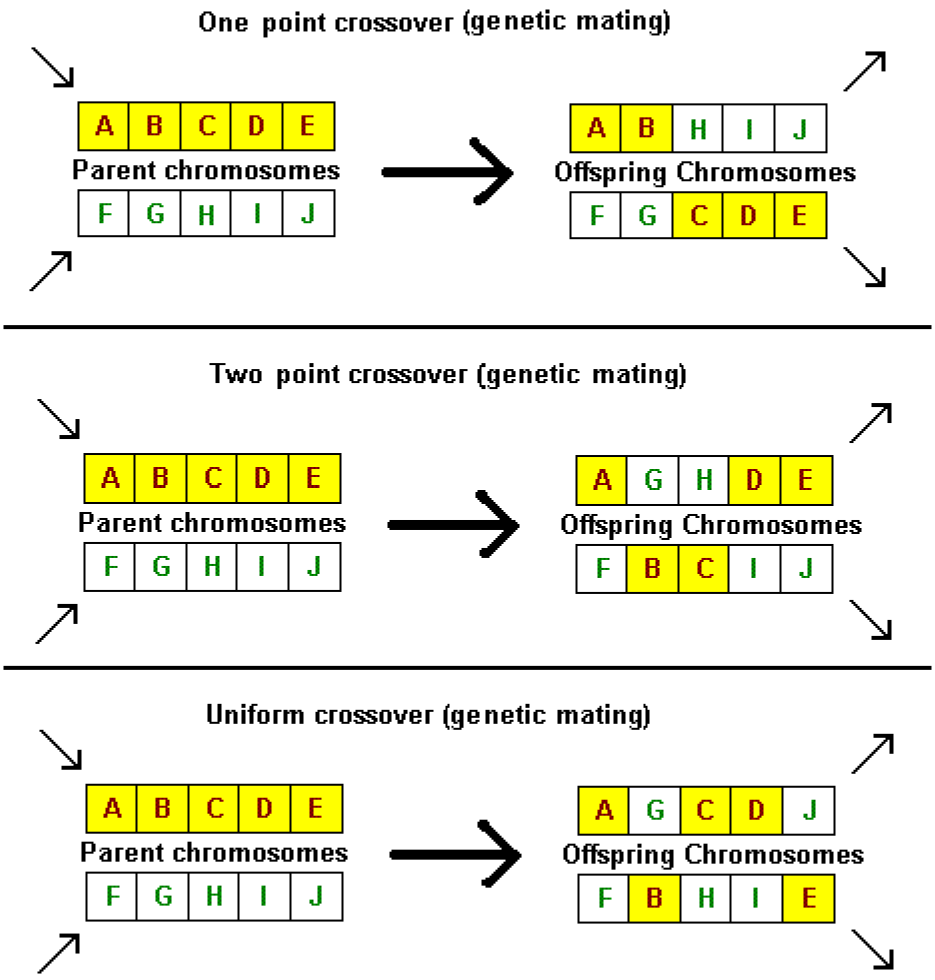


Figure 22. Types of crossover

Single Point Crossover

In this method a single crossover point is selected. The portion of the values from beginning of chromosome to the crossover point is copied from one parent and the remaining portion is copied from the other parent.

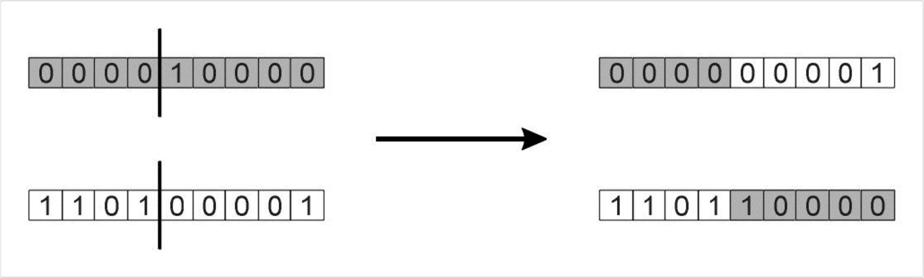


Figure 23. Single Point Crossover (Binary Encoding)

Two Point Crossover

In this mechanism, the difference lies in the selection of two points instead of one for the crossover.

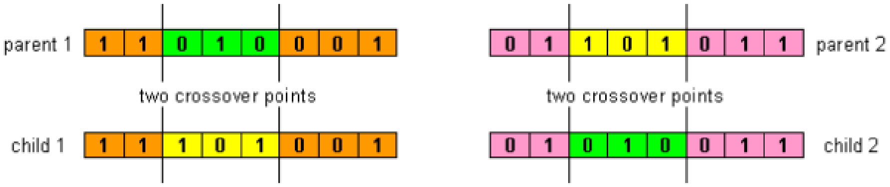


Figure 24. Two Point Crossover

Uniform Crossover

This category makes things easier by making each locus as a potential point for crossover. Figure 24 shows the example of uniform crossover where parents contributes on the basis of equal probability to produce the offspring. The resulting off springs are as following:

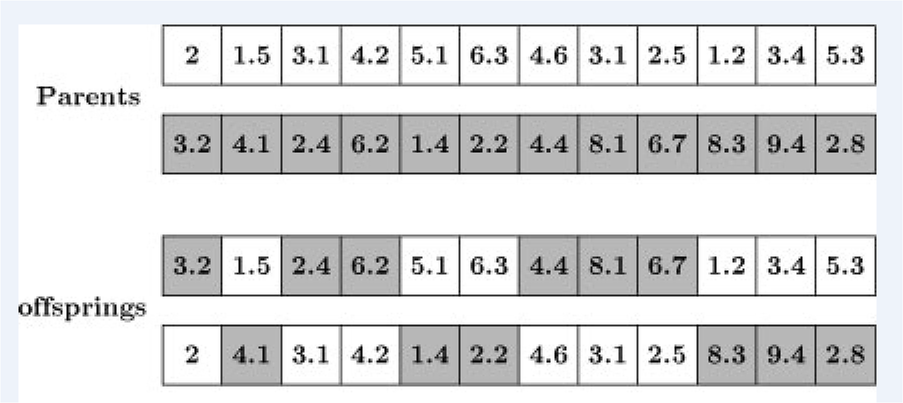


Figure 25. Uniform Crossover

4.10.2 Mutation

The individuals are randomly altered in mutation. As discussed in crossover, two parameters are associated with mutation as well; mutation probability and mutation operator. The first applies a probability in the range 0 to 1. The literature surveyed conveys that as mutation tends to slow the convergence of a GA, hence, the mutation probability should be set very low (such as 0.05). And the second involves various kinds of operators but the overall concept revolves around two distinct operations; one, where a function replaces a function or a terminal replaces a terminal, and two, where an complete sub tree replaces by another sub tree as follows (Tree Encoding) [1]:

---

[1] http://www.geneticprogramming.com/Tutorial/index.html

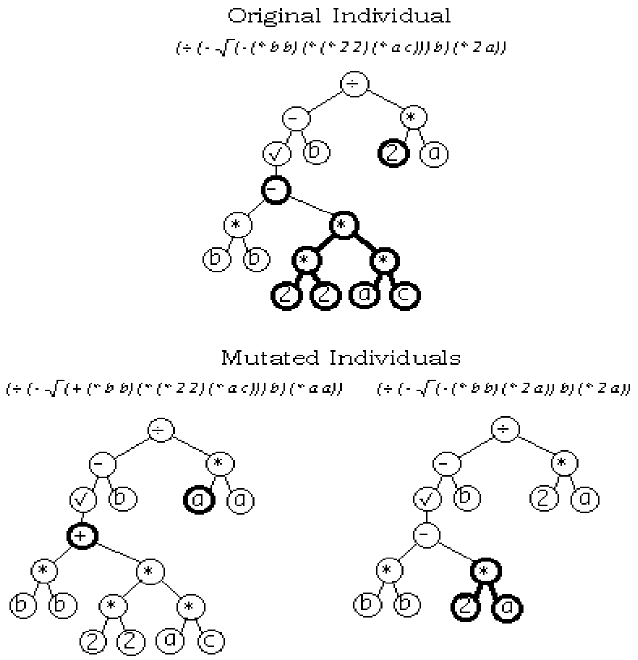


Figure 26. Scenarios of Mutation

The types of mutation operators are as following:

- Bit Inversion
- Order Changing
- Real Value Mutation

4.10.3 Inversion

To rectify the shortcomings of the crossover operator this genetic operator is utilized. The crossing sites in a crossover are selected at random it results in the disruption of longer schemata.

4.11 FITNESS VALUE

In order to optimize a given problem, an objective function is used to derive the maximum or minimum of a given function. The solutions resulting in the process are the chromosomes and each chromosome has a certain **fitness value** (Rekiek et. al 2006). It is obtained after application of the fitness function which must be cleared defined.

4.12 POPULATION

It is the heart of all evolutionary algorithms as most of them use it as a search space. Size of population shows the number of states chromosomes in a single population or one generation (Obtiko, 1998). If the amount of chromosomes is very large, GA will slow down and if the chromosomes are very less, the options of applying genetic operators decrease as well. So, as per the encoding of a problem, population size should be chosen very wisely. However, the population size can remain fix or can be updated via two distinguished modes; First, steady state update and second, generational update. In the first approach, single offspring is produced which replaces a single member of the older population. And in the second type, N off spring are produced from a population size N for next time step. As a result the new population completely replaces the parent population.

Basic Algorithm of a GA can be stated as follows (Sivandam and Deepa, 2008):

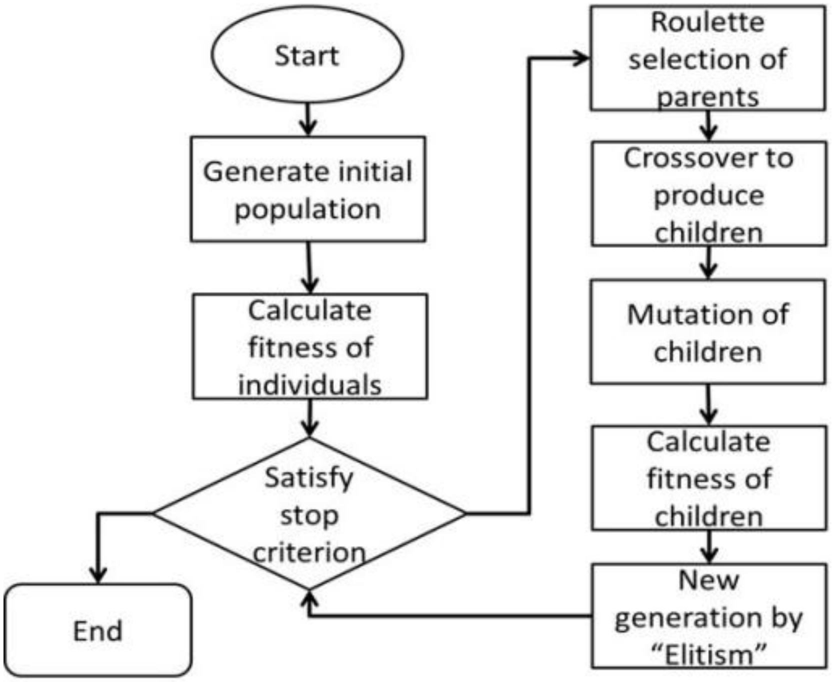


Figure 27. Basic Algorithm (GA)

4.13 STOPPING CRITERIA

Stopping Conditions of genetic algorithm can be either of the many given below:

4.13.1 Generations

Number of generations can be set as the stopping criteria for the algorithm to stop when it reaches the desired fixed value of generation.

4.13.2 Time Limit

Specified time limit can also be used to stop the algorithm.

4.13.3 Fitness Limit

Another stopping criterion is fitness limit. Desired fitness value can be set to stop the algorithm.

4.13.4 Stall Time Limit

When there is saturation or no further improvement in the objective function, specified time interval can be used to stop the algorithm.

4.14 ADVANTAGES AND LIMITATIONS OF GAs

GAs have many Advantages and number of limitations. Few of them, as discussed by Sivandam and Deepa (2008), have been listed below:

4.14.1 Advantages

- It has a wider Solution Space.
- It can easily discover the global optimum.
- It can be used for multi-objective problems (MOPs).
- It only uses function evaluations.
- It can easily be modified for different problems.
- It can handle noisy functions well.

- It has complex fitness landscape.
- It can perform very well when the size of problems of optimization increases.

4.14.2 Limitations

- It cannot use gradients.
- There is a problem of identifying fitness function.
- There may be premature convergence which can occur sometimes.
- For identification of local optima, this technique is not rated well.
- There is problem of choosing the correct values for parameters like size of population, Mutation rate, Crossover rate, mechanism of selection, etc.
- There is no effective terminator.
- Local search technique is coupled with it to give the best results.
- The configuration is not straight forward.

CHAPTER 5: PROPOSED APPROACH

This study focuses on the development of mixed model assembly line for armored personnel carriers (APCs), with the aim of minimizing total utility work and generating a global sequence for the variants of base model of APC, using Genetic Algorithm (GA). The current study and its application in automotive industry, is related to APC sequencing problem wherein a sequence of models was generated that minimized the total utility work in light of local problems such as balancing of a mixed model assembly line (MMAL), facility constraints (length of stations), quality control staff activities (in terms of hours spent) and support staff activities (in terms of hours spent). Each of the local problems when optimized aggregated towards the global optimization of total utility work by using the evolutionary algorithm (EA) i.e GA. 'Utility Work' is categorized as the number of working hours that additional utility workers should handle to avoid assembly line stoppages (V. Giard 2010). Costs will increase either way if additional utility workers are hired or the line is stopped because a MMAL has a fixed conveyor speed. Moreover, to satisfy the demand floated from the market, Hyun et. al (1998) proposed to maintain a Minimum Part Set (MPS) that operates in a cyclic manner in the planning horizon. This paper uses the same model as that used by Hyun et. al but with modifications in reply to the three local problems addressed above (Later on this model was also used by Tunail 2010). Instead of providing the sequence of models to the algorithm, the subject research will generate an optimized sequence matching with both the generated MPS and facility constraints. Moreover, spacing constraints were catered for as per the workstation capacity limitations.

As mentioned in the literature review the best heuristic used for mixed model assembly line sequencing problem (MMALS-P) is GA. So GA has been employed in this research to solve the MMALS-P. In previous chapter, the use of GA instead of other EAs such as ACO has already been discussed in detail. The code for the algorithm was developed in **MATLAB** Version 7.12.0.635 (R2011a) with License Number 161052.

The variables which have been used by Tunail et. al (2010) can be categorized into static and dynamic variables as following:

S. No.	Parameters	Description
1.	**M**	Number of Primary Models
2.	**m**	Number of Sub Models
3.	**J**	Number of Stations
4.	$\mathbf{t_{jm}}$	Operation Time at station ‘j’for model ‘m’
5.	$\mathbf{L_j}$	Fixed length of station ‘j’
6.	$\mathbf{v_c}$	Conveyor Speed

Table 3. Static variables of MMALS-P (Tunal. et. al., 2010)

S. No.	Parameters	Description
1	$\mathbf{d_m}$	Demand of Model ‘m’
2	$\mathbf{D_m}$	Number of products of sub model type ‘m’
3	**H**	Greatest Common Divisor (GCD) of $D_1, D_2, \ldots, D_M$
4	**I**	Number of products produced in one cycle: $$I = \sum_{m=1}^{M} d_m$$
5	γ	It is Launch Interval, which is given by: $$\gamma = \frac{T}{I \times J}$$
6	**T**	Total Time to produce one cycle of MPS
7	$\mathbf{U_{ij}}$	Utility Work required for the ‘ith’ product in a sequence at station ‘j’
8	$\mathbf{Z_{ij}}$	The starting position of the work, on the ‘ith’ product in a sequence, at station ‘j’
9	$\mathbf{x_{im}}$	It is considered as one, if the ‘ith’ product in a sequence is model ‘m’, otherwise its value is 0

Table 4. Dynamic variables of MMALS-P (Tunal. et. al., 2010)

5.1 CALCULATION OF TOTAL NUMBER OF SEQUENCES

The formula which calculates total number of sequences is given by Tunail et. al (2010):

$$\text{total sequences} = \frac{\left(\sum_{m=1}^{M} d_m\right)!}{\prod_{m=1}^{M} (d_m!)}$$

where, M is equal to the number of primary models, m shows the model type of sub-model, and d_m indicate the demand for model ‘m’. It shows that whenever the size of the problem increases, the number of feasible solutions increases exponentially. It has been discussed earlier that minimization of total utility work (TUW) is one of the objectives of sequencing problems, where TUW will occur whenever the operations on a job can’t be completed within the system boundaries.

Figure 28. Model A – VCC2

Figure 29. Model B – VCC1

Figure 30. Model C – VCC1(A2+)

5.2 OBJECTIVE FUNCTION

The formulations, in this study, have been used to modify the scope of the sequencing model proposed by Hyun et. al (1998). He proposed that for minimization of total utility work, the following objective function is required to be optimized:

$$Minimize \sum_{j=1}^{J}(\sum_{i=1}^{I} U_{ij} + Z_{(i+1)j}/v_c) \tag{1}$$

where, U_{ij} = The amount of utility work required for the 'ith' product in a sequence at station 'j',

Z_{ij} = Starting position of the work on the 'ith' product in a sequence at station 'j', and

I = The number of products produced in one cycle.

For the planning horizon, T, which is also the total operation time required to produce one cycle of MPS products, the following modified equation has been proposed:

$$T = \sum_{j=1}^{J}\sum_{m=1}^{M} t_{jm} d_m + \sum_{j=1}^{J} QC_j + SS_J \tag{2}$$

where, t_{jm} = operation time at station 'j' for model 'm', QC_j = time consumed by the Quality Control staff at station 'j' and SS_J = time spent by the Support Staff at station 'J'.

Number of products produced in one cycle was given by the following equation (Hyun 1998):

$$I = \sum_{m=1}^{M} d_m \tag{3}$$

Having calculated I, there was a need to compute the launch interval (γ) of the models on the assembly line (Hyun 1998):

$$\gamma = \frac{T}{I \times J} \tag{4}$$

5.3 CONSTRAINTS

The modified constraints of the objective function are as follows:

$$\sum_{m=1}^{M} x_{im} = 1, \qquad \forall i, \tag{5}$$

$$\sum_{i=1}^{I} x_{im} = d_m, \qquad \forall m, \tag{6}$$

$$Z_{(i+1)j} = \max[0, \min(Z_{ij} + v_c(\sum_{j=1}^{J}\sum_{m=1}^{M} x_{im}t_{jm} + \sum_{j=1}^{J} QC_j + SS_J) - (\gamma \times v_c), L_j - (\gamma \times v_c))], \forall i, j \tag{7}$$

$$U_{ij} = \max[0, (Z_{ij} + v_c(\sum_{j=1}^{J}\sum_{m=1}^{M} x_{im}t_{jm} + \sum_{j=1}^{J} QC_j + SS_J) - L_j)/v_c], \forall i, j \tag{8}$$

$$x_{im} = 0 or 1, \forall i, m \tag{9}$$

$$Z_{1j} = 0, Z_{ij} \geq 0, \forall i, j \tag{10}$$

$$U_{ij} \geq 0, \forall i, j \tag{11}$$

Equation (5) acts as ‘model existence indicator’ and ensures that exactly one model was assigned to each position in a sequence, Eq. (6) allows the demand for each model to be satisfied, Eq. (7) gives the starting position of the worker at each station ‘j’ on product ‘i+1’ in sequence, Eq. (9) shows that x_{im} is ‘one’ if the ‘ith’ model in the sequence is model ‘m’ and otherwise it is ‘zero’, Eq. (10) proposes that first product on any station will start from the beginning of the station, and Eq. (11) restricts utility work to always stay either 0 or positive for all stations and products.

5.4 ASSUMPTIONS (TUNAL ET. AL, 2010)

- The assembly line is divided into **J** stations.
- All the work stations are closed.
- Conveyor system of assembly line is moving at a constant speed, $\mathbf{v_c}$.
- Minimum part set (MPS) production is used and this set is a vector showing a product mix such that

  $$(d_1, \dots, d_m) = (\frac{D_1}{h}, \dots, \frac{D_M}{h})$$

  It operates in a cyclic manner over the planning horizon and if the MPS is repeated ‘h’ times, in order to meet the total demand in the planning horizon.
- Launch of products on the conveyor is at a fixed rate.

- Processing times are deterministic and were part of the initial set of inputs.
- Workers’ moving time is ignored.

For the ‘modified constraints’, and generation of sequence of models, following assumptions were considered:

- Number of sub models for any primary model was less than or equal to 3, i.e., m = 1 to 3.
- The number of utility workers is fixed.
- The MMAL is already balanced.
- The facility had a fixed length of 52 m.
- Stage wise Quality Control staff was attached at the end of each station.
- Support Staff was involved at the final stage for fault rectification.
- Floating stock is available at every stage for replacement of the assemblies/sub assemblies.

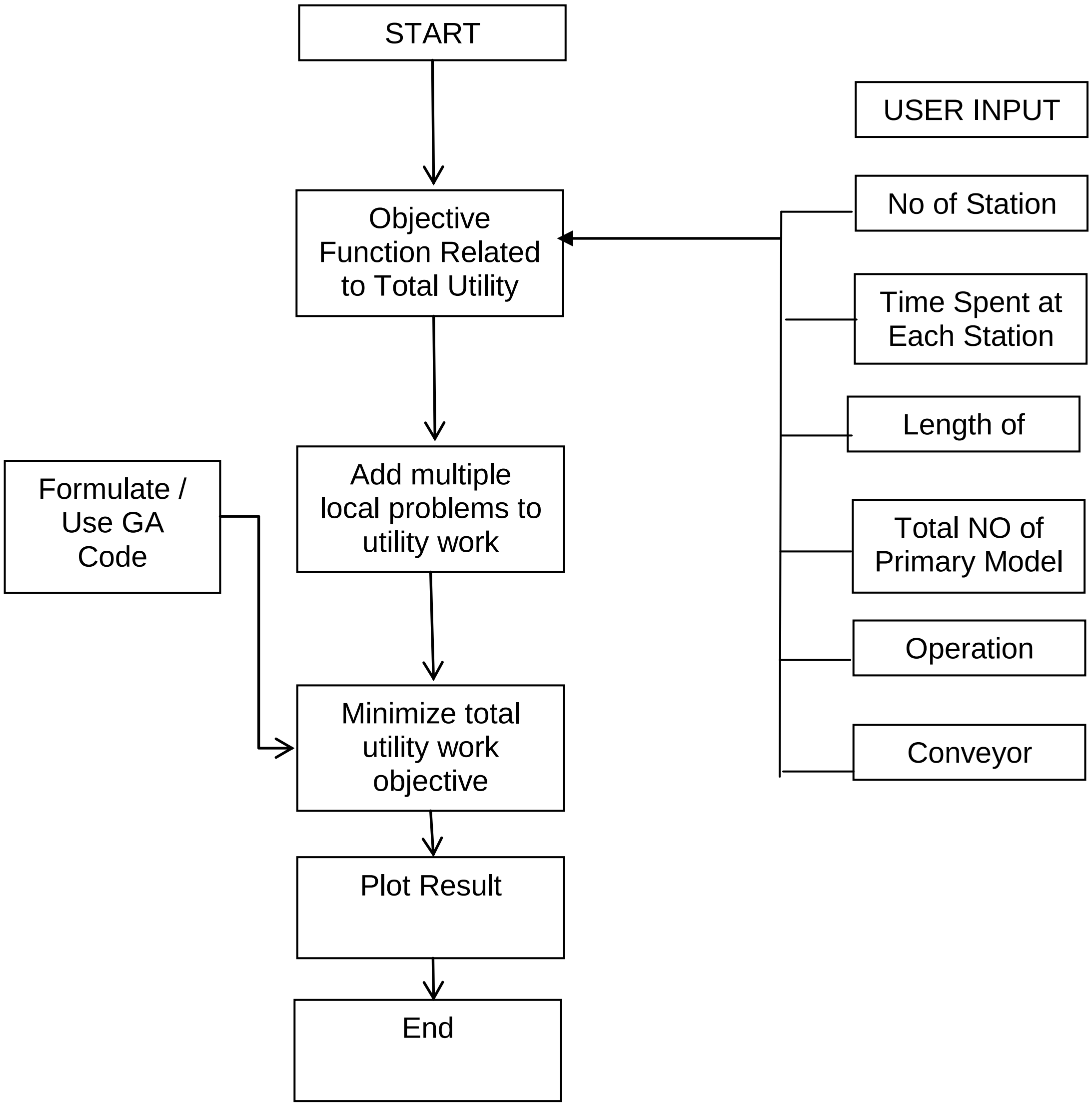


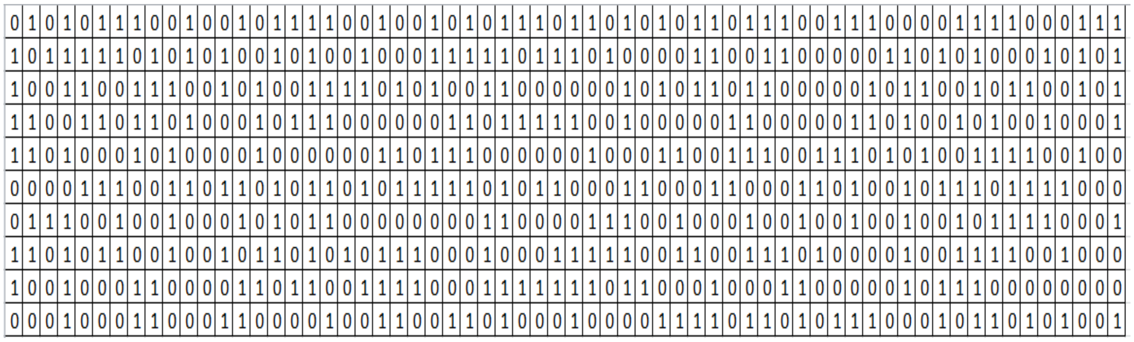
Figure 31. Methodology

The Evolutionary Algorithm, GA has optimized the objective function (Eq. 1). Designing a GA requires a careful inspection and determination for initialization of variables because it may look same for all kinds of problems, but the details of its functioning components vary with every problem at hand. So, in light of the problem definition, the components of GA were devised as given below:

5.5 REPRESENTATION AND POPULATION SIZE

Binary encoding has been used to encode the chromosomes (individuals) as fixed length bit strings of 16 bits each in a population. The procedure of encoding followed an initialization process which had the spread of population as the input and number of genes as the output. For example, if the chromosome initialized was [1 0 1 0 0 1 1 1 1 0 0 1 0 0 0 0], a process of binary to gray conversion was carried out, as gray codes make stable digitizers (U. A. Thattun 2007), to convert binary codes to gray codes. The encoded chromosome after being converted to gray codes was [1 1 1 1 0 1 0 0 0 1 0 1 1 0 0 0]. The population size was initialized as 10 showing that there were 10 chromosomes in one generation, i.e., single population.

The figure below shows a population space of 10x64 which includes 4 chromosomes of 16 bits each.



0	1	0	1	0	1	1	1	0	0	1	0	0	1	0	1	1	1	1	0	0	1	0	0	1	0	1	0	1	1	1	0	1	1	0	1	0	1	0	1	1	0	1	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1
1	0	1	1	1	1	1	0	1	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	1
1	0	0	1	1	0	0	1	1	1	0	0	1	0	1	0	0	1	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	1	0	1	1	0	0	0	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1
1	1	0	0	1	1	0	1	1	0	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	0	1	0	0	0	1
1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	0	0	1	1	1	1	0	0	1	0	0
0	0	0	0	1	1	1	0	0	1	1	0	1	1	0	1	0	1	1	0	1	0	1	1	1	1	1	0	1	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	1	0	0	1	0	1	1	1	0	1	1	1	1	0	0	0
0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	1	1	0	0	0	1
1	1	0	1	0	1	1	0	0	1	0	0	1	0	1	1	0	1	0	1	0	1	1	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	1	1	0	0	1	1	1	0	1	0	0	0	0	1	0	0	1	1	1	1	0	0	1	0	0	0
1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0	0	0	1	0	0	0	0	1	1	1	1	0	1	1	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1	0	1	0	0	1

Figure 32. Initial population

5.6 FITNESS FUNCTION

The fitness function was the same as the objective function (Eq. 1).

5.7 PROBABILITY OF CROSSOVER

The probability of crossover was kept as 0.7 because as per the literature reviewed, it seems a better practice to leave some portion of the generation to survive to the updated generation. Moreover, 'two-point crossover' was used wherein two random points are chosen in a chromosome to perform the operation of crossover.

0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	1	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	1	0	1	0	1	0	1	0	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1	1	0	1	1	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	0
0	1	1	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	1	1	1	0
0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0
0	0	1	0	1	1	1	1	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	1	1	1	0
0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	1
0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0	0	0
0	0	0	0	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1	1	0	0

Figure 33. Population after the operation of 'two point Crossover'

5.8 PROBABILITY OF MUTATION

For the case of probability of mutation, the value has been fixed as 0.15 by employing the concept of 'uniform mutation'. In this case, first, a fraction of the entries of a chromosome are selected with each value in the fraction having a certain probability of being mutated and second, each value is replaced by a random number existing in the range of the values.

0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	1	1	0	0	1	1	0	0	1	0	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1	1	1	0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	0	0	1	1	1	0	0	1	0	0	1	1	1	1
0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	1

Figure 34. Population after the operation of 'Uniform Mutation'

5.9 TOURNAMENT SELECTION AND TOURNAMENT SIZE

The tournament size has been kept as 7. The value initialized was obtained from a sampling procedure, called **tournament selection**. In this procedure, some of the individuals are chosen randomly from the population and several tournaments are run between them iteratively. It also helps to keep control over the ‘selective pressure’ that is, the ratio of the probability of the best selected individual to the average probability of selection of all the individuals. It has also been proven in research that tournament sizes between 02 and 10 prove out to be worthy for many GAs (Pohlheim 2008).

CHAPTER 6: RESULTS, ANALYSIS AND FUTURE WORK

6.1 APPLICATION OF PROPOSED SETUP FOR EXISTING SETUP

MMAL for a APC sequencing problem addresses the input of models on the assembly line in light of the demand floated from the market (d_m), i.e. army in this case. Production facilities then cater for the floated demand by splitting it into production over a certain horizon, T. The working hours available for each day at the target production facility is 06 hours. The space available is 52 m currently divided into three stations (J = 3) of length 6.5 m (L_j) each. The existing setup consists of only three assembly stations of APCs and detail is mentioned at table 6. The primary models of APC (VCC2, VCC1 and VCC1-A2+) are three i.e. M = 3. The models A (VCC2), B (VCC1) and C (VCC1-A2+) are the variations of same base product model A. Due to different variations in the optional models, there are different operational times as well. The tasks being executed on existing assembly line are not taking into account the 'precedence relationship', as a result, the present assembly line is not balance and optimized in terms of MMAL concept. Tasks of the existing setup have been tabulated against three assembly stations to accommodate this group of tasks to produce an APC. The operational tasks times related to three models of APC (A, B and C) have also been calculated at existing assembly stations, which have been mentioned at table 7.

The figure 35 shows that total space available for the assembly line is 52 m which is not being fully utilized. The tasks times mentioned in figure 35 are related to model A only. The task time of quality control staff has also been indicated in figure 35. The graphical output / representation for existing AL is shown in figure 36 and 37. Initially the code was run for 100 generations where different peaks were found at generation number 10, 18, 60 etc with the 'best of generation fitness value' as 1.85E-3, 1.9E-3 and 2.01E-3 respectively. This fitness value keep on improving and here GA recorded the best value at 60th generation and produced the minimum TUW as 644.65 (107.44 hours per product) against the best sequence as CCBBAA. The figure 37 shows that the code was run for 1000 generations, number of peaks are improving which indicates the overall improvement in objective function with the help of GA.

The following data has been given to the proposed algorithm which produced the result as shown in the succeeding tables:

Stations	**Detail**
Assembly stage I	Hull, final drive, differential assembly, sprocket wheel, installation of arms, hub assemblies and shock absorbers, road wheels, track fitting.
Assembly stage II	Cooling fan, radiator assembly, fuel tank, hoses, power pack, gauges, tachometer, linkages.
Final stage	All Beading, All seats, road test, fault rectification and final paint.

Table 5. Tasks for each assembly station on existing Assembly Line

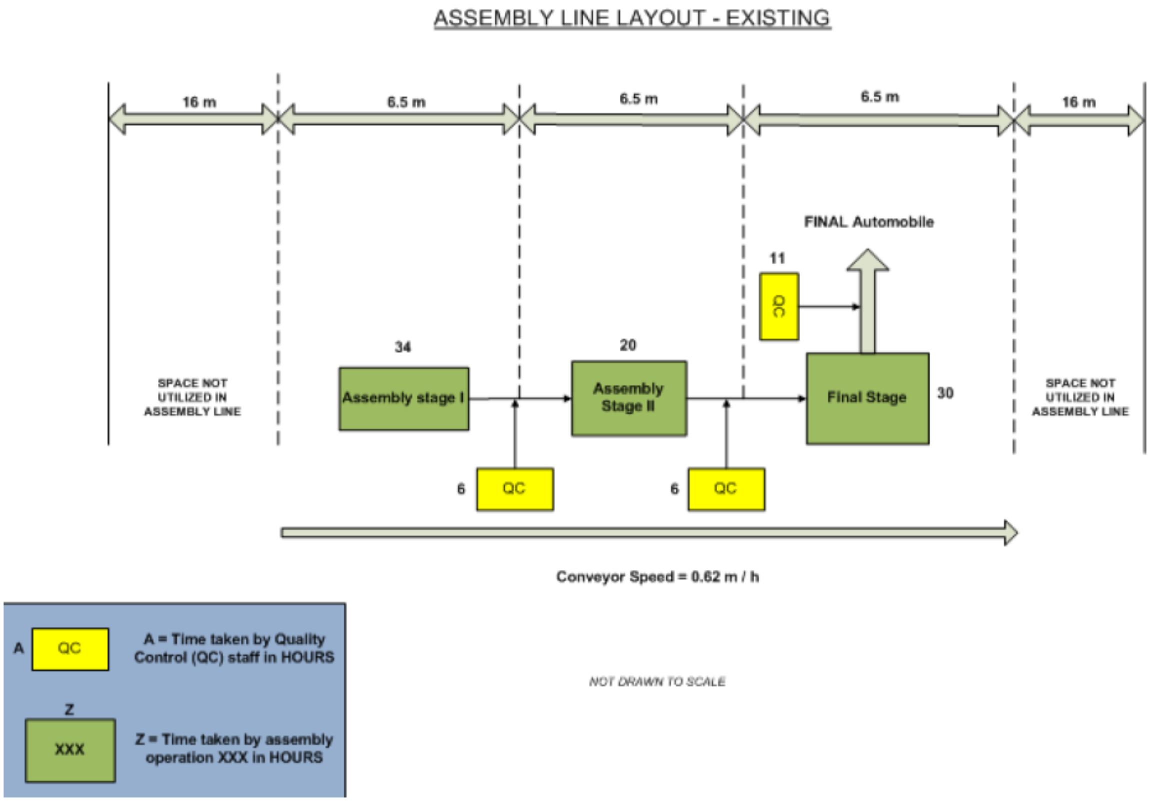


Figure 35. Assembly Stations existing setup

S. No.	Variable	Initialized value (Existing)
1	Number of Stations (J)	3
2	Length of Stations (L_j)	6.5 m
3	Conveyor speed (v_c)	0.303 m / h
4	Primary Models (M)	3

Table 6. Variables for Existing setup

Stations	Model A	Model B	Model C
1	40	42	40
2	26	26	26
3	41	41	41

Table 7. Operation times for MMAL (Hours) - Existing setup

6.2 RESULTS FOR EXISTING SET UP

S. No.	Generations	MPS	Best Sequence Generated	Total Utility Work (Hours)	Minimized Utility Work per product (Hours)
1	100	[2 2 2]	CCBBAA	644.65	107.44
2	500	[2 2 2]	CCBBAA	639.63	106.60
3	1000	[3 3 3]	CCCBBBAAA	630.65	70.07
4	5000	[3 3 3]	CCAABACBB	630.65	70.07
5	50000	[3 3 3]	CAAABCCBB	630.65	70.07

Table 8. Results of GA for - Existing setup

When the input data of proceeding tables was given to the GA code it produced the results as shown in the table 8. GA was run up to 50,000 generations. The minimum TUW comes out to be 630.65hours for a particular minimum part set and for a single product it is 70.07hours. As assembly line is not following a particular sequence and it is not optimized so we have a lot of cushion to minimize this TUW to get the best result. Table 8 shows that GA also picked up the best sequence generated against this minimum TUW at a particular generation. When code was run for first hundred generations it produced minimum TUW as 644.65 hours (107.44 hours per product) with the best sequence as CCBBAA with MPS [2,2,2] i.e for models A= 2, B= 2 and C= 2. When code was run for 1000 generations the minimum TUW comes out to be 630.65 hours (70.07 hours per product) against a MPS of [3,3,3],i.e., 3 models for each of the primary models A, B and C . the best sequence was picked as CCCBBBAAA.

The graph below shows GA being run against 100 and 1000 number of generations for the existing assembly line:

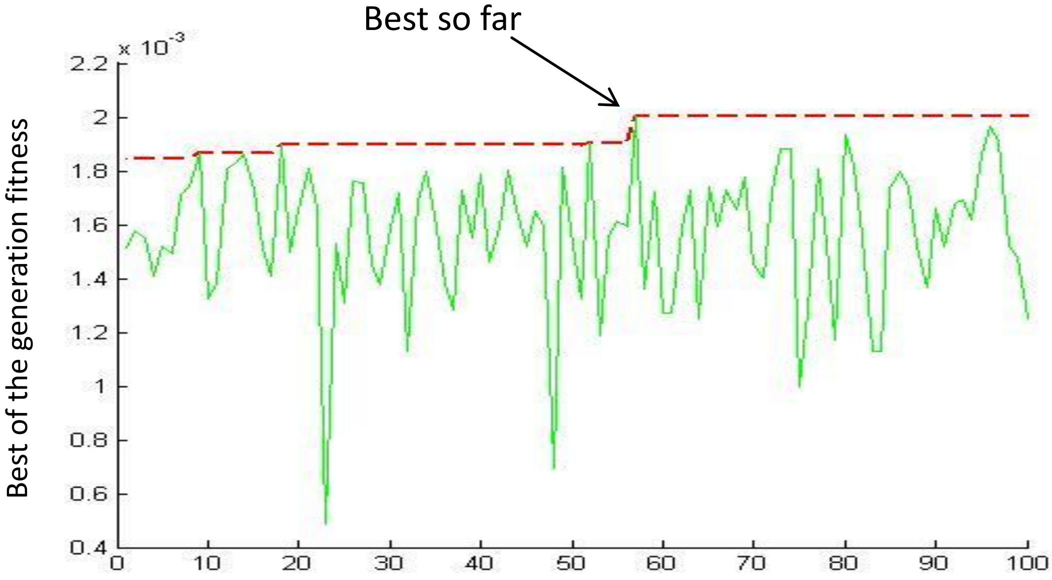


Figure 36. Simulation of GA, 100 - Generations

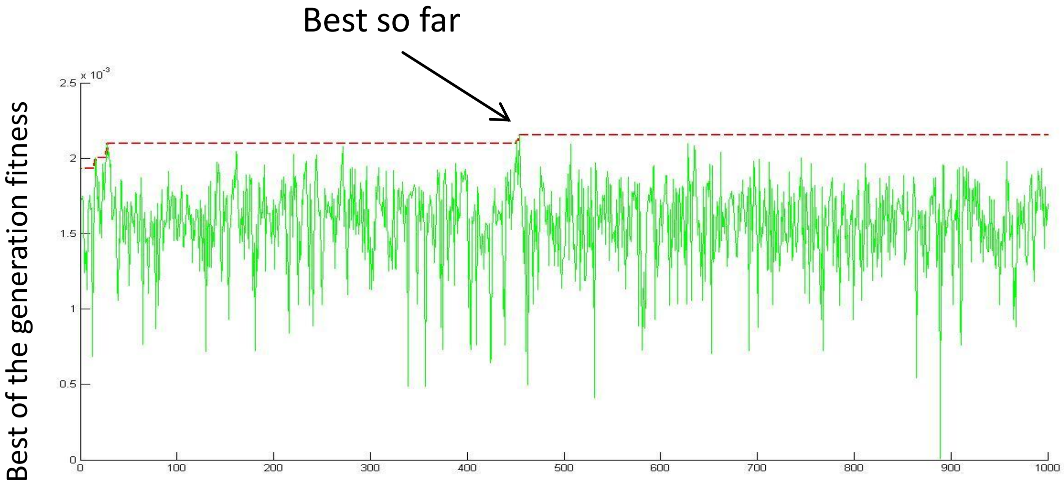


Figure 37. Simulation of GA, 1000 - Generations

6.3 METHOD TO GENERATE THE BEST SEQUENCE

The method to generate a selected sequence is as follows. First, a base sequence is picked from the population space generated. Then, permutations of the sequence are obtained which contain repetitions. These repetitions can be removed by employing a short code of picking unique sequences. Finally, out of all the unique sequences generated, the best sequence is picked against the row which displays the optimal MPS with least TUW.

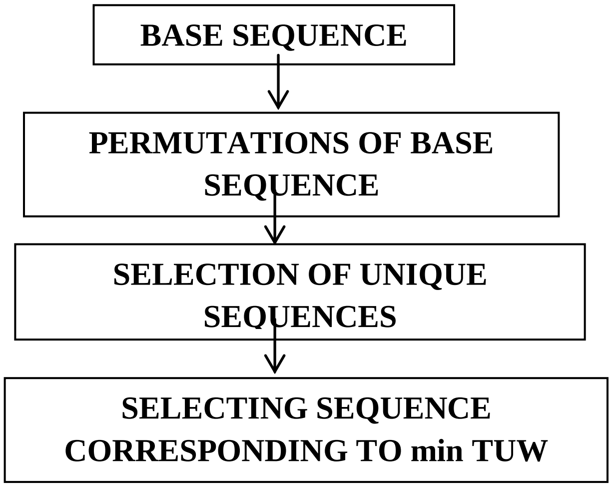
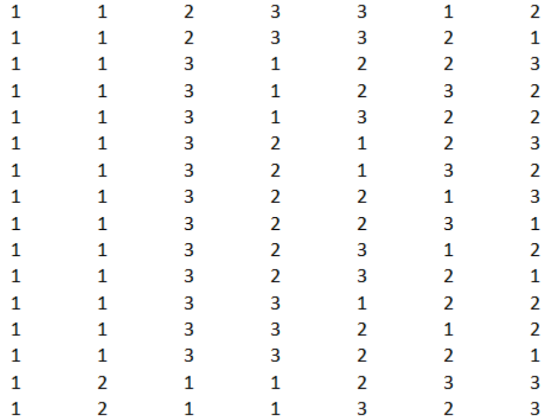


Figure 38. Procedure for selected sequence

The sample sequence snapshot is shown below. This sequence was generated for MPS = [3,2,2] where ‘1’ stands for model A, ‘2’ indicate model B and ‘3’ shows model C. For complete sequence, see Annexure A.



1	1	2	3	3	1	2
1	1	2	3	3	2	1
1	1	3	1	2	2	3
1	1	3	1	2	3	2
1	1	3	1	3	2	2
1	1	3	2	1	2	3
1	1	3	2	1	3	2
1	1	3	2	2	1	3
1	1	3	2	2	3	1
1	1	3	2	3	1	2
1	1	3	2	3	2	1
1	1	3	3	1	2	2
1	1	3	3	2	1	2
1	1	3	3	2	2	1
1	2	1	1	2	3	3
1	2	1	1	3	2	3

Figure 39. Sample sequence

6.4 APPLICATION OF PROPOSED ALGORITHM FOR PROPOSED SETUP

Figure 40 shows the layout of proposed setup. The available 52 meter space has been fully utilized for assembly line for eight stations, out of which five stations are assembly stations, two are disassembly stations and one is hull repair station. The two disassembly stations have been incorporated inside the APC – repair shed to utilized the overhead crane facility for disassembly of the heavy / major parts of the APC. Number of factors was considered to make the decision of changing the existing assembly line to the proposed layout. It includes the precedence relationship among the tasks at different assembly stations. Spacing constraints were also considered to optimize the number of stations to utilize the space for leveling the work load among optimized assembly stations.

The variables initialized J, Lj, Vc, M etc for proposed setup have been mentioned at table 9.the task time of each model (A, B, C) for five assembly stations have been calculated and mentioned at table 11. Grouping of the tasks was carried out and mentioned at table 10. The sub – details of all the stations tasks of proposed setup have been mentioned at tables 13, 14, 15, 16, 17, 18, 19and table 20. The tables 16 to table 20 give the detail of tasks of proposed assembly stations. Figure 41 shows the precedence diagram of proposed five assembly stations and the tasks times which have been mentioned at same figure are of model A. table 12 shows the precedence relationship matrix of the five assembly stations. Similarly the assembly line can be balanced for other models of APC. As the focus of this study is the sequencing of MMAL so it was assumed that the assembly line is balanced.

The above mentioned data was given to the proposed algorithm (based on the assumptions previously mentioned), it produced the results which have been mentioned at table 21.

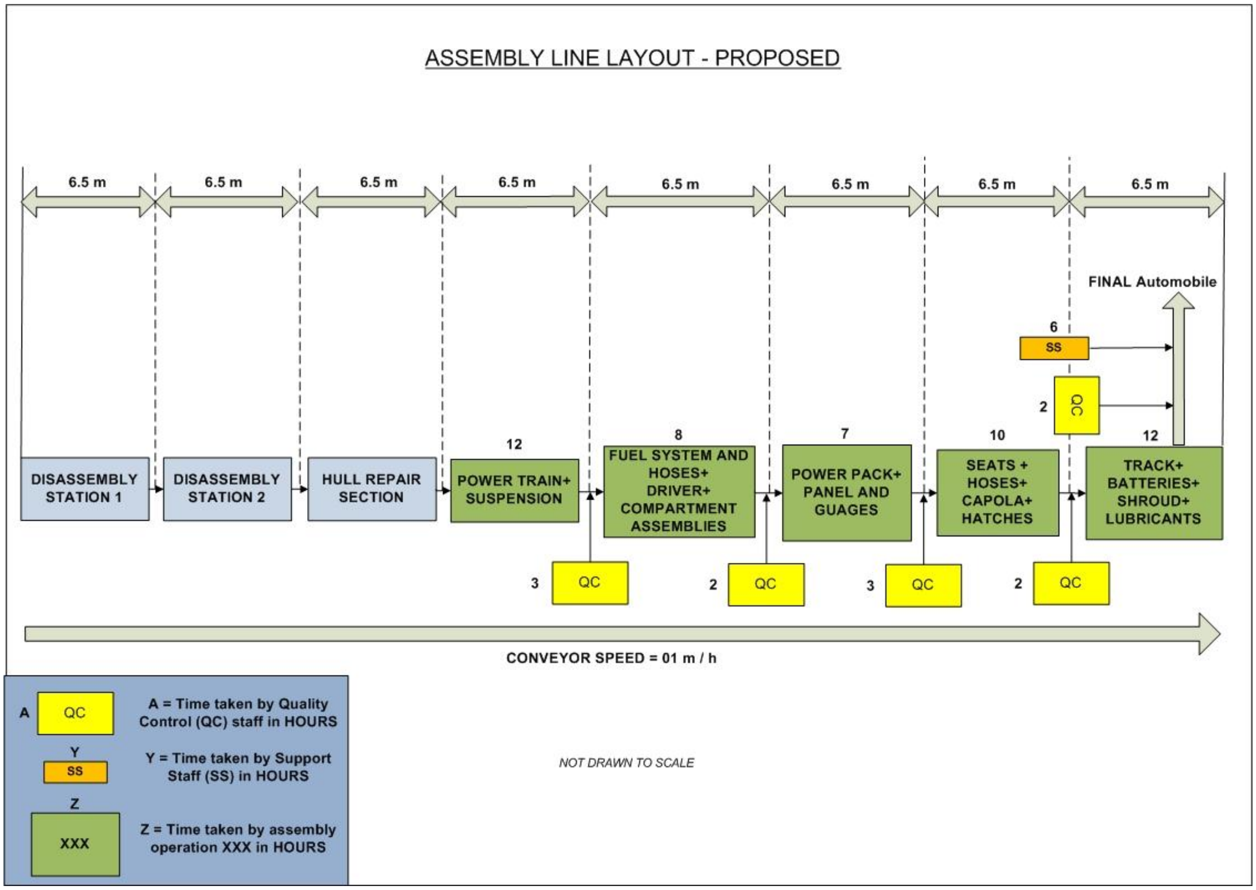


Figure 40. Proposed Assembly Line Layout

S. No.	**Variable**	**Initialized value (Proposed)**
1	Number of Stations (J)	5
2	Length of Stations (L_j)	6.5 m
3	Conveyor speed (v_c)	0.49 m / h
4	Primary Models (M)	3

Table 9. Variables for proposed setup

S. No.	Assembly Stations	Tasks
1	Station 1	Suspension and power train
2	Station 2	Driver’s compartment assemblies and fuel system
3	Station 3	Power pack, panels and gauges
4	Station 4	Hoses, seats, cupola and hatches
5.	Station 5	Track, batteries, shroud and lubricants

Table 10. Proposed Assembly Stations and their tasks

S. No.	Assembly Stations	Model A	Model B	Model C
1	Station 1	15	17	17
2	Station 2	10	12	12
3	Station 3	10	10	11
4	Station 4	12	13	13
5.	Station 5	20	20	21

Table 11. Task Times - Proposed Assembly Station

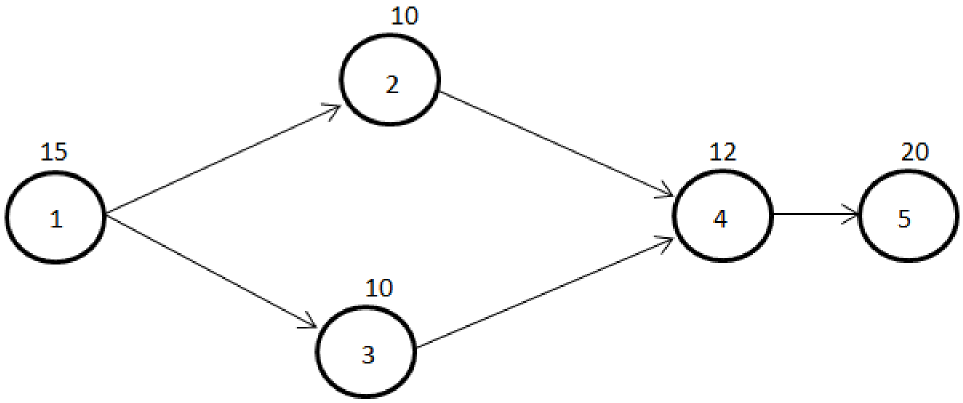


Figure 41. Precedence diagram - proposed setup

	Task station 1	Task station 2	Task station 3	Task station 4	Task station 5
Task station 1	0	1	1	0	0
Task station 2	-1	0	0	1	0
Task station 3	-1	0	0	1	0
Task station 4	0	-1	-1	0	1
Task station 5	0	0	0	-1	0

Table 12. Precedence relationship matrix – proposed setup

The following tables shows the Station wise details of all the tasks of eight stations including five assembly stations, three other stations (2 x disassembly stations and a hull repair station) inside the APC repair shed.

S/No	Description
1	Remove Batteries connections
2	Clean Vehicle
3	Remove Tracks
4	Remove Batteries
5	Drain oil Fuel & Water
6	Remove Shroud
7	Discharge Fixed Fire Extinguisher
8	Remove Ramp Door Assembly

9	Remove Access Covers
10	Remove All Lights & Horn
11	Remove Rear Fire Extinguisher
12	Remove Periscopes
13	Remove Cupola & Seats
14	Remove Air Cleaner
15	Remove Grill Assembly

Table 13. Tasks of Disassembly Station 1

S/No	Description
1	Remove Power Pack Assembly
2	Remove Driver Compartment Control
3	Remove Front Bilge Pump
4	Remove Wiring Harness
5	Remove Ramp Control Assembly
6	Remove Rear Bilge Pump
7	Remove Fuel Tank
8	Remove Battery Box Components
9	Remove Control Differential Assembly
10	Remove Final Drives
11	Remove Idler Wheels
12	Remove Track Adjuster
13	Remove Idler Supports
14	Remove Balance Arms
15	Remove Shock Absorbers And Torsion Bars

Table 14. Tasks of Disassembly Station 2

S/No	Description
1	Threading & Taping
2	Replacement of u/s Welded Nuts and screws
3	Replacement and welding of All unserviceable Brackets
4	Replacement of All unserviceable Engine Mounts
5	Welding of Cupola Brackets
6	Grinding of Welding Areas
7	Removal of Broken Screws
8	Repairing of Battery Box
9	De-burring of Hull
10	Replacement of unserviceable Track Mounts

Table 15. Tasks of Hull Repair Station 3

S/No	Description
1	Torsion Bar
2	Idler Arm Assembly
3	Final Drive
4	Sprocket And Final Drive
5	Center Support Assys (Right)
6	Front & Rear Support Assys (Right)
7	Center Support Assys (Left)
8	Front & Rear Support Assys (Left)
9	Torsion Bar
10	Shock Absorbers
11	Track Adjuster
12	Idler Wheel
13	Road Wheel

Table 16. Tasks of Assembly Station 1

S/No	Description
1	Fuel Tank And Bracket
2	Bracket And Fuel Hoses
3	Guard And Cover
4	Valve And Pump Assy
5	Ramp Cylinder
6	Ramp Latch Handle
7	Harness Dvr Compartment
8	Floor Plates And Cushion
9	Selector And Arm Assembly
10	Arm And Paddle Assembly
11	Harness

Table 17. Tasks of Assembly Station 2

S/No	Description
1	Power Pack Assembly
2	Wiring Harness
3	Control Cables
4	Front and rear Bilge Pump
5	Air Filter Indicator
6	Voltage Regulator
7	Wiring Panel
8	Main Panel
9	Left Rear Tail Light
10	Right Rear Tail Light
11	Fire Extinguisher
12	Control Differential
13	Muffler Assembly

Table 18. Tasks of Assembly Station 3

S/No	Description
1	Accelerator Assembly
2	Tachometer Adopter
3	Starter Ground Lead
4	Differential Hose
5	Differential Drive Shaft
6	Control Valve Assembly
7	Exhaust Pipe
8	Final Shaft
9	Grill Assembly
10	Fan Belt And Hose
11	Air Cleaner Hose
12	Horn And Exterior Light
13	Hatch And Cupola
14	Box Assembly
15	Fitting of Seat

Table 19. Tasks of Assembly Station 4

S/No	Description
1	Power Plant Door Extension Assembly
2	Trim Van Assembly
3	Extension Assembly
4	Final Drive Cover
5	Bty Box
6	Batteries
7	Batteries Terminal
8	Ramp Seals
9	Ramp Door Assembly
10	Engine Panel
11	Rear Engine Panel
12	Spacer And Cover

13	Adjust Stop Light
14	Hull Plugs And Cover
15	Track And its Adjustment
16	Commander Seat
17	Plate Form And crew Seat
18	Right Track Shroud
19	Left Track Shroud
20	Check Oil
21	Fill Cooling Sys
22	Fill Tank With Diesel
23	Fill Pivot Master Cyl
24	Apply Grease to Vehicle

Table 20. Tasks of Assembly Station 5

6.5 RESULTS FOR PROPOSED SETUP

S. No.	**Generations**	**MPS**	**Sequence Generated**	**Total Utility Work (Hours)**	**Minimized Utility Work per product (Hours)**
1	1	[1 1 1]	CBA	174.26	58.08
2	10	[2 2 2]	AACCBB	140.25	23.37
3	50	[2 2 2]	AABCCB	139.22	23.21
4	70	[2 2 2]	AABBCC	139.22	23.21
5	100	[3 3 3]	AAABCBCCB	113.26	12.58
6	500	[3 3 3]	AAABCBCBC	113.26	12.58
7	1000	[3 3 3]	AAACBBCCB	113.26	12.58
8	5000	[3 3 3]	AAABCBCCB	113.26	12.58
9	10000	[3 3 3]	AAACCBBCB	113.26	12.58
10	50000	[3 3 3]	AAACCCBBB	113.26	12.58

Table 21. Results of GA for - proposed setup

GA was run from one to fifty thousand generation (50,000) and the output was recorded at table 21. The results of proposed set up shows that the minimized total utility work was dropped from 630.65 hours (70.07 hours per product) to 113.26 hours (12.58 hours per product) along with the best sequence generated. Serial 5 to serial 10 of the table 21 shows that the minimum TUW remains the same, against different best sequences generated, when GA was run up to 50, 000 generations. This gives the flexibility to the assembly line to run different selected sequences to produce the same minimum TUW.

The graphs for 100 generations and 500 generations of the proposed setup have been shown at figures 42 and 43 respectively. The graphs show that the number of peaks starts increasing as ‘the best fitness value of the generations’ start increasing. In figure 42, the first peak is at 4th generation with 6.5E-3 (fitness value of the generation) and it is improved to 9.9E-3 (fitness value) at 88th generation. When the GA code was run for 500 generations (figure 43) there are more number of peaks indicating further improvement in the fitness value. Finally the GA decodes this value and displays in terms of minimum TUW.

The graph below shows GA being run against 100 and 500 number of generations for the proposed assembly line:

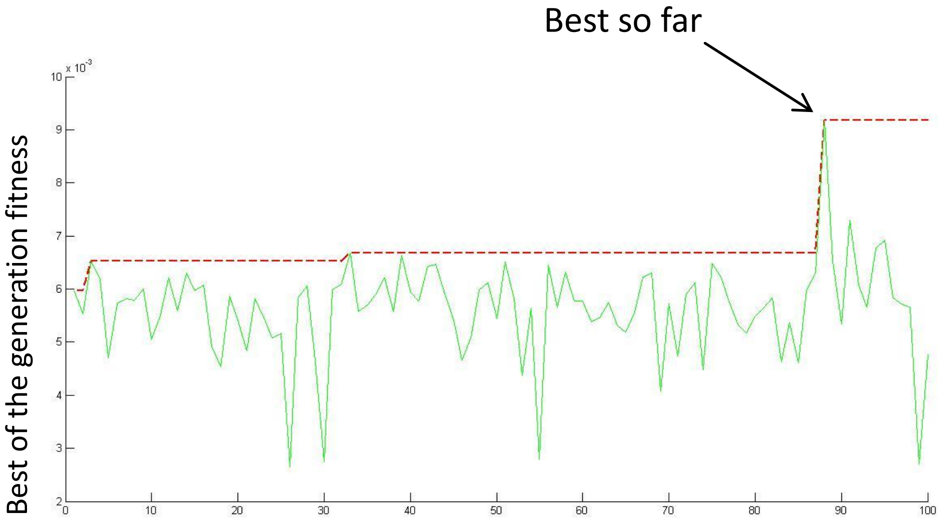


Figure 42. Simulation of GA, 100 – Generations

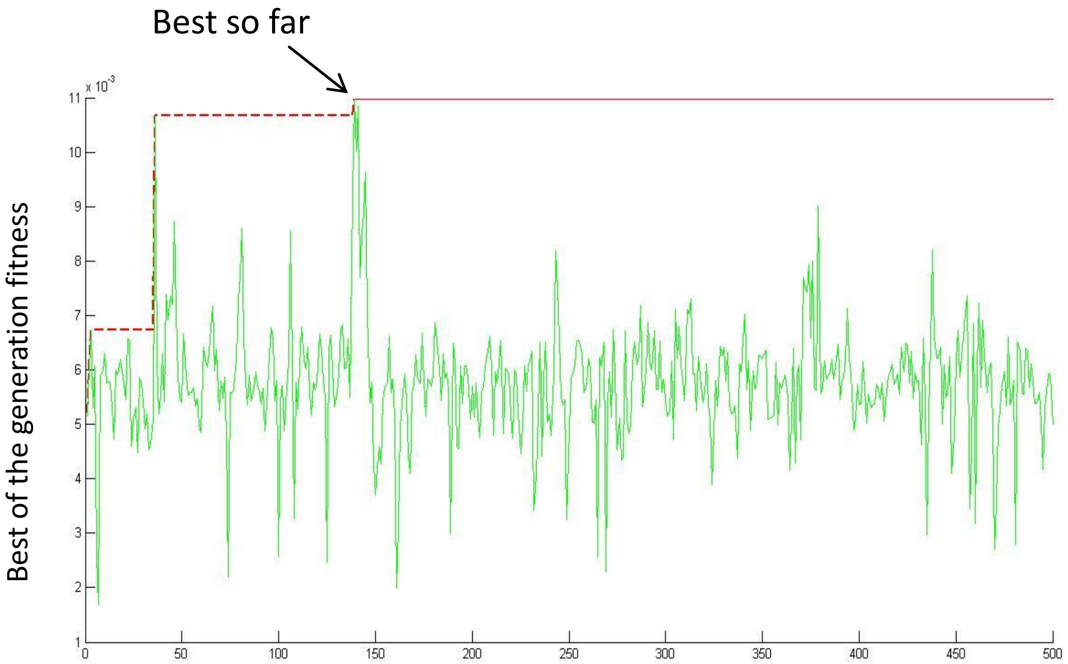


Figure 43. Simulation of GA, 500 - Generations

When we consider model A for the existing assembly line the total assembly hours are 107 hrs and min TUW is 70.07 hrs. So the total times comes out to be 177.07 hrs (which includes the min TUW). As the total working hours at the production facility are 6 hrs, so approx. 30 working days (177.07/6 = 29.51 working days) are required for assembly of only one APC (model A) and next APC in the line will be ready after 7 working days (keeping in view the last assembly station time). If we deduct the weekends and holidays, the left working days in a year are 245 (365-120 = 245 working days). Approximately 31 APC (245-30 = 215 and 215/7 = 30 APC) can be made ready in a year (30 +1 = 31) on the assembly line.

When we consider model A for the proposed assembly line the total assembly hours are 67 hrs and min TUW is 12.58 hrs. So the total time comes out to be 79.58 hrs (which includes the min TUW). As the total working hours at the production facility are 6 hrs, so approx. 13.2 working days (79.58/6 = 13.26 working days) are required for assembly of only one APC (model A) and next APC in the line will be ready after 3 working days (keeping in view the last assembly station time). If we deduct the weekends and holydays, the left working days in a year are 245 (365-120 = 245 working days). Approximately 78 APCs (245-13 = 232 and 232/3 = 77

APC) can be made ready in a year (77 +1 = 78). The overall percentage increase of the from 31 APCs per year to 78 APCs per year is 151.61%.

6.6 OVERALL WORK FLOW AT THE PRODUCTION FACILITY TO INDICATE THE BOTTLE NECKS

The overall work flow at the production facility is shown in the following figure.

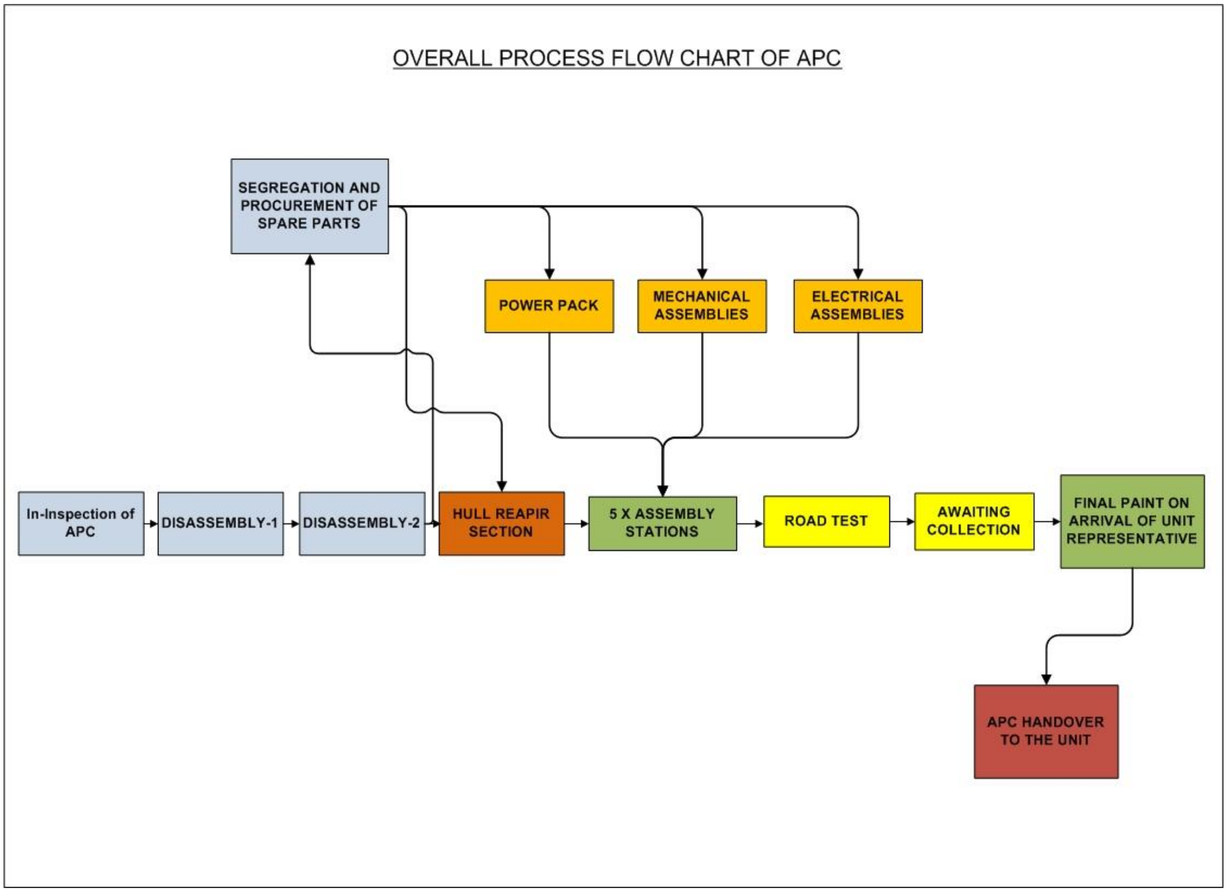


Figure 44. Overall Process Flow Chart APC

Figure 44 shows that the pre assembly and post assembly times play a vital role in the overall progress / production of the organization. Table 22 shows that the times of procurement of spares and their inspection takes a lot time which may vary from one month to three months (and even more) depending on the situation and efficiency of contractors. The awaiting collection time again may vary to a large extent. Table 22 may give a fair idea in this regard. The figure 44 and table 22 indicate that the assembly line is only a small portion of the overall setup.

A lot of factors contribute towards the overall efficiency of the organization. There are large number of bottle necks and other pre assembly line branches which need to be optimized. For example when the disassembly of APCs is carried out after the procurement stage, the efficiency of auxiliary branches start contributing toward the final assembly line. After carrying out the static and final road test of APC the ‘awaiting collection’ is marked as another bottle neck whose time varies from 15 days to 30 days (and even more). For further study the time data of above mentioned tasks have been mentioned at table 22 which may give a fair idea in identification of bottle necks in overall process flow of the organization.

S. No	Overall Flow Tasks	Time (hrs)
1.	Initial inspection of the APC	12
2.	Disassembly of the APC at two stages	12
3.	Procurement and inspection of the required spares (assuming that spare and inspection is done in one month-30 working days)	180
4.	Hull repair station	6
5.	Time consumed by one APC (Model A) at five assembly stations of the proposed setup	67
6.	Road test and fault rectification	18
7.	Awaiting collection assuming (15 days- 2 weeks)	90
8.	Final paint on the arrival of unit representative	12
9.	APC Handing over to the unit concerned	6

Table 22. Times Related to Overall Process Flow Tasks

6.7 CONCLUSION

In this study, a MMAL was optimized by minimizing total utility work and generating the most feasible global sequence in light of the demand floated by the user (which is army in this case), precedence constraints of the operations at hand, and quality control including support staff activities. The model of Hyun et. al (1998) was modified in this case. The GA was executed for first the existing MMAL setup (Figure 35) and then for the proposed MMAL (Figure 40). The best solution was obtained as per the assumptions and facility constraints.

6.8 FUTURE WORK

The proposed future work for the current research would include:

- If sudden increase in market demand is projected, the MMAL algorithm should accordingly cater for the involvement of utility workers.
- The constraint on the number of primary models can be removed for bigger problems at hand.
- Parallel MMALs can be proposed as per the increase in market demand to cater for the increased output.

Annex A

Sequence Generated for MPS [3,2,2] corresponding to models A,B and C

1	1	1	2	2	3	3
1	1	1	2	3	2	3
1	1	1	2	3	3	2
1	1	1	3	2	2	3
1	1	1	3	2	3	2
1	1	1	3	3	2	2
1	1	2	1	2	3	3
1	1	2	1	3	2	3
1	1	2	1	3	3	2
1	1	2	2	1	3	3
1	1	2	2	3	1	3
1	1	2	2	3	3	1
1	1	2	3	1	2	3
1	1	2	3	1	3	2
1	1	2	3	2	1	3
1	1	2	3	2	3	1
1	1	2	3	3	1	2
1	1	2	3	3	2	1
1	1	3	1	2	2	3
1	1	3	1	2	3	2
1	1	3	1	3	2	2
1	1	3	2	1	2	3
1	1	3	2	1	3	2
1	1	3	2	2	1	3
1	1	3	2	2	3	1
1	1	3	2	3	1	2
1	1	3	2	3	2	1
1	1	3	3	1	2	2
1	1	3	3	2	1	2
1	1	3	3	2	2	1
1	2	1	1	2	3	3
1	2	1	1	3	2	3
1	2	1	1	3	3	2
1	2	1	2	1	3	3
1	2	1	2	3	1	3
1	2	1	2	3	3	1
1	2	1	3	1	2	3
1	2	1	3	1	3	2
1	2	1	3	2	1	3
1	2	1	3	2	3	1
1	2	1	3	3	1	2
1	2	1	3	3	2	1
1	2	2	1	1	3	3
1	2	2	1	3	1	3
1	2	2	1	3	3	1

1	2	2	3	1	1	3
1	2	2	3	1	3	1

CERTIFICATE OF COMPLETENESS

It is hereby certified that the dissertation submitted by A/O Amjad Baig, Reg No. NUST201362454MCEME35113F, Titled: Global Sequence Generation and Minimization of Total Utility Work for Mixed Model Assembly Lines" has been checked/reviewed and its contents are complete in all respects.

Supervisor's Name: **Dr. Sajid Ullah Butt**

Signature: ______________________

Date: ______________________