Journal of the Nigerian Association of Mathematical Physics
Volume **25 (November, 2013), pp 251 – 266**
© J. of NAMP

Solving Assembly Line Balancing Problems: A Case Study of a Manufacturing Company

Edokpia R.O. and Okonta C. I.

Department of Production Engineering,
University of Benin, Nigeria.

Abstract

Assembly lines are flow-line production systems, where a series of workstations, on which interchangeable parts are added to a product, are linked sequentially according to the technological restrictions. The problem of assembly line balancing is a non-deterministic polynomial-time- hard optimization problem.

This paper utilises three different priority-based heuristics and Genetic Algorithm (GA) in solving assembly line balancing problem. The GA also adopts a fitness function based on realized cycle time and a crossover based on fitness ranking.

The assembly line of a production system was solved using the number of stations, line efficiency and smoothness index as the performance criteria. The objective is to minimise the number of workstations and /or to minimise the cycle time. The existing assembly line having five stations with 74.29% efficiency and a smoothness index of 5 was optimised to four stations with line efficiency of 92.86% and smoothness index of 2.

The results obtained revealed the effectiveness and high efficiency of using this genetic algorithm in solving ALBPs. The suitability in giving optimum solutions to simple assembly line balancing problem (SALBP) results from the robustness and flexibility of the genetic algorithm.

***Keywords*:** Assembly Line Balancing, Heuristic Encoded Genetic Algorithm, Realized Cycle Time.

NOMENCLATURE AND SYMBOLS

ALBP-	Assembly Line Balancing Problem
CR-	Realized cycle time
CT-	Cycle time
LE-	Line efficiency
LOT-	Longest Operation Time
GA-	Genetic Algorithm
GGA-	Grouping Genetic Algorithm
RPW-	Ranked Positional Weight
SA-	Simulated Annealing
SALBP-	Simple Assembly Line Balancing Problem
SI-	Smoothness index
ST-	Station time
m-	Total number of workstations
n -	Number of chromosomes
C_i	Parent chromosome
F_i-	Fitness of a chromosome
i-	Chromosome number
O_i	Offspring
P_c-	Crossover probability
P_i-	Selection probability
P_m-	Mutation probability

Corresponding author: ***Edokpia R.O.,*** E-mail: ralphedokpia@yahoo.com, Tel.: +2348023368811

1.0 Introduction

Assembly line is a manufacturing layout in which equipment or work processes are grouped together according to the progressive steps by which the product is made. The assembly line is defined by the sequence of steps required to make the product.

The assembly line balancing problem is a decision problem on how best tasks are to be assigned to the various workstations in order to increase efficiency. This increase in efficiency can be approximated to the minimization of the number of workstations and the maximization of the production rate. Thus, minimizing the total assembly cost while satisfying the demands and some restrictions like precedence relations among tasks and some system specific constraints.

The assembly line in which a single product is produced is referred to as the simple assembly line balancing problem (SALBP). The simple assembly line balancing problem can be classified into two types based on the objectives. When the minimization of the number of workstations along the assembly line is the concern of the assembly line balancing, the assembly line balancing problem is called the type-1 ALBP or SALBP-1. The second class known as type-2 or SALBP-2 has the objective of minimizing the cycle time for a given number of workstations thereby increasing the rate of production.

Assembly line can be deterministic or stochastic. When works are processed at a constant speed, the assembly line is said to have a deterministic task time; when there are variations in the task time, the assembly line is referred to as stochastic.

From the interview with the production manager, the production system under study has not been rebalanced since it commenced production in 1985. This study is an attempt to create a template upon which further improvement can be made through line balancing. The aim of this study is to use a combination of known heuristics and genetic algorithm to solve and optimize assembly line balancing problems. This GA approach uses the longest operation time, ranked positional weight and the kilbridge-wester heuristics in assigning tasks to the various workstations. This work is intended to embrace the use of models in evaluating and optimizing the performance of production systems.

The objective is to increase the line efficiency and to reduce the cycle time thereby maximize the production rate and the smoothness of the assembly line.

2.0 Literature Review

The application of genetic algorithm in solving assembly line balancing problems has become an area of great interest to most recent researchers on line balancing. Many researchers have proposed different encoding scheme and different methods of generating an initial population.

Anderson and Ferris [1] considered the application of genetic algorithm in solving assembly line balancing problem. The study underscores the importance of the correct choice of a scaling parameter and mutation rate to ensure the good performance of a genetic algorithm. Comparisons between the parallel and serial implementation of assembly line balancing using genetic algorithm were also made.

Yu and Yin [2] presented an adaptive genetic algorithm as an intelligent algorithm for the assembly line balancing. In this paper, the probability of crossover and mutation was adjusted dynamically according to the individual fitness value. The individuals with higher fitness values were assigned to lower probabilities of genetic operator, and vice versa.

Norozi et al [3] proposed a new approach of hybrid genetic algorithm-simulated annealing (GA.SA) implementation in order to meet objectives of the assembly line balancing problems. In order to check the efficiency of hybrid search techniques, a comparison was made between the results obtained by hybrid GA.SA and GA and this comparison validates the effectiveness of their approach.

Razali and Geraghty [4] adopted the biologically inspired evolutionary computing tool which is genetic algorithm to solve assembly line balancing problem with the objective of minimizing the idle time in the workstation. The key issue in this paper was how to generate a feasible sequence of task which does not violate the precedence constraint. In order to generate only feasible solution, a repairing strategy based on topological sort was integrated in the genetic algorithm procedure.

Levitin et al [5] introduced two different procedures for adapting the GA to the robotic assembly line balancing problem by assigning robots with different capabilities to workstations. The recursive assignment procedure and a consecutive assignment procedure were introduced. The results of the GA were improved by a local optimization (hill climbing) work-piece exchange procedure.

Chong et al [6] made a comparison between a randomly generated initial population and a heuristic treated initial population. Both populations were tested with a proposed GA using established test problems from literature. His work also showed that the GA using a fitness function based on realized cycle time is capable of generating good solutions.

Ponnambalam et al [7] proposed a multi-objective genetic algorithm to solve assembly line balancing. The performance criteria considered include the numbers of workstations, the line efficiency, the smoothness index before trade and transfer and the smoothness index after trade and transfer. The developed genetic algorithm in his work was compared with six heuristics algorithm namely, ranked positioned weight, kilbridge and wester, moodie and young, Hoffmann procedure matrix, immediate update first fit and rank and assign heuristic methods. It was better in all the performance measures than these heuristics.

The purpose of this study is to utilise a heuristic encoded GA which combine three heuristics in solving ALBPs. The performance measures used in evaluating this GA are the number of stations, line efficiency and smoothness index. From the basis of these criteria, the GA is able to give optimum solutions for assembly line balancing problems within a reasonable implementation time.

3.0 Methodology

In this paper, the single model assembly line balancing problem is considered and the proposed genetic algorithm is described as follows:

a) Representation of chromosomes

The genetic algorithm proposed adopts the heuristic based encoding system. A chromosome is represented by any sequence of heuristics which are used in assigning task to the various workstations following the workstation oriented approach. In this approach, tasks are assigned to various workstations as long as the total station load does not exceed the prescribed cycle time.

The heuristics adopted in this GA are the longest operation time heuristic, the ranked positional weight (RPW) technique and the kilbridge-wester heuristic.

b) Random generation of initial population

This procedure involves the generation of random chromosomes. A chromosome is a feasible solution with its length determined by the number of tasks and genes used to represent the heuristics used.

A population size of *2n to 4n* is usually taken as initial population size [8], where n is the number of task.

c) Evaluation of fitness function

The fitness function used in this paper is the line efficiency.

Line Efficiency (LE) is the ratio of cumulative station time to the cycle time multiplied by the number of work stations. It shows the percentage utilization of the line [6]. To maximize the line efficiency we incorporate the realized cycle time instead of the prescribe cycle time. The realized cycle time is the maximal station time after the task assignment process. According to [6],

$$\text{Line efficiency (LE)} = \frac{\sum_{i=1}^{m} ST_i}{m \times CR} \quad (1)$$

where: ST =Station time is duration of station

m =Total number of workstations

CR=Realized cycle time

i= Chromosome number.

The smoothness index for each chromosome was evaluated. A smoothness index of 0 indicates a perfect balance [9].

$$\text{Smoothness index (SI)} = \sum_{i=1}^{m} \sqrt{(ST_{max} - ST_i)^2} \quad (2)$$

d) Selection

After the evaluation of the chromosomes, some of the chromosomes are chosen in order to create the next generation. In this paper, 80% of the total population is selected randomly [10] with a selection probability given as

$$P_i = \frac{F_i}{\sum_{i=1}^{n} F_i} \quad [8] \quad (3)$$

Where:

P_i= selection probability

F_i = Fitness of a chromosome

n = Number of chromosomes

i= Chromosome number

The worst 20% are allowed to die. This method is similar to the roulette wheel selection procedure and is based on the theory of the survival of the fittest [10].

e) Crossover

The selected chromosomes are ranked and paired according to their efficiency and smoothness index (the first two fittest chromosomes are paired). This GA adopts the two point order crossover operation where two points are randomly chosen and the genetic material between them is swapped to give two offspring [11]. The crossover probability is assumed to be unity, that is P_c=1.0

f) Mutation

The mutation operator has the effect of creating a new offspring which cannot be created by the ordinary crossover operator. After crossover has been applied, we apply the mutation operator based on a mutation probability (P_m say 0.2). We use the scrambled mutation operator in which two points are randomly selected and scramble the elements within it [6].

After the new offspring are created with the crossover and mutation operators, the successor generation is generated with all the new offspring and 20% of the preceding population selected as pareto optimal solutions based on their fitness values. This concept of replacement is referred to as the elitism strategy.

g) Termination

This GA procedure can be repeated as many times as desired. It will be terminated after the prescribed numbers of generations has been completed say 50 and if after ten (10) successive generations, no improvement was realized.

4.0 The Heuristics Used.

The three heuristic algorithms combined and recombined are explained below.

A. Longest operation time technique (LOT)

- **i.** Construct the precedence diagram from the precedence table
- **ii.** Arrange the task in descending order of their task time i.e. from the longest to the shortest.
- **iii.** Assign the longest operation first while maintaining precedence and cycle time restriction. [12].

B. The ranked positional weight technique (RPW).

- **i.** Construct the precedence diagram
- **ii.** Determine the positional weight of each task. The positional weight of a task is the summation of the task time and the processing times of all its successors.
- **iii.** Rank the tasks in descending order based on the positional weight. i.e. from the highest PW to the lowest PW
- **iv.** Assign the task with the highest RPW first and proceed in that manner maintaining precedence and cycle time restrictions.[9]

C. The Kilbridge-Wester heuristic.

- **i.** Construct the precedence diagram.
- **ii.** From the precedence diagram, list in column I all tasks without precedence. In column II list all tasks that have those in column I as their immediate precedence. Continue to the other columns in the same way.
- **iii.** Assign task to the workstations starting with column I and continue while maintaining cycle time restriction [13].

The proposed genetic algorithm combined and recombined these heuristics.

5.0 Manufacturing Scenario:

The company under study, located in Delta State, Nigeria started production in 1985. The raw materials used for the manufacturing of flexible polyurethane foam are grouped into two namely;

i. The major or primary raw materials

ii. The minor or secondary raw materials

The major or primary raw materials are chemicals without which flexible polyurethane foam production will not be possible and these chemicals include; polyol, toluene di-isocyanate (TDI) and water.

Polyol is an organic chemical which belongs to a tertiary class of alcohol. Before production starts, it should be chilled to a temperature range of 22 – 25^{0}C

Toluene di-isocyanate is a clear, almost colourless, low viscous and toxic liquid with a characteristic pungent smell. It forms the back bone of the chain reaction mechanism of flexible polyurethane foam. Prior to production, it is chilled to a temperature range of 20 – 25^{0}C by a chilling machine.

Water is the main blowing agent in flexible polyurethane foam production.

The minor raw materials are those chemicals without which foam production and reactions can take place. They are usually called stabilizers and activators. These chemicals include; silicone oil, amine, stannous octate. Other additives include; methylene chloride (auxiliary blowing agent), colourant and fillers. When auxiliary blowing agent is to be used, they must be cooled below their boiling point.

After these required chemicals had undergone the several reactions at the production stage in a functional layout to give the actual foam blocks, several tasks are performed on the blocks in an assembly line in order to have the final product 'mattress'. The series of tasks performed on the foam block after the initial production process is known as foam conversion. The flow diagram of the production system is shown in Figure. 1

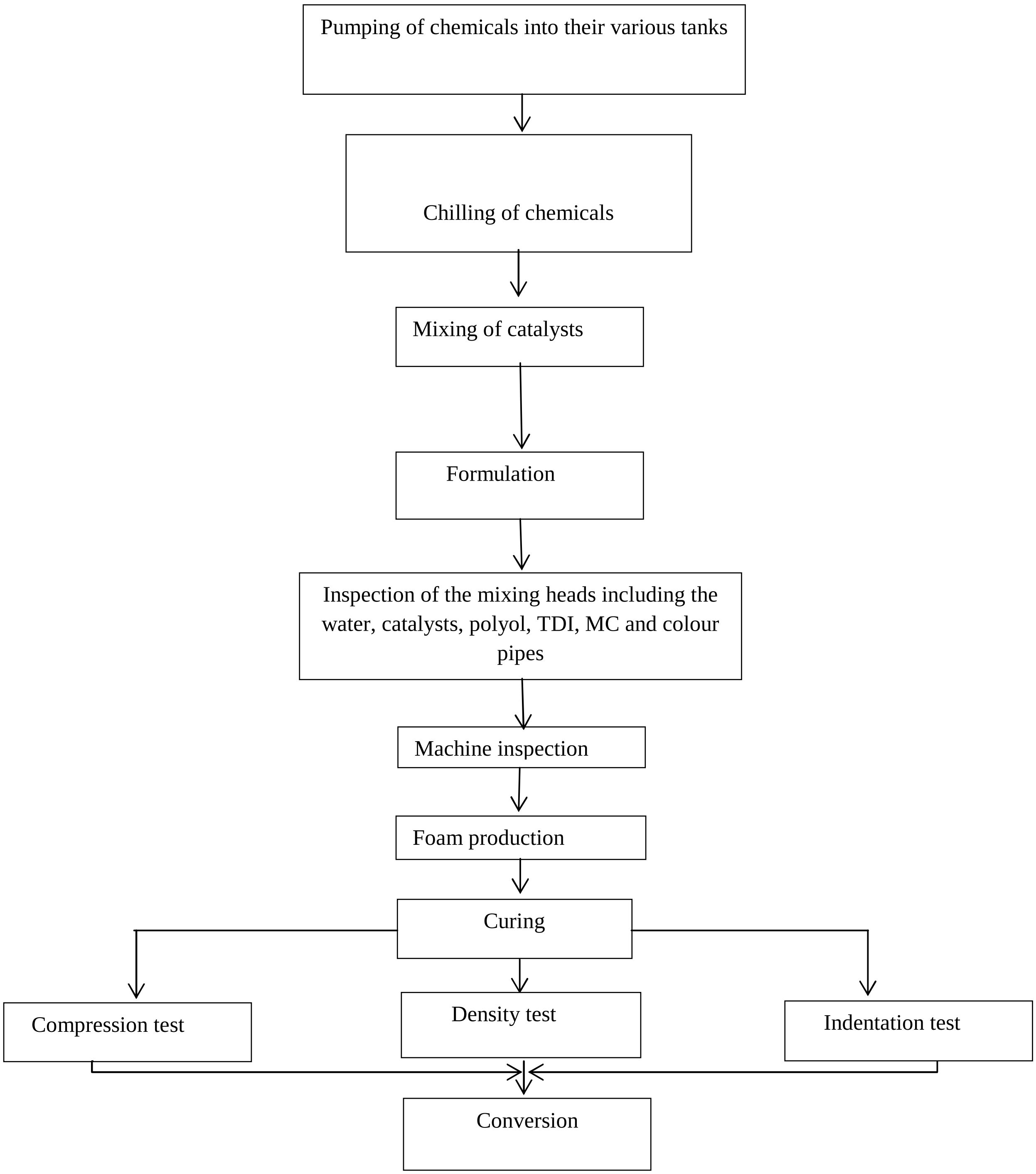

Figure1: The Flow Diagram of the Production System

Table 1 shows the outline of the various tasks performed during the conversion process.

Table 1.Summary of tasks involved

Task	Task description	Task number	Task time (minutes)
Block sizing	Measurement of the actual size of a block and marking out of the desired size of mattress.	1	1
Cover cutting	Cutting the bed cover to the required size.	2	4
Vertical slicing	Cutting the block to the desired breadth and length.	3	2
Corner sewing	Forming a rectangular cover for the four sides of the mattress	4	3
Horizontal slicing	Cutting the block to the desired height	5	5
Rubbing	Rubbing the entire bed with an adhesive	6	4
Bed covering	Covering the bed with bed covers	7	2
Mattress sewing	Sewing the mattress to the bed covers	8	5

The production time available per day=8hrs=480mins
Desired units of output=70 units
Actual units of output varies between 50-60 units

Cycle time (CT) $= \frac{\text{Production time (mins)}}{\text{Desired units output}} = \frac{480\text{ (mins)}}{70} = 6.86 \approx 7$mins.

The minimum number of stations = $\frac{\text{Total task time (mins)}}{\text{Cycle time (mins)}} = \frac{26}{7}$ =3.714 i.e. 4 stations.

Assumptions

1. The foam blocks are in good condition
2. Only a particular size of mattress is produced with dimensions *72×64×22* inches
3. There is no accident or machine breakdown.

This illustrative problem with 8 tasks and a cycle time of 7mins is described in the precedence table shown below. The precedence diagram is shown in Figure 2

Table 2.The precedence table (CT) = 7 Minutes

Task Number	Task Time (minutes)	Immediate Predecessor Task
1	1	-
2	4	-
3	2	1
4	3	2
5	5	1,3
6	4	1,3,5
7	2	1,2,3,4,5,6
8	5	1,2,3,4,5,6,7

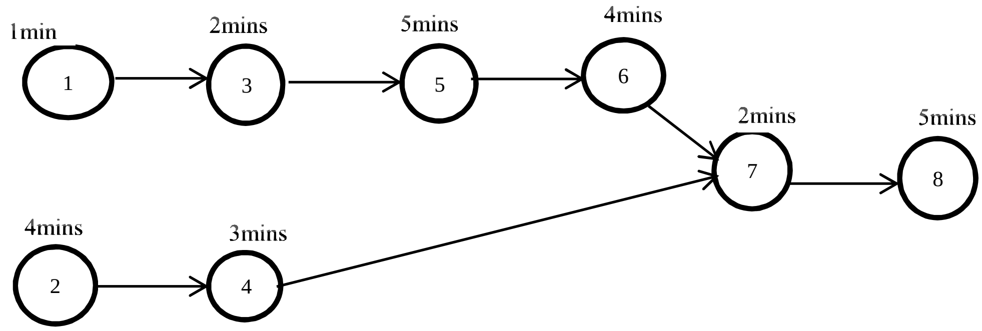

Figure 2.Precedence Diagram

The existing assembly balance is shown in Table 3

Table 3.The existing assembly line balance

Station i	Task No.	Task Time	Station time (ST_i)	$ST_{max} - ST_i$
1	1 3	1 2	3	4
2	5	5	5	2
3	2 4	4 3	7	0
4	6 7	4 2	6	1
5	8	5	5	2

Line efficiency (LE) = $= \frac{\sum_{i=1}^{m} ST_i}{m \times CR} = \frac{(3+5+7+6+5)}{5\times7} \times 100 = 74.29\%$

Smoothness index (SI) = $\sum_{i=1}^{m}\sqrt{(ST_{max} - ST_i)^2}$ = $\sqrt{(4^2 + 2^2 + 1^2 + 2^2)}$ = 5

This illustration problem is now solved using the three heuristics and the proposed genetic algorithm.

Using longest operation time heuristic

Table 4 shows the task ranking based on their operation time and the allocation of tasks to the different workstations, the line efficiency and the smoothness index using the LOT heuristic is shown in Table 5

Table 4.Ranking of the tasks using the LOT heuristic

Task	Time	Rank
5,8	5	1
2,6	4	2
4	3	3
3,7	2	4
1	1	5

Table 5.Assembly line balance using LOT

Station i	Task No.	Task Time	Station time (ST_i)	$ST_{max} - ST_i$
1	2 4	4 3	7	0
2	1 3	1 2	3	4
3	5	5	5	2
4	6 7	4 2	6	1
5	8	5	5	2

Line efficiency (LE) = 74.29% and Smoothness index (SI) =5

Using ranked positional weight

Table 6 shows the task ranking based on their positional weight and the allocation of tasks to the different workstations, the balanced line is shown in Table 7

Table 6.The positional weight and rank

Task	Positional Weight	Rank
1	19	1
3	18	2
5	16	3
2	14	4
6	11	5
4	10	6
7	7	7
8	5	8

Table 7.Assembly line balancing using RPW

Station i	Task No.	Task Time (minutes)	Station time (ST_i)	$ST_{max} - ST_i$
1	1 3	1 2	3	4
2	5	5	5	2
3	2	4	4	3
4	6 4	4 3	7	0
5	7 8	2 5	7	0

Line efficiency (LE) =74.29% and Smoothness index (SI) =5.39

Using Kilbridge-Wester heuristic

Table 8 shows the task ranking based on their precedence and the allocation of tasks to the different workstations, the balanced line is shown inTtable 9

Table 8.Ranking of the tasks using Kilbridge-Wester heuristic

Column	Task	Rank
I	1,2	1
II	3,4	2
III	5	3
IV	6	4
V	7	5
VI	8	6

Table 9.Assembly line balancing using kilbridge-wester

Station i	Task No.	Task Time (minutes)	Station time (ST_i)	$ST_{max} - ST_i$
1	1 2 3	1 4 2	7	0
2	4	3	3	4
3	5	5	5	2
4	6 7	4 2	6	1
5	8	5	5	2

Line efficiency (LE) =74.29% and Smoothness index (SI) =5

USING THE PROPOSED GENETIC ALGORITHM
We have the following steps;

Step 1: representation of chromosome.
The chromosome is a string of 8 genes corresponding to the 8 tasks.
Let; Gene A be task assigned using heuristic A,
Gene B be task assigned using heuristic B,
Gene C be task assigned using heuristic C
Hence, the heuristic encoding scheme.

Step2: random generation of initial population.
Usually, the population size ranges from *2n to 4n* [8]. In this illustration,
the population size is taken to be 2n+4 where n= no of tasks. Therefore, population size= (2×8) +4=20. The randomly generated initial population with 20 chromosomes is shown in Table 10.

Table 10.The randomly generated initial population

Chromosomes	Genes							
	1	2	3	4	5	6	7	8
1	C	B	C	A	A	B	B	C
2	C	C	B	A	C	B	A	A
3	A	A	B	A	B	C	A	B
4	B	B	C	C	C	A	A	A
5	B	A	C	C	A	A	B	B
6	A	B	B	A	B	B	A	A
7	A	C	A	A	C	B	A	C
8	B	A	C	B	C	B	B	B
9	B	B	A	B	B	A	A	C
10	B	A	A	B	B	C	C	A
11	C	B	A	B	B	A	C	A
12	C	C	B	A	B	C	B	C
13	A	C	B	A	A	A	A	B
14	A	C	B	A	B	C	A	C
15	B	C	A	C	B	A	C	A
16	C	B	B	B	C	C	C	B
17	C	C	B	A	B	C	A	C
18	C	C	A	B	C	C	B	A
19	C	B	A	A	A	A	B	B
20	B	B	A	B	A	A	B	B

Step 3: evaluation of objective and fitness function.

Calculate the line efficiency and the smoothness index using equations 1 and 2. Table 11 indicates number of stations, line efficiency, the smoothness index and rank of each chromosome in the initial population.

Step 4: Selection

Using the pareto approach, select 80% of the 20 chromosomes i.e. 16 chromosomes randomly based on fitness using the selection probability using equation (3). Table 12 shows the selection probability for each chromosome in the population

Table 11.The objective function and fitness values

chromosome	No of station	Line efficiency	Smoothness index	Rank
1	4	92.86	2	1
2	4	92.86	2	1
3	5	74.29	5	3
4	5	74.29	5	3
5	5	74.29	5	3
6	4	92.86	2	1
7	5	86.67	2	2
8	4	92.86	2	1
9	5	74.29	5.39	4
10	5	86.67	2	2
11	5	74.29	5.39	4
12	4	92.86	2	1
13	4	92.86	2	1
14	4	92.86	2	1
15	5	86.67	2	2
16	5	74.29	5.39	4
17	4	92.86	2	1
18	5	86.67	2	2
19	5	74.29	5.39	4
20	5	74.29	5.39	4

Table 12.The selection probabilities

chromosome	Line efficiency	Selection probability	Cumulative probability
1	92.86	0.055	0.055
2	92.86	0.055	0.11
3	74.29	0.044	0.154
4	74.29	0.044	0.198
5	74.29	0.044	0.242
6	92.86	0.055	0.297
7	86.67	0.051	0.348
8	92.86	0.055	0.403
9	74.29	0.044	0.447
10	86.67	0.051	0.498
11	74.29	0.044	0.542
12	92.86	0.055	0.597
13	92.86	0.055	0.652
14	92.86	0.055	0.707
15	86.67	0.051	0.758
16	74.29	0.044	0.802
17	92.86	0.055	0.857
18	86.67	0.051	0.908
19	74.29	0.044	0.952
20	74.29	0.044	1

Table 13.The Crossover of the First Generation

Chromosomes	Genes							
	1	2	3	4	5	6	7	8
C_1	C	B	**C**	**A**	**A**	B	B	C
C_2	C	C	**B**	**A**	**C**	B	A	A
O_1	C	B	**B**	**A**	**C**	B	B	C
O_2	C	C	**C**	**A**	**A**	B	A	A
C_6	A	B	**B**	**A**	**B**	B	A	A
C_8	B	A	**C**	**B**	**C**	B	B	B
O_3	A	B	**C**	**B**	**C**	B	A	A
O_4	B	A	**B**	**A**	**B**	B	B	B
C_{12}	C	C	**B**	**A**	**B**	C	B	C
C_{13}	A	C	**B**	**A**	**A**	A	A	B
O_5	C	C	**B**	**A**	**A**	C	B	C
O_6	A	C	**B**	**A**	**B**	A	A	B
C_{14}	A	C	**B**	**A**	**B**	C	A	C
C_{17}	C	C	**B**	**A**	**B**	C	A	C
O_7	A	C	**B**	**A**	**B**	C	A	C
O_8	C	C	**B**	**A**	**B**	C	A	C
C_5	B	A	**C**	**C**	**A**	A	B	B
C_7	A	C	**A**	**A**	**C**	B	A	C
O_9	B	A	**A**	**A**	**C**	A	B	B
O_{10}	A	C	**C**	**C**	**A**	B	A	C
C_{10}	B	A	**A**	**B**	**B**	C	C	A
C_{15}	B	C	**A**	**C**	**B**	A	C	A
O_{11}	B	A	**A**	**C**	**B**	C	C	A
O_{12}	B	C	**A**	**B**	**B**	A	C	A
C_{18}	C	C	**A**	**B**	**C**	C	B	A
C_3	A	A	**B**	**A**	**C**	B	A	A
O_{13}	C	C	**B**	**A**	**C**	C	B	A
O_{14}	A	A	**A**	**B**	**C**	B	A	A
C_4	B	B	**C**	**C**	**C**	A	A	A
C_9	B	B	**A**	**B**	**B**	A	A	C
O_{15}	B	B	**A**	**B**	**B**	A	A	A
O_{16}	B	B	**C**	**C**	**C**	A	A	C

Step 5: Crossover.
A two point crossover is applied to the two selected chromosomes using a crossover probability P_c=1.0 based on their fitness value. In this crossover, we randomly select two points (in this case the 3rd and 5th gene) on the parent chromosomes and exchange the genetic materials between the points (swap crossover). This generates two new offsprings having the genetic composition of the two parents as shown in Table 13. Parent chromosome is designated as C_i and the offspring with O_i. Applying the two point crossover to the selected chromosomes using *Pc*=1

Step 6: Mutation
Apply the scramble mutation with *Pm*=0.2. This implies 0.2×20=4th 0ffspring. Choose the two points randomly as 3rd and 5th genes and scatter the genetic component within the points as shown in Table 14

Table 14. The Mutation of the First Generation

chromosomes	genes							
	1	2	3	4	5	6	7	8
C_4	B	A	**B**	**A**	**B**	B	B	B
O_4	B	A	**B**	**B**	**A**	B	B	B

The Pareto Optimum Solutions of First Generation:

Optimum solutions of first generation are shown in Table 15.

Table 15. pareto optimal chromosomes

chromosomes	genes							
	1	2	3	4	5	6	7	8
C_1	C	B	C	A	A	B	B	C
C_2	C	C	B	A	C	B	A	A
C_6	A	B	B	A	B	B	A	A
C_8	B	A	C	B	C	B	B	B

The pareto optimal solutions realized from the initial population are shown in Tables 16 -19

Table 16.Chromosome 1 Assembly line balancing

Station i	Task No.	Task Time (minutes)	Station time ST_i	CR-ST_i
1	1 3 2	1 2 4	7	0
2	5	5	5	2
3	6 4	4 3	7	0
4	7 8	2 5	7	0

Line efficiency (LE) = 92.86% and Smoothness index (SI) =2

Table 17.Chromosome 2 Assembly line balancing

	Task No.	Task Time (minutes)	Station time ST_i	CR-ST_i
1	1 2	1 4	7	0
2	5	5	5	2
3	4 6	3 4	7	0
4	7 8	2 5	7	0

Line efficiency (LE) = 92.86% and Smoothness index (SI) =2

Table 18.Chromosome 6 Assembly line balancing

Station i	Task No.	Task Time (minutes)	Station time ST_i	CR-ST_i
1	2 1 3	4 1 2	7	0
2	5	5	5	2
3	6 4	4 3	7	0
4	7 8	2 5	7	0

Line efficiency (LE) = 92.86% and Smoothness index (SI) =2

Table 19.Chromosome 8 Assembly line balancing

Station i	Task No.	Task Time	Station time ST_i	CR-ST_i
1	1 2	1 4	5	2
2	3 5	2 5	7	0
3	4 6	3 4	7	0
4	7 8	2 5	7	0

Line efficiency (LE) = 92.86% and Smoothness index (SI) =2

The Successor Generation

The next generation after crossover and mutation is shown in Table 20.

Table 20.The successor generation

Chromosomes	Genes							
	1	2	3	4	5	6	7	8
1	C	B	B	A	C	B	B	C
2	C	C	C	A	A	B	A	A
3	A	B	C	B	C	B	A	A
4	B	A	B	B	A	B	B	B
5	C	C	B	A	A	C	B	C
6	A	C	B	A	B	A	A	B
7	A	C	B	A	B	C	A	C
8	C	C	B	A	B	C	A	C
9	B	A	A	A	C	A	B	B
10	A	C	C	C	A	B	A	C
11	B	A	A	C	B	C	C	A
12	B	C	A	B	B	A	C	A
13	C	C	B	A	C	C	B	A
14	A	A	A	B	C	B	A	A
15	B	B	A	B	B	A	A	A
16	B	B	C	C	C	A	A	C
17	C	B	C	A	A	B	B	C
18	C	C	B	A	C	B	A	A
19	A	B	B	A	B	B	A	A
20	B	A	C	B	C	B	B	B

Step 7: termination.
This GA will terminate after the prescribed stopping criteria have been met. The stopping criterion used in this illustration is the stall generations. The stall generation is the number of iterations with no improvement in the best fitness value [6] in this case, the stall generation is 10.

6.0 Results and Discussions

The results produced by the proposed genetic algorithm for the illustrative problem gave an assembly line with minimum possible number of workstations, which is four (4). By adopting the concept of realized cycle time which selects the maximum station time obtained as the actual cycle time, a cycle time of 6 minutes was obtained for existing five workstations instead of the prescribed cycle time of 7 minutes.
The result of each of the three heuristics utilised in this study as illustrated in Tables 5, 7 and 9 had shown the superiority of the adopted genetic algorithm in realizing an optimum solution for assembly line balancing problems. This is verified as the three heuristics have line efficiencies of 74.29% with five stations each while the genetic algorithm approach has the efficiency of 92.86% with four stations.

7.0 Conclusion

The genetic algorithm methodology utilised in this paper combined three different heuristics in solving assembly line balancing problems. With the concept of the realized cycle time and parent selection based on fitness ranking, the GA undergoes less iteration to obtain optimum solutions for ALBPs. The assembly line network of the production system considered was optimised from the existing five stations with an efficiency of 74.29% and smoothness index of 5 to a network having four stations with an efficiency of 92.86% and a smoothness index of 2.

References

[1] Anderson E.J and Ferris M.C (1994). Genetic algorithm for combinatorial optimization: the assembly line balancing problem. INFORMS Journal on Computing, 6, pp.161-173

[2] Yu J and Yin Y (2010). Assembly line balancing based on adaptive genetic algorithm. International Journal of Advance Manufacturing Technology 48, pp. 347–354

[3] Norozi A, Ariffin M.K.A, Ismail N and Mustapha F (2011). An optimization technique using hybrid GA-SA algorithm for multi-objective scheduling problem. Scientific Research and Essays Vol. 6(8), pp. 1720-1731

[4] Razali N. M and Geraghty J (2011). Biologically Inspired Genetic Algorithm to Minimize Idle Time of the Assembly Line Balancing. Third world congress nature and biologically inspired computing (NaBIC2011), Salamanca, Spain.

[5] Levitin G, Rubinovitz J, Shnits B (2004). A genetic algorithm for robotic assembly line balancing. European Journal of Operational Research. doi:10.1016/j.ejor.2004.07.030

[6] Chong K.E, Omar M.K and Bakar N.A (2008). Solving Assembly Line Balancing Problem using Genetic Algorithm with Heuristics- Treated Initial Population. Proceedings of the World Congress on Engineering Vol II London, U.K

[7] Ponnambalam S. G, Aravindan P, Mogileeswar Naidu G (2000) Multi-objective genetic algorithm for solving assembly line balancing International Journal Advance Manufacturing 16(5), pp. 341–352

[8] Rao S. S (2009) Engineering Optimization: Theory and Practice, Fourth Edition, John Wiley & Sons, New York.

[9] Mahto D and Kumar A (2012). An Empirical Investigation of Assembly Line Balancing Techniques and Optimized Implementation Approach for Efficiency Improvements Global Journal of Researches in Engineering Mechanical and Mechanics Engineering Volume XII Issue III version I, USA.

[10] Aydin M.A (2008); M.S.c thesis, Institute of Sciences Bahçeşehir University: Solving university course timetabling problem using genetic algorithm.

[11] Obitko M (1998). Crossover and mutation-Introduction to Genetic Algorithm.www.obitko.com/tutorials/genetic algorithm. Retrieved on 29th July, 2013.

[12] Güner M, Yücel O and Ünal C (2013). Applicability of different line balancing methods in the production of apparel. Tekstil ve konfeksiyon 23(1), pp. 77-84

[13] Ponnambalam S.G, Aravindan P and Mogileeswar G. N (1999). A Comparative Evaluation of Assembly Line Balancing Heuristics. International Journal Advance Manufacturing Technology (1999) 15, pp. 577-586