Journal of the Nigerian Association of Mathematical Physics
Volume **25 (November, 2013), pp 267 – 280**

On The Use of Heuristics and Genetic Agorithm For Solving Line Balancing Problems.

Edokpia R.O. and Okonta C. I.

Department of Production Engineering,
University of Benin, Nigeria.

Abstract

The problem of assembly line balancing is a non-deterministic polynomial-time (NP) - hard optimization problem. Some approximation algorithms for the problem have been proposed but most of them are either not optimal or too complex to apply.

This paper utilizes the combination of longest operation time, ranked positional weight and Kilbridge-Wester heuristics and finally Genetic Algorithm to solve assembly line balancing problem solved by Ponnanbalam et al (2000) in which they used 14 heuristics and GA to solve the ALBP . The GA adopts a fitness function based on realized cycle time and a crossover based on fitness ranking.

The computational effectiveness and efficiency of using genetic algorithm in solving ALBP was validated by comparison with a multi objective genetic algorithm, utilizing fourteen heuristic rules for solving simple assembly line balancing problems.

The three heuristics genetic algorithm was found to perform better, from the view point of optimization giving a line efficiency of 92.59% and smoothness index of 2.45.

***Keywords*:** Assembly Line Balancing, longest operation time technique, Ranked positional weight technique, Kilbridge- Wester heuristic, Genetic Algorithm

NOMENCLATURE AND SYMBOLS

ALB-	Assembly Line Balancing
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
C_i-	Parent chromosome
F_i-	Fitness of a chromosome
i-	Chromosome number
m-	Total number of workstations
n -	Number of chromosomes
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

An assembly line is a manufacturing process in which two or more separate tasks are fitted together in a sequential manner to form a new product. The tasks are generally interchangeable. In Operations Management the decision on how best tasks are to be assigned to the various workstations in order to increase efficiency is referred to as assembly line balancing problem (ALBP).

The variable of interest for the ALB consists of number of tasks, processing time, precedence relationships and the cycle time. The goals of the ALB are to minimize the number of workstations (m), minimize the workload variance, minimize the idle time and maximize the line efficiency [1].

The assembly line in which a single product is produced is referred to as the simple assembly line balancing problem (SALBP). Though SALBP is a class of NP-hard optimization problems; effective exact methods are available in solving small and medium-size problems [2].

Approximate methods (heuristics and metaheuristics) have been developed in order to overcome the size limitation of the exact methods aiming at providing good solutions that are as near to the optimal solution as possible [3]. Nevertheless, further algorithmic improvement is necessary for solving large-scale problems [2].

GAs are numerical optimisation algorithms inspired by both natural selection and natural genetics. The primary characters are the population search strategy, information exchanging between the individuals in the population, and the evolution process. Genetic algorithms keep a group of near-optimal solutions rather than a single-current solution, which is its greatest difference from the other meta-heuristic algorithms [4].

In order to find optimal solution to the ALB problem via GA methods, four critical elements are required. First, an appropriate representation is required. This is accomplished by representing a task sequence in terms of chromosome. Second, a fitness function is required to evaluate the quality of different potential solutions. Third, a set of genetic operators (parent selection, crossover and mutation) which generate new chromosomes as a function of older chromosomes must be defined. Finally, algorithm parameters must be decided.

The aim of this study is to reassess the line balancing problem solved by [5] in which they used 14 heuristics and GA but in this case, 3 heuristics are utilized and to make comparison of the results obtained.

The performance measures considered are; the number of workstations, the line efficiency and the smoothness index

2.0 Literature Review

Ponnambalam et al [5] proposed a multi-objective genetic algorithm to solve assembly line balancing problems. 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 their 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.

Suwannarongsri et al [1] considered a new hybrid tabu search (HTS) method for solving assembly line balancing problems in the paper. The tabu search (TS) method was combined with the genetic algorithm (GA) to identify and provide solutions for the assembly line balancing problems. From the simulation results compared with the conventional method, it was found that the proposed HTS method is capable of producing solutions superior to the conventional method. It was concluded that the HTS method is an alternative potential algorithm to solve assembly line balancing problems

Norozi et al [6] 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 validated the effectiveness of their approach.

Razali and Geraghty [7] 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 [8] 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.

Brudaru et al [9] dealt with the design of balanced assembly lines with parallel workstations in the case when the execution times are real sampled fuzzy numbers. In order to solve this problem, the paper proposed an efficient greedy algorithm that constructs an assembly structure containing both serial and parallel workstations for a prescribed confidence threshold. The greedy algorithm was grafted on a genetic algorithm resulting a powerful tool for solving this problem The performance of the hybrid genetic algorithm related to efficiency of defuzzyfication rules, optimality of the number of workstations, absolute and relative deviation from the optimal value, were experimentally analysed.

Chong et al [10] 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.

Fathi et al [11] addressed the wrong application of the well-known rank positional weight technique which may invalidate some of the conclusions in [5] who considered a multi objective genetic algorithm utilizing several simple heuristic rules for solving simple assembly line balancing problems. The positional weights of the tasks were wrongly computed against the original definition developed by Helgeson and Birnie [12]. Despite the mistake, the validity of methodology of the mentioned paper cannot be questioned [11].

3.0 Methodology

The GA approach adopted in this paper for solving assembly line balancing problems is described as follows:

I. Representation of chromosomes

The GA 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.

II. 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 [13], where n is the number of task.

III. 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 [10]. 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 [10],

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

Where: ST =Station time is duration of station

CR=Realized cycle time

m =Total number of workstations

i= Chromosome number

The smoothness index for each chromosome was evaluated. A smoothness index of 0 indicates a perfect balance [14].

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

IV. 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 with a selection probability given as

$P_i = \frac{F_i}{\sum_{i=1}^{n} F_i}$ [13] (3)

Where: Fi = 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 [15].

V. 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 [16]. The crossover probability is assumed to be unity, that is P_c=1.0

VI. 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 [10].
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.

VII. 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 heuristics combined and recombined are explained below.

Longest operation time technique (LOT)

1. Construct the precedence diagram from the precedence table
2. Arrange the task in descending order of their task time i.e. from the longest to the shortest.
3. Assign the longest operation first while maintaining precedence and cycle time restriction [17]

The ranked positional weight technique (RPW).

1. Construct the precedence diagram
2. 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.
3. Rank the tasks in descending order based on the positional weight. i.e. from the highest PW to the lowest PW
4. Assign the task with the highest RPW first and proceed in that manner maintaining precedence and cycle time restrictions.[12, 18]
5.

The Kilbridge-Wester heuristic.

1. Construct the precedence diagram.
2. 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.
3. Assign task to the workstations starting with column I and continue while maintaining cycle time restriction.

The proposed genetic algorithm combined and recombined these heuristics [18]

THE APPROACH BY PONNAMBALAM ET AL [5]

To evaluate the performance of this algorithm, a comparison between a multi objective genetic algorithm utilizing fourteen simple heuristic rules proposed by [5] was made using the numerical illustration in that paper. The example problem has 12 tasks and a cycle time of 10 units. The precedence network of the presented example is graphically shown in Figure. 1

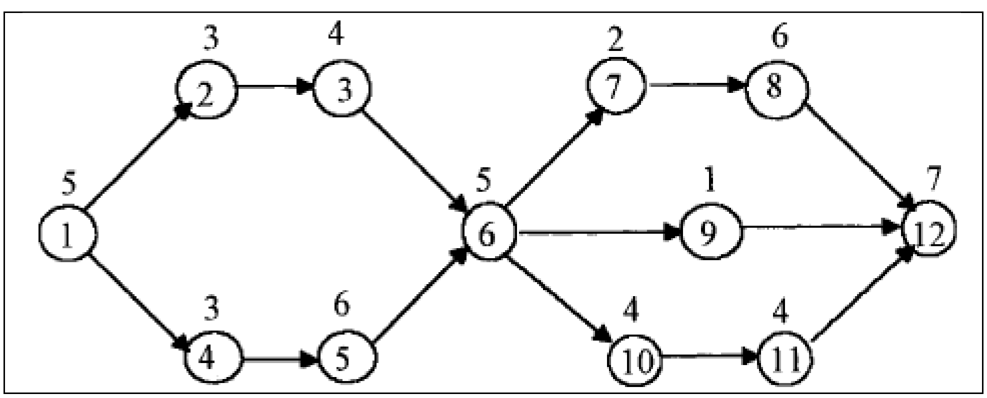

Figure 1.Precedence diagram of assembly network. [5]

The precedence table is shown in Table 1

Table 1.The precedence table Cycle time (CT) = 10.

Task No.	Task Time	Immediate Predecessor task
1	5	-
2	3	1
3	4	2
4	3	1
5	6	4
6	5	3,5
7	2	6
8	6	7
9	1	6
10	4	6
11	4	10
12	7	8,9, 11

The List of heuristic rules used by them while representing genes of chromosomes are;

1. Maximum ranked positional weight RPW
2. Maximum total number of follower tasks
3. Maximum task time
4. Maximum number of immediate follower tasks
5. Maximum backward recursive
6. Minimum total number of predecessor tasks
7. Minimum reverse positional weight
8. Minimum lower bound
9. Minimum upper bound
10. Minimum slack
11. Minimum task number
12. Random task assignment Random
13. Maximum task time of follower task
14. Maximum positional weight of follower task.

These 14 different heuristic rules were used to calculate the positional weights of tasks. Ranking of the tasks were based on the positional weights. However, it was observed that most of the positional weights calculated for the RPW technique were inaccurate as they were calculated as (34, 27, 24, 29, 26, 20, 15, 13, 8, 15, 11, 7). In this work, the correct result for the illustration problem using the original definition of RPW was computed as (50, 36, 33, 38, 35, 29, 15, 13, 8, 15, 11, 7).

An initial random population of 20 chromosomes, having 14 genes representing each of the heuristics was evaluated based on the objective and scalar fitness functions. The two point crossover and insertion mutation were utilized.

The solution obtained after 30 GA generations before the application of trade and transfer phase of the Moodie and Young method is shown in Table 2

Table 2.Initial assignment of tasks before trade and transfer phase.

Station	Task number	Task time (T_e)	Station time (ST_i)	$CT - ST_i$
1	1 4	5 3	8	2
2	2 5	3 6	9	1
3	3 6 9	4 5 1	10	0
4	7 8	2 6	8	2
5	10 11	4 4	8	2
6	12	7	7	3

Sources: [5]

Line efficiency (LE) =83.33% and Smoothness index (SI) =4.69

A. USING THE PROPOSED THREE HEURISTIC RULES GA

The three heuristics used in this study are described

i. Longest operation time technique (LOT)

The LOT method rank tasks in accordance with task times as shown in Table 3. Tasks having equal task times are given equal considerations but one has to be selected arbitrarily. This equality in task time gives rise to the different possible sub routes of the system. When two tasks possess equal chance of being selected, it is then possible to follow the route that may or may not enhance the performance of the system.

Table 3.Ranking of the tasks using the LOT heuristic

Task	Time	Rank
12	7	1
5, 8	6	2
1, 6	5	3
3, 10, 11	4	4
2,4	3	5
7	2	6
9	1	7

ii. The ranked positional weight technique (RPW).

The RPW technique rank tasks based on their positional weight. The positional weight of a task is the summation of the task time and the processing times of all its successors. Table 4 shows the task ranking based on positional weight. Sub routes are obtained when two or more tasks have equal ranked positional weight. Task number 7 and 10 having equal positional weight are assigned arbitrarily this lead to another possible route of the assembly line as shown in the Table 4.

Table 4.The positional weight and rank

Task	Positional Weight	Rank
1	50	1
4	38	2
2	36	3
5	35	4
3	33	5
6	29	6
7, 10	15	7
8	13	8
11	11	9
9	8	10
12	7	11

iii. The Kilbridge-Wester heuristic.

In Kilbridge and Wester heuristic, numbers are assigned to each operation describing how many predecessors it has. Operations with the lowest predecessors are assigned first to the workstations. Table 5 shows the task ranking based on their precedence. Tasks having the same number of predecessors possess equal chance in the assignment procedure this lead to a number of possible sub routes that could be followed.

Table 5.Ranking of the tasks using Kilbridge-Wester heuristic

column	task	rank
I	1	1
II	2,4	2
III	3,5	3
IV	6	4
V	7,9,10	5
VI	8,11	6
VII	12	7

B. IMPLEMENTATION OF THE GENETIC ALGORITHM

The propose genetic algorithm combined these three principles in navigating through the possible feasible routes of an assembly line network to achieve optimality as illustrated below;

Step 1: representation of chromosome.
The chromosome is a string of 12 genes corresponding to the 12 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* [13]. In this illustration,
the population size is taken to be *2n+1* where n= no of tasks. Therefore, population size= (2×12) +1=25. The randomly generated initial population of 25 chromosomes is shown in Table 6.

Step 3: evaluation of objective and fitness function.
Calculate the line efficiency and the smoothness index using equations 1 and 2. Table 7 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 25 chromosomes i.e. 20 chromosomes randomly based on fitness using the selection probability using equation (3). Table 8 shows the selection probability for each chromosome in the population

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 5th and 8th 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 9. Parent chromosome is designated as C_i and the offspring with O_i.
Appling the two point crossover to the selected chromosomes using *Pc*=1:

Step 6: Mutation
Apply the scramble mutation with *Pm*=0.2. This implies 0.2×25=5th 0ffspring. Choose the two points randomly as the 6th and 11th genes and scatter the genetic component within the points as shown in Table 10

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 [10] in this case, the stall generation is 10

Table 6.The randomly generated initial population

chromosomes	Genes											
	1	2	3	4	5	6	7	8	9	10	11	12
1	B	C	C	C	B	C	A	C	A	A	B	B
2	C	A	B	C	B	C	A	B	A	C	B	A
3	A	C	C	A	B	C	B	B	B	C	C	C
4	C	B	C	C	B	A	B	B	B	A	C	B
5	C	B	C	B	A	B	B	B	B	A	C	A
6	B	C	A	A	A	C	C	A	A	B	C	B
7	A	C	A	B	C	C	B	A	A	A	B	A
8	A	B	B	A	A	C	C	C	C	B	B	B
9	A	B	A	B	A	A	C	C	A	C	B	C
10	A	B	A	A	C	B	A	B	B	B	C	C
11	C	C	B	B	A	C	A	C	C	A	C	B
12	B	A	A	C	C	A	B	C	C	A	B	B
13	C	A	B	B	A	B	C	C	A	A	A	C
14	B	B	B	A	A	A	C	C	C	C	B	B
15	C	B	B	C	B	C	A	A	C	C	A	C
16	B	C	A	C	A	C	C	A	B	B	B	A
17	A	B	A	A	C	A	C	A	B	A	C	B
18	B	B	B	C	C	A	A	B	B	A	A	A
19	B	A	C	A	A	B	A	A	B	B	A	B
20	C	C	C	C	C	B	B	A	A	B	C	C
21	A	C	A	C	B	C	A	A	B	B	B	B
22	A	A	C	A	A	B	B	C	B	B	C	C
23	C	C	B	B	C	A	A	B	B	B	A	C
24	B	C	A	B	C	C	A	A	C	C	A	C
25	C	B	C	C	C	A	C	B	A	C	A	B

Table 7.The objective function and fitness values

chromosome	No of station	Line efficiency	Smoothness index	Rank
1	6	83.33	5.09	4
2	7	79.37	6.56	6
3	7	71.43	8	7
4	6	83.33	5.09	4
5	6	83.33	5.09	4
6	6	83.33	5.48	5
7	6	92.59	2.45	1
8	6	83.33	5.48	5
9	6	83.33	4.69	2
10	6	83.33	5.09	4
11	6	83.33	4.89	3
12	6	83.33	4.89	3
13	6	83.33	4.69	2
14	6	83.33	5.48	5
15	6	83.33	4.89	3
16	6	92.59	2.45	1
17	6	92.59	2.45	1
18	6	83.33	5.09	4
19	6	92.59	2.45	1
20	6	92.59	2.45	1
21	6	83.33	4.89	3
22	6	83.33	4.89	3
23	6	83.33	5.09	4
24	6	92.59	2.45	1
25	7	79.37	6.56	6

Table 8.The selection probabilities

Chromosome	Line efficiency	Selection probability
1	83.33	0.039
2	79.37	0.037
3	71.43	0.034
4	83.33	0.039
5	83.33	0.039
6	83.33	0.039
7	92.59	0.044
8	83.33	0.039
9	83.33	0.039
10	83.33	0.039
11	83.33	0.039
12	83.33	0.039
13	83.33	0.039
14	83.33	0.039
15	83.33	0.039
16	92.59	0.044
17	92.59	0.044
18	83.33	0.039
19	92.59	0.044
20	92.59	0.044
21	83.33	0.039
22	83.33	0.039
23	83.33	0.039
24	92.59	0.044
25	79.37	0.037

Table 9.The Crossover of the First Generation

Chromosomes	Genes											
	1	2	3	4	5	6	7	8	9	10	11	12
C_7	A	C	A	B	C	C	B	A	A	A	B	A
C_{16}	B	C	A	C	A	C	C	A	B	B	B	A
O_1	A	C	A	B	A	C	C	A	A	A	B	A
O_2	B	C	A	C	C	C	B	A	B	B	B	A
C_{17}	A	C	A	A	C	A	C	A	B	A	C	B
C_{19}	B	A	C	A	A	B	A	A	B	B	A	B
O_3	A	C	A	A	A	B	A	A	B	A	C	B
O_4	B	A	C	A	C	A	C	A	B	B	A	B
C_{20}	C	C	C	C	C	B	B	A	A	B	C	C
C_{24}	B	C	A	B	C	C	A	A	C	C	A	C
O_5	C	C	C	C	C	C	A	A	A	B	C	C
O_6	B	C	A	B	C	B	B	A	C	C	A	C

C_9	A	B	A	B	A	A	C	C	A	C	B	C
C_{13}	C	A	B	B	A	B	C	C	A	A	A	C
O_7	A	B	A	B	A	B	C	C	A	C	B	C
O_8	C	A	B	B	A	A	C	C	A	A	A	C

C_{11}	C	C	B	B	A	C	A	C	C	A	C	B
C_{12}	B	A	A	C	C	A	B	C	C	A	B	B
O_9	C	C	B	B	C	A	B	C	C	A	C	B
O_{10}	B	A	A	C	A	C	A	C	C	A	B	B

C_{15}	C	B	B	C	B	C	A	A	C	C	A	C
C_{21}	A	C	A	C	B	C	A	A	B	B	B	B
O_{11}	C	B	B	C	B	C	A	A	C	C	A	C
O_{12}	A	C	A	C	B	C	A	A	B	B	B	B

C_{22}	A	A	C	A	A	B	B	C	B	B	C	C
C_1	B	C	C	C	B	C	A	C	A	A	B	B
O_{13}	A	A	C	A	B	C	A	C	B	B	C	C
O_{14}	B	C	C	C	A	B	B	C	A	A	B	B

C_4	C	B	C	C	B	A	B	B	B	A	C	B
C_5	C	B	C	B	A	B	B	B	B	A	C	A
O_{15}	C	B	C	C	A	B	B	B	B	A	C	B
O_{16}	C	B	C	B	B	A	B	B	B	A	C	A

C_{10}	A	B	A	A	C	B	A	B	B	B	C	C
C_{18}	B	B	B	C	C	A	A	B	B	A	A	A
O_{17}	A	B	A	A	C	A	A	B	B	B	C	C
O_{18}	B	B	B	C	C	B	A	B	B	A	A	A

C_{23}	C	C	B	B	C	A	A	B	B	B	A	C
C_6	B	C	A	A	A	C	C	A	A	B	C	B
O_{19}	C	C	B	B	A	C	C	A	B	B	A	C
O_{20}	B	C	A	A	C	A	A	B	A	B	C	B

Table 10. The Mutation of the First Generation

Chromosomes	Genes											
	1	2	3	4	5	6	7	8	9	10	11	12
C_5	C	C	C	C	C	C	A	A	A	B	C	C
O_5	C	C	C	C	C	A	C	A	B	A	A	C

The optimum solutions of the first generation is shown in Table 11

Table 11. Optimum Solutions of First Generation

Chromosomes	Genes											
	1	2	3	4	5	6	7	8	9	10	11	12
C_7	A	C	A	B	C	C	B	A	A	A	B	A
C_{16}	B	C	A	C	A	C	C	A	B	B	B	A
C_{17}	A	C	A	A	C	A	C	A	B	A	C	B
C_{19}	B	A	C	A	A	B	A	A	B	B	A	B
C_{20}	C	C	C	C	C	B	B	A	A	B	C	C

The assembly line of the pareto optimal solutions realized from the initial population are shown in Tables 12 -15

Table 12.Chromosome 7 and 16 Assembly line balancing

No of station	Task No.	Task Time	Station time(ST_i)	$ST_{max} - ST_i$
1	1 4	5 3	8	1
2	5 2	6 3	9	0
3	3 6	4 5	9	
4	7 8	2 6	8	1
5	10 11 9	4 4 1	9	0
6	12	7	7	2

Line efficiency (LE) = 92.59% and Smoothness index (SI) =2.45

Table 13.Chromosome 17 Assembly line balancing

No of station	Task No.	Task Time	Station time(ST_i)	$ST_{max} - ST_i$
1	1 4	5 3	8	1
2	5 2	6 3	9	0
3	3 6	4 5	9	
4	10 11	4 4	8	1
5	7 8 9	2 6 1	9	0
6	12	7	7	2

Line efficiency (LE) =92.59% and Smoothness index (SI) =2.45

Table 14.Chromosome 19 Assembly line balancing

No of station	Task No.	Task Time	Station time(ST_i)	$ST_{max} - ST_i$
1	1 2	5 3	8	1
2	4 5	3 6	9	0
3	3 6	4 5	9	
4	10 11	4 4	8	1
5	7 8	2 6	9	0
6	12	7	7	2

Line efficiency (LE)= 92.59% and Smoothness index (SI) =2.45

Table 15.Chromosome 20 Assembly line balancing

No of station	Task No.	Task Time	Station time(ST_i)	$ST_{max} - ST_i$
1	1 2	5 3	8	1
2	4 5	3 6	9	0
3	3 6	4 5	9	
4	7 8	2 6	8	1
5	10 11 9	4 4 1	9	0
6	12	7	7	2

Line efficiency (LE) = 92.59% and Smoothness index (SI) =2.45

The Successor Generation

The next generation after crossover and mutation is shown in Table 16

Table 16.The successor generation

Chromosome	Genes											
	1	2	3	4	5	6	7	8	9	10	11	12
1	A	C	A	B	A	C	C	A	A	A	B	A
2	B	C	A	C	C	C	B	A	B	B	B	A
3	A	C	A	A	A	B	A	A	B	A	C	B
4	B	A	C	A	C	A	C	A	B	B	A	B
5	C	C	C	C	C	A	C	A	B	A	A	C
6	B	C	A	B	C	B	B	A	C	C	A	C
7	A	B	A	B	A	B	C	C	A	C	B	C
8	C	A	B	B	A	A	C	C	A	A	A	C
9	C	C	B	B	C	A	B	C	C	A	C	B
10	B	A	A	C	A	C	A	C	C	A	B	B
11	C	B	B	C	B	C	A	A	C	C	A	C
12	A	C	A	C	B	C	A	A	B	B	B	B
13	A	A	C	A	B	C	A	C	B	B	C	C
14	B	C	C	C	A	B	B	C	A	A	B	C
15	C	B	C	C	A	B	B	B	B	A	C	B
16	C	B	C	B	B	A	B	B	B	A	C	A
17	A	B	A	A	C	A	A	B	B	B	C	C
18	B	B	B	C	C	B	A	B	B	A	A	A
19	C	C	B	B	A	C	C	A	B	B	A	C
20	B	C	A	A	C	A	A	B	A	B	C	B
21	A	C	A	B	C	C	B	A	A	A	B	A
22	B	C	A	C	A	C	C	A	B	B	B	A
23	A	C	A	A	C	A	C	A	B	A	C	B
24	B	A	C	A	A	B	A	A	B	B	A	B
25	C	C	C	C	C	B	B	A	A	B	C	C

The summary of the results obtained by the two approaches is shown in Table 17.

Table 17. summary of the results

No	FACTORS	FOURTEEN HEURISTICS GA	THREE HEURISTICS GA
1	Number of workstations generated	6	6
2	Line efficiency	83.33%	92.59%
3	Smoothness index	4.69	2.45
4	Number of heuristics used	14	3
5	Cycle time	10mins	9mins

5.0 Results and Discussions

Optimization could be defined as the effort, way, technique, method or system to use for calculating or finding the best possibilities of utilization of resources (which can be people, time, process, vehicles equipment, raw materials, supplies and others) needed to achieve an expected result, with it being the best possible solution to the problem [19]. In an optimisation problem, a list, quite possibly of infinite length, of possible solutions is being searched in order to locate the solution that best describes the problem at hand.

The multi-objective genetic algorithm proposed and utilised by [5] uses fourteen heuristics. It was found to perform better than any of the combined heuristics. The performance measures used in this GA are the number of excess stations, the line efficiency and the smoothness index. The actual result obtained from the genetic algorithm is six stations, 83.33% line efficiency and a smoothness index of 4.69. An improvement in the line efficiency and smoothness index of the line was achieved by the application of the trade and transfer phase of the Moodie and Young method.

The results produced by the proposed genetic algorithm for the illustrative problem gave an assembly line with the minimum possible number of workstations, which is six (6) with a line efficiency of 92.59% and smoothness index of 2.45. The ideal of the realized cycle time adopted give room for optimality to be achieved in terms of cycle time without making several infeasible assumptions in selecting cycle time. By adopting this concept of realized cycle time, a cycle time of 9 minutes was obtained for the six workstations instead of the prescribed cycle time of 10 minutes.

Comparing the procedure adopted by this three heuristics GA in accordance with the definition of optimization, with that of the fourteen different heuristic rules adopted by [5], it is observed that the three heuristics GA is easier and faster to solve. It gives optimal solutions without additional procedure to the GA methodology.

The result obtained by the fourteen heuristic genetic algorithm approach and that of the three heuristic genetic algorithm approach as illustrated in Table 17 had shown that the adopted genetic algorithm in realizing an optimum solution to assembly line balancing problems is better.

6.0 Conclusion

In this paper, the proposed genetic algorithm which combines three different heuristics and pair the chromosomes according to their fitness has been used as an easy and straight forward approach in solving assembly line balancing problems in order to obtain optimum solutions within a reasonable time of its implementation. The utilization of just three heuristics had made the GA easier to understand and solve. The use of realised cycle time has been helpful in increasing the efficiency of assembly line balance when applied.

References

[1] Suwannarongsri S, Limnararat S and Puangdownreong D (2007) Hybrid Tabu Search Method for Assembly Line Balancing Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization, Beijing, China. pp. 443-448.

[2] Kriengkorakot N and Pianthong N (2007) The Assembly Line Balancing Problem: Review articles. KKU Engineering Journal Vol. 34 No.2, pp. 133 - 140

[3] Betancourt L.C (2007); Ph.D Thesis, Technical University of Catalonia: ASALBP: The Alternative Subgraphs Assembly Line Balancing Problem. Formalization and Resolution Procedures.

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

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

[6] 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.

[7] 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.

[8] 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

[9] Brudaru. O, Valmar. B and Popovici. D (2008). Hybrid genetic algorithm for assembly line balancing with fuzzy times and parallel workstations. Buletinul institutului politehnic din iaşi publicat de universitatea tehnică „gheorghe asachi" din iaşi tomul liv (lviii), fasc. 2, secţia automatică şi calculatoare

[10] 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.

[11] Fathi M, Ariffin M. K. A and Ismail N (2010). A note on "A multi-objective genetic algorithm for solving assembly line balancing problem" International Journal Advance Manufacturing Technology 50, pp. 771–773

[12] Helgeson W.P and Birnie D.P (1961) "Assembly line balancing using the ranked positional weight technique", Journal for Industrial Engineering 12 (6), pp 394-398

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

[14] 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 12 Issue 3 Version 1. 0, USA

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

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

[17] 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

[18] Ponnambalam S.G, Aravindan P and Mogileeswar G. N (1999). A Comparative Evaluation of Assembly Line Balancing Heuristics. International Journal Advance Manufacturing Technology 15(8), pp. 577-586

[19] Matondang M. Z and Jambak M.I (2010). Soft Computing in Optimizing Assembly Lines Balancing. Journal of Computer Science 6 (2), pp. 141-162