ISSN 2229-5518

Optimization of balancing for a mixed multi model assembly line

Anoop Kumar Elia and Dr. D. Choudhary
Guru Nanak Dev Engineering College, Bidar- 585403(Karnataka State) INDIA
anoopkumar965@gmail.com

Abstract— The fundamental of Assembly line balancing problems is to assign the tasks to an ordered sequence of stations, such that the precedence relations are satisfied and some of the measurements of effectiveness are optimized. (e.g. to minimize the number of work stations; minimize the balance delay; minimizing line balancing loss, idle time minimizing, minimizing cycle time etc). The problem of Assembly Line Balancing deals with the distribution of activities among all the workstations so that there will be maximum utilization of human resources and facilities without affecting the work sequence. In this work, a mixed multi model assembly line (MMuALB) is optimized for the best performance. Four models with different quantities of production are treated for assembly line optimization. Initially, the average of task times of all the four models is computed in order to allocate the tasks to stations. The total number of stations required for allocation is also optimized. Once the allocation of tasks is completed, based on the averaged task times the load balancing is calculated for each station, for each model and for all the station models. The idle time of each station model is reduced by an iterative algorithm by incrementing the quantity of the appropriate model. A generic assembly line of Buxey 29 tasks problem is solved for allocation of tasks of each of the four models for the least idle time of the stations.

Index Terms— *Allocation of tasks, line optimization, minimization, models load, number of stations, idle time of each station, iterations, balancing load, minimize the balance delay.*

1 Introduction

Assembly lines are the most commonly used method in mass production environments because they enable assembly of complex products by workers with limited training. The main objective of assembly systems designers is to increase the efficiency of the line by maximizing the ratio between throughput and required costs. Thus, assembly line design is a problem of considerable industrial importance. Assembly Line production is one of the widely used basic principles in production system. The problem of Assembly Line balancing deals with the distribution of activities among the workstations so that there will be maximum utilization of human resources and facilities without disturbing the work sequence. An assembly line is formed of a finite set of work elements which are also referred to as tasks. Each task is identified by a processing time for the operation it represents and a set of relationships for precedence, which specifies the allowable ordering of the tasks. Assembly line balancing (ALB) is defined as a process in which a group of tasks to be performed are allocated on an ordered sequence of assembly line. Systematic design of assembly lines is not a simple and easy task for the designers. Manufacturers and Designers have to deal with their existing factory layout in the initial phase. The Cost associated and reliability of the system, complexities involved in tasks, selection of equipment, operating criteria of assembly line, multiple constraints, scheduling methodologies, allocation of stations, control of inventory, buffer allocation are the most important area of concern.

The first published paper of the assembly line balancing problem (ALBP) was made by Salveson (1955) [1] who suggested a linear programming solution. Since then, the topic of line balancing has been of great interest to researchers. However, since the ALB problem falls into the NP hard class of combinatorial optimization problems (Gutjahr and Nemhauser, 1964) [2] it has consistently developed the efficient algorithms for obtaining optimal solutions. Thus numerous research efforts have been directed towards the development of computer-efficient approximation algorithms or heuristics (e.g. Kilbridge and Wester, 1961 [3]; Helgeson and Birnie, 1961 [4]; Hoffman, 1963 [5]; Mansoor, 1964 [6]; Arcus, 1966[7]; Baybar, 1986a [8] and exact methods to solve the ALB problems. (e.g. Jackson, 1956 [9]; Bowman, 1960 [10];Van Assche and Herroelen, 1978 [11]; Mamoud, 1989 [12]; Sarin et al., 1999 [13]).

For balancing an assembly line, one has to take into considerations the following issues such as number of products or models, deterministic or stochastic nature of task durations, line layout, flow of work pieces, and level of automation. Accordingly, need to think of different classes of assembly line (Boyson *et al.*, 2007) [14]. For a detailed review of the related Literature on generalized assembly line balancing, there is a need to refer Scholl (1999) [15], and Becker and Scholl (2006) [16]. There are three ways of handling an optimization problem involved in assembly line balancing. These are heuristic approach (Boctor, 1995 [17]; Amen, 001 [18]; Scholl and Becker, 2006 [16]), programming approach (Pinnoi and Wilhelm, 1998 [19]; Bukchin and Rabinowitch, 2006 [20];Peeters, 2006 [21]) and simulation approach (Grabau *et al.*, 1997 [22] and McMullen and Frazier, 1998 [23]). Assembly lines

ISSN 2229-5518

are special flow-line production systems that are typical in the industrial assembly of high volume standardized commodities (Scholl, 1999) [15].

During the assembly process the product traverses the assembly line, station by station, while in each workstation a fixed predetermined set of tasks is performed. Each task is an atomic working unit, which usually requires specific machinery and skill. The assembly line design involves the assignment of these tasks into the work- stations, subject to given precedence relationships among the tasks. The least complex configuration of an assembly line is the single-model assembly line. The most common objective of single model assembly line balancing problem (SALB-P) is to maximize the efficiency of the assembly line by minimizing the required capacity per unit of throughput. This goal can be attained either by minimizing the number of workstations for a given a required cycle time or by minimizing the cycle time for a given the number of workstations. Comprehensive surveys of related research appear in Baybars (1986) [24], Ghosh and Gagnon (1989) [25] and Scholl and Becker [16]. The mixed-model assembly line is a more complex environment in which several variants of the product, referred to as models, are assembled simultaneously on the line. The line balancing problem in a mixed-model environment (MALB- P) involves the assignment of tasks of all models to the workstations. This problem is much more complex since it entails the additional considerations of the interactions between the assembled models. MALB-P reflects modern assembly lines more realistically, where demand is characterized by high variability and relatively small volume for each model. This phenomenon can be observed in the increased number of models of cars, TVs, computers, VCRs and many other products.This problem has been investigated during the last four decades, where some of the earliest works were those of Thomopoulos. For recent surveys of the various types of assembly line balancing problems, need to refer Erel and Sarin (1998) [26] and Becker and Scholl [16]. Designer's goal is to design a Assembly line considering higher efficiency, less balance delay, smooth production, optimized processing time, cost effectiveness, overall labour efficiency and just in time production (JIT). The aim is to propose a line by exploiting the best of the design methods which will deal in actual fact with user preferences.

2 Proposed Algorithm

For equivalent single models, the algorithm is defined below. The algorithm delivers the number of feasible solutions.

a. Predict the average number of stations required using NOOFSTATIONS=NOOFTASKS/3
b. Round off the NOOFSTATIONS to the lower integer.
c. Assign a new station STATION[1] with a cycle time T = MINCYCLETIME
d. Determine all the tasks that do not have the predecessor TASKSWOPRED = { i, j,…., n}
e. Assign one task in TASKSWOPRED to STATION[1]
f. Remove the tasks that is assigned to STATION [1] from the graph and update it as TASKSWOPRED = {j, k,…., n }.
g. Update the station cycle time as T = MINCYCLETIME - t_i
h. Repeat steps e to g, until T is positive and update the T and TASKSWOPRED each time.
i. When T turns negative, look for any other tasks in TASKSWOPRED to fit in STATION [1], but the T should remain positive.
j. When T turns zero or negative for all the tasks in TASKSWOPRED, create a new station as STATION [2].
k. Repeat steps e to j.
l. Repeat step e to k for all feasible solutions.
m. Try the solutions for a pre-decided number of stations. If the solutions derived are not feasible, repeat e to k after update the T as MINCYCLETIME+1.
n. When all the feasible solutions are obtained, store the updated T.
o. Decrease the number of stations to 1 less than the NOOFSTATIONS and run the above procedure again.
p. Increase the number of stations to 1 more than the NOOFSTATIONS and run the above procedure again.
q. Freeze the number of stations with best performance as NOOFSTATIONS
r. Define the number of models as NUMMOD
s. Define the quantity of each model to be produced as MOD_i , i=1,2,3..M

ISSN 2229-5518

t. Calculate the entitled load bearing capacity of all stations for each model for the required quality as ENT_CAP, i=1,2,3..M

u. Calculate the time required for completion of all the tasks by each station for each model for the required quality as ACT_LOAD, i=1,2,3..M

v. Calculate the Smoothed load assignment SMT_LOAD_ASSGN = ENT_CAP-ACT_LOAD, i=1,2,3..M

w. Determine the maximum of the SMT_LOAD_ASSGN, i=1,2,3..M

x. Increment the MOD _i for which the SMT_LOAD_ASSGN is maximum

y. Repeat the steps t to w.

z. Run the step y for predetermined number of times and choose the quantity of each model based on the criteria of total of SMT_LOAD_ASSGN, for i=1, 2 ... M, is minimum.

3. Simulation Results:

For experimental; purpose, Buxey 29 tasks Problem [27] is chosen. The precedence diagram for the Buxey is presented in Fig.1. In case of multiple models, the equivalent task diagram can be derived in the form shown in Fig.1. For the sake of simplicity, a single model precedence diagram is shown and solved in this work. The Buxey problem has a total of 29 tasks and the associated tasks are shown above each task in Fig.1.

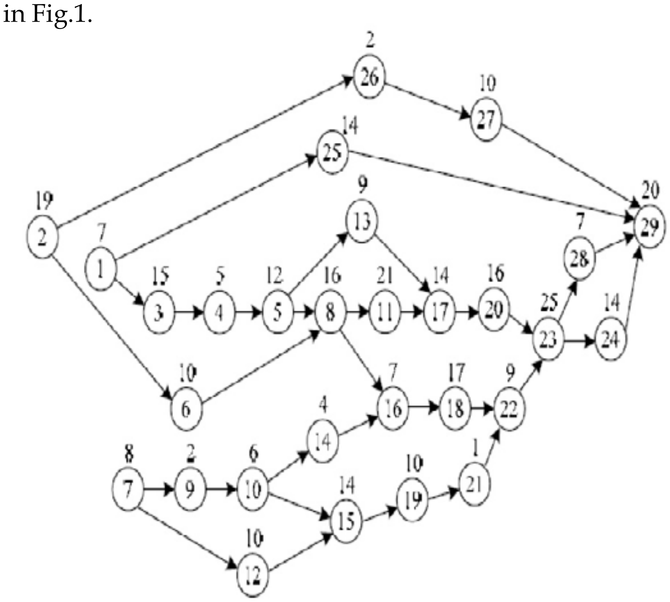

Figure1 Buxey 29 tasks precedence diagram.

Table1 shows the precedence matrix for the Buxey 29 tasks problem. In the matrix, the columns and rows represent the task number. It shows the precedence relation between the tasks. For example, in row 2 and column 6, it is indicated as 1 in the matrix, which means the task 6 is preceded by task 2. If the value is zero, it means there is no precedence relationship in the diagram. There are four models considered in the assembly line with different task times. The last five rows of the Table 1 shows the times associated with each task for four different models and their average. In this work, two factors are varied to optimize the assembly line for best performance. The first factor to be varied is the number of stations. However, the best solution for practical implementation to be chosen based on the minimum cycle time and the complexity involved in transportation and assignment of tasks to these stations as well as the cost of maintenance of these stations. The second factor to be varied is the quantity of each model to be manufactured. With this, one can optimize the assembly line for better balancing and to reduce the idle time of the station as low as possible.

Table 1: Precedence matrix for the Buxey 29 Tasks problem and the task times for each model

Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	0	0	1	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
2	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	1	0	0	0
3	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	0	0	0
4	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	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	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
7	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	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
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	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	0	0
12	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	0	0	0	0
13	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	0	0
14	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	0	0	0
15	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
16	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	0
17	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
18	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
19	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
20	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
21	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
22	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
23	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
24	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
25	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
26	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
27	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
28	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
29	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
	Task Times																												
Model 1	7	19	15	5	12	10	8	16	2	6	21	10	9	4	14	7	14	17	10	16	1	9	25	14	14	2	10	7	20
Model 2	5	13	17	6	14	9	9	15	3	5	19	9	8	5	13	6	15	16	11	13	2	8	21	11	13	3	9	6	18
Model 3	6	15	13	6	13	8	10	13	3	6	20	9	9	5	12	6	16	15	12	12	2	8	27	11	12	3	8	6	21
Model 4	8	21	22	5	11	12	11	18	3	8	22	11	10	5	15	6	12	15	9	11	1	7	25	13	12	3	7	7	19
Average	6.5	17	16.75	5.5	12.5	9.75	9.5	15.5	2.75	6.25	20.5	9.75	9	4.75	13.5	6.25	14.25	15.75	10.5	13	1.5	8	24.5	12.25	12.75	2.75	8.5	6.5	19.5

By running the algorithm mentioned above from steps a to p, various solutions are obtained and assignment of tasks to each station is derived. The lower the number of stations, lowers the cost of maintenance, but may be higher the idle time of stations when all different models are considered. As a first step, the average time of each task of different models is considered to derive the assignment of tasks to each station in each solution. The solutions are obtained for 8 stations, 9 stations and 10 stations.

Table 2: Feasible solutions for 8 stations for the Buxey 29 Tasks problem

Task ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Solution 1	1	2	1	1	3	2	3	4	3	3	5	3	5	5	4	5	6	6	4	7	4	6	7	8	1	2	2	8	8
Solution 2	1	1	1	2	3	2	2	4	2	3	5	4	3	3	4	5	6	6	5	7	5	6	7	8	2	2	3	8	8
Solution 3	2	1	3	3	3	2	1	4	1	1	4	5	3	1	5	5	6	6	5	7	6	6	7	8	2	1	2	8	8
Solution 4	2	1	3	3	3	1	1	4	1	4	5	2	3	4	4	5	6	6	5	7	5	6	7	8	2	1	2	8	8
Solution 5	1	2	1	1	3	3	2	4	2	3	5	2	5	5	4	5	6	6	4	7	4	6	7	8	1	2	3	8	8
Solution 6	1	2	2	2	3	3	1	4	1	3	5	1	5	5	4	5	6	6	4	7	4	6	7	8	1	2	3	8	8
Solution 7	1	3	1	1	2	3	2	4	2	5	4	2	2	5	5	5	6	6	5	7	6	6	7	8	1	3	3	8	8

Table 2 shows 7 solutions obtained for a case of 8 stations, which means there are 7 possible ways the tasks can be assigned to 8 stations considering the algorithm from steps a to p. This assignment of tasks is made based on the average time of all the models for each task. However, the best solution among the 7 solutions can be chose based on the total time required for each solution. Table 4.3 shows the total time taken by each station under each solution. Of all the solutions, Solution 1 provides the best cycle total time of 327 sec. The cycle time for the solution 1 is 41 sec.

Table 3: Total time taken by each station for the Buxey 29 Tasks problem

Solution ->	1	2	3	4	5	6	7
Station 1	43	54	45	46	49	46	59
Station 2	39	45	55	55	46	49	48
Station 3	43	43	54	54	48	48	48
Station 4	43	42	53	42	43	43	50
Station 5	42	41	42	41	42	42	44
Station 6	39	39	41	47	47	47	41
Station 7	38	38	38	38	38	38	38
Station 8	40	40	40	40	40	40	40
Total time	327	342	368	363	353	353	368

Now, after the solution 1 is frozen, which is based on the average time of tasks for all the models, the load balancing for each of the models in the 8 stations model is computed using the steps r to z of the algorithm. In this work, the quantity of each of the four models is considered as 10. For a quality of 10 for each model on a 8 station model, iteration 1 shows the idle time of the 8 stations. There will be a idle time 65 sec when model 1 being manufactured and 52.5 sec when model 2 is manufactured and so on. A total idle time of 192.5 sec is present if the quantity is 10 for each of the model.

As part of next iteration, find out the model for which the idle time is highest in the previous iteration and increment the quantity of that model. For example, model 1 has an idle time of 65 sec which is the highest in the iteration 1. Hence increment the quantity from 10 to 11 for model 1. Again calculate the idle time with new quantities in iteration 2. The last column of the table 4.4 shows the model which has the highest idle time in the previous iteration and whose quantity needs to be incremented in the iteration that follows.

Table 4: Idle time of 8 stations and the quantity of models for each iteration

	Smoothing-Model 1	Smoothing-Model 2	Smoothing-Model 3	Smoothing-Model 4	Total Smoothing	Model 1 - Qty	Model 2 - Qty	Model 3 - Qty	Model 4 - Qty	Incrementing model
Iteration 1	65	52.5	56.25	18.75	192.5	10	10	10	10	0
Iteration 2	24.5	52.5	56.25	18.75	152.0	11	10	10	10	1
Iteration 3	24.5	52.5	17.875	18.75	113.6	11	10	11	10	3
Iteration 4	24.5	14.75	17.875	18.75	75.9	11	11	11	10	2
Iteration 5	16	14.75	17.875	18.75	67.4	12	11	11	10	1
Iteration 6	31	57.75	61.875	20.625	171.3	12	11	11	11	4
Iteration 7	31	57.75	23.5	20.625	132.9	12	11	12	11	3
Iteration 8	31	20	23.5	20.625	95.1	12	12	12	11	2
Iteration 9	9.5	20	23.5	20.625	73.6	13	12	12	11	1
Iteration 10	9.5	20	14.875	20.625	65.0	13	12	13	11	3
Iteration 11	37.5	63	29.125	22.5	152.1	13	12	13	12	4
Iteration 12	37.5	25.25	29.125	22.5	114.4	13	13	13	12	2
Iteration 13	3	25.25	29.125	22.5	79.9	14	13	13	12	1
Iteration 14	3	25.25	9.25	22.5	60.0	14	13	14	12	3
Iteration 15	3	12.5	9.25	22.5	47.3	14	14	14	12	2
Iteration 16	44	30.5	34.75	24.375	133.6	14	14	14	13	4
Iteration 17	3.5	30.5	34.75	24.375	93.1	15	14	14	13	1
Iteration 18	3.5	30.5	3.625	24.375	62.0	15	14	15	13	3
Iteration 19	3.5	7.25	3.625	24.375	38.8	15	15	15	13	2
Iteration 20	50.5	35.75	40.375	26.25	152.9	15	15	15	14	4

This procedure may be repeated for predetermined number of iteration and the combination of quantities for all the four models to be chosen based on the least idle total time. In the present case, iteration 19 provides the model quantities as 15, 15, 15 and 13 respectively for model 1, model 2, model 3 and model 4. This leaves out an idle time of 38.8 sec. This can also be witnessed in Fig. 2. If this procedure is not adopted, then one needs to satisfy with a quantity of 10 for each model which leaves out a total idle time as 192.5 sec.

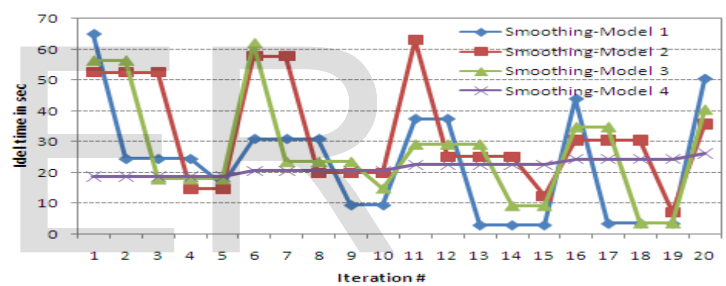

Figure 2: Idle time of stations for each iteration

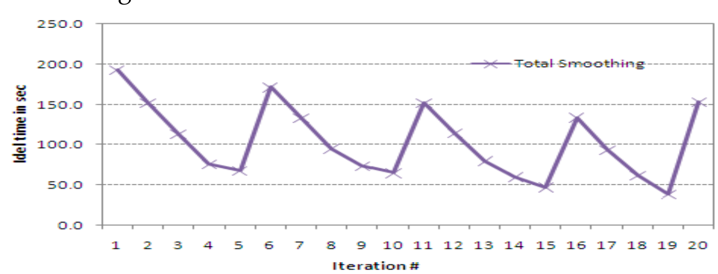

Figure 3: Total Idle time of stations for all the models for each iteration

The maximum number of iterations chosen in this work is 20 and this can be increased or decreased based on the user's choice. By further running the algorithm for more number of iteration than 20, one may still obtain the total idle time much lesser than 38.8 sec. By observing the lowest points of the profile of the total smoothing in Fig. 3, the trend shows that by further increasing the number of iterations, the idle time can be reduced further.

Table 5: Feasible solutions for 9 stations for the Buxey 29 Tasks problem.

ISSN 2229-5518

Task ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Solution 1	1	1	3	3	4	2	1	4	1	2	5	2	4	3	6	6	5	7	7	6	7	7	8	9	3	1	2	8	9
Solution 2	2	1	3	3	4	1	1	4	2	2	5	2	6	3	6	4	6	5	7	7	7	7	8	9	3	2	2	8	9
Solution 3	1	2	1	2	3	3	1	4	1	1	4	2	6	3	6	5	6	5	7	7	7	7	8	9	5	2	3	8	9
Solution 4	1	1	2	2	3	1	4	3	4	4	4	5	3	6	5	6	5	7	6	6	6	7	8	9	2	1	7	8	9
Solution 5	2	1	2	3	4	3	1	4	2	3	5	1	4	3	5	6	6	6	7	7	7	7	8	9	2	3	3	8	9
Solution 6	1	3	1	1	2	3	1	5	1	4	5	2	3	4	4	6	6	6	7	7	7	7	8	9	2	4	4	8	9
Solution 7	1	2	1	2	2	3	3	5	4	4	5	4	3	4	4	6	6	7	7	6	7	7	8	9	1	2	3	8	9
Solution 8	2	1	3	3	4	2	1	5	1	1	5	2	4	3	4	6	6	7	7	6	7	7	8	9	3	1	2	8	9
Solution 9	1	3	1	2	2	3	2	4	2	3	4	5	2	5	6	6	5	7	7	6	7	7	8	9	1	3	5	8	9
Solution 10	1	2	1	2	3	3	4	3	5	5	4	6	4	7	6	7	5	7	6	5	6	7	8	9	1	2	2	8	9
Solution 11	1	2	1	2	2	4	3	4	4	4	5	3	3	4	6	6	5	6	7	7	7	7	8	9	1	2	3	8	9
Solution 12	1	1	2	2	3	1	2	5	2	2	5	3	3	3	4	6	6	6	4	7	5	7	8	9	4	1	7	8	9
Solution 13	1	1	2	2	3	1	3	3	3	4	4	5	5	5	6	6	5	7	7	6	7	7	8	9	2	1	4	8	9
Solution 14	1	2	1	2	3	2	4	3	5	5	4	6	4	6	7	6	5	6	7	5	7	7	8	9	1	2	3	8	9
Solution 15	1	1	2	2	3	1	4	3	5	5	4	6	4	7	6	7	5	7	6	5	6	7	8	9	2	1	3	8	9

Again, the number of stations is varied from 8 to 9 and the algorithm as mentioned above for steps a to p is run. In this case, there are 15 feasible solutions are obtained as shown in Table 4. By increasing the number stations, there is a significant increase in the number of solutions. However, the best solution for practical implementation to be chosen based on the minimum cycle time and the complexity involved in transportation and assignment of tasks to these stations. The cost of other resources also should be considered when choosing the best feasible solution.

Table 6: Total time taken by each of 9 stations for the Buxey 29 Tasks problem.

Solution ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Station 1	40	43	44	44	47	43	50	40	47	47	47	40	40	43	40
Station 2	36	39	48	51	52	61	51	48	50	53	57	52	55	52	55
Station 3	41	41	37	47	40	53	50	53	49	54	50	49	54	53	53
Station 4	38	36	44	41	59	38	39	55	70	59	41	50	49	52	52
Station 5	36	37	36	39	50	52	52	52	39	38	64	52	39	38	38
Station 6	34	38	38	38	53	53	50	50	49	39	41	42	38	40	39
Station 7	37	34	34	41	34	34	37	37	37	36	34	38	37	35	36
Station 8	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
Station 9	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
Total time	327	333	346	366	400	399	394	400	406	391	399	388	377	378	378

Here again, the solution 1 yields the best possible solution since the cycle time is minimum of all. Solution 1 takes a total time of 327 sec which is same as in the case of 8 station models. The cycle time in this case is 37 sec. If the cost of installation of the stations is given priority, it is the 8 station model, which suits best for this problem over 9 station model.

Table 7: Idle time of 9 stations and the quantity of models for each iteration

	Smoothing-Model 1	Smoothing-Model 2	Smoothing-Model 3	Smoothing-Model 4	Total Smoothing	Model 1 - Qty	Model 2 - Qty	Model 3 - Qty	Model 4 - Qty	Incrementing model
Iteration 1	100.0	44.4	98.9	24.4	267.8	10	10	10	10	0
Iteration 2	64.0	44.4	98.9	24.4	231.8	11	10	10	10	1
Iteration 3	64.0	44.4	64.8	24.4	197.7	11	10	11	10	3
Iteration 4	64.0	44.4	30.7	24.4	163.6	11	10	12	10	3
Iteration 5	28.0	44.4	30.7	24.4	127.6	12	10	12	10	1
Iteration 6	28.0	10.9	30.7	24.4	94.0	12	11	12	10	2
Iteration 7	28.0	10.9	3.4	24.4	66.8	12	11	13	10	3
Iteration 8	8.0	10.9	3.4	24.4	46.8	13	11	13	10	1
Iteration 9	38.0	48.9	40.6	26.9	154.3	13	11	13	11	4
Iteration 10	38.0	15.3	40.6	26.9	120.8	13	12	13	11	2
Iteration 11	38.0	15.3	6.4	26.9	86.7	13	12	14	11	3
Iteration 12	2.0	15.3	6.4	26.9	50.7	14	12	14	11	1
Iteration 13	48.0	53.3	50.4	29.3	181.1	14	12	14	12	4
Iteration 14	48.0	19.8	50.4	29.3	147.6	14	13	14	12	2
Iteration 15	48.0	19.8	16.3	29.3	113.4	14	13	15	12	3
Iteration 16	12.0	19.8	16.3	29.3	77.4	15	13	15	12	1
Iteration 17	58.0	57.8	60.3	31.8	207.9	15	13	15	13	4
Iteration 18	58.0	57.8	26.2	31.8	173.8	15	13	16	13	3
Iteration 19	22.0	57.8	26.2	31.8	137.8	16	13	16	13	1
Iteration 20	22.0	24.2	26.2	31.8	104.2	16	14	16	13	2

By executing the algorithm from steps r to z, the idle times for the each models are obtained during each iterations as shown in table 7. The procedure is same as mentioned and the least of the total idle times is obtained during iteration 8, for which the quantity of models are 13, 11, 13 and 10 respectively. The idle times for each model and the total idle time can be viewed from Figs. 4 and 5. The total idle time is 46.8 sec and this is slightly higher than that of 8 stations model, which is 38.8 sec.

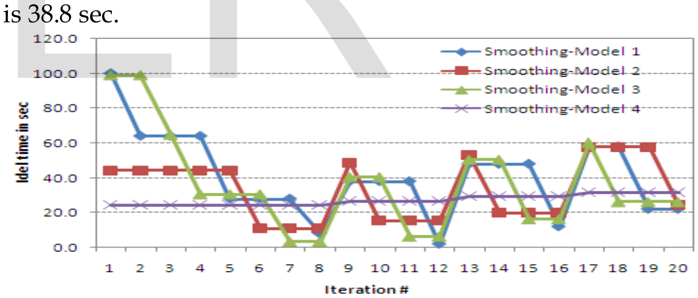

Figure 4: Idle time of stations for each iteration

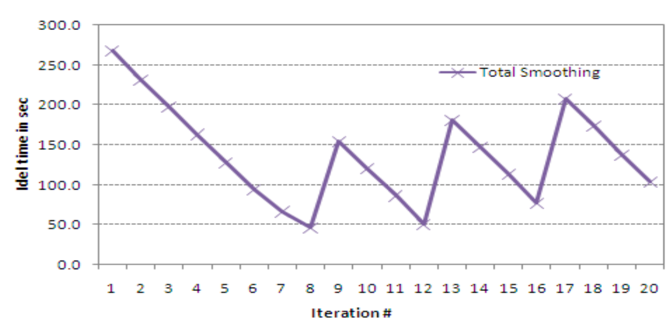

Figure 5: Total Idle time of stations for all the models for each iteration

By looking at the trend of the bottoms of the profile in Fig. 4, it can be noticed that by increasing the number of iterations, the minimum idle time may not become less the 46.8. Hence, it can be concluded that the number of iterations required to obtain the least idle time is dependent on the type of problem, number station, number of models and their quantity, which is difficult to pre-guess.

Table 8: Feasible solutions for 10 stations for the Buxey 29 Tasks Problem.

Task ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Solution 1	1	3	2	2	2	3	1	4	2	4	5	6	6	5	8	5	6	7	8	7	8	8	9	10	1	3	4	9	10
Solution 2	1	1	2	2	3	4	1	5	4	4	6	4	3	7	5	8	7	8	6	7	6	8	9	10	2	3	3	9	10
Solution 3	1	3	1	5	5	4	1	5	1	2	7	2	6	2	3	6	8	6	4	8	4	7	9	10	2	4	4	9	10
Solution 4	1	1	4	4	5	2	1	5	2	2	7	3	6	4	3	6	8	6	4	8	6	7	9	10	2	2	3	9	10
Solution 5	1	1	2	2	5	3	1	5	3	3	7	3	6	3	4	6	8	6	4	8	6	7	9	10	2	3	4	9	10
Solution 6	1	1	2	2	2	3	1	4	3	3	5	3	6	3	8	6	7	6	8	7	8	8	9	10	4	2	5	9	10
Solution 7	1	2	3	3	5	3	1	5	1	1	6	1	6	6	4	7	8	7	4	8	5	7	9	10	2	3	4	9	10
Solution 8	1	2	1	2	4	3	1	5	1	2	6	3	4	7	5	8	7	8	6	7	6	8	9	10	3	2	4	9	10
Solution 9	1	1	2	2	4	3	1	4	3	4	5	3	5	5	6	7	6	8	7	7	7	8	9	10	2	3	3	9	10
Solution 10	1	1	3	3	3	2	1	4	4	4	5	2	5	5	8	7	6	7	8	6	8	8	9	10	2	3	4	9	10
Solution 11	1	2	2	3	4	3	1	5	1	1	6	1	4	3	5	8	7	8	6	7	6	8	9	10	3	4	4	9	10
Solution 12	1	2	2	3	4	4	1	5	1	3	6	3	4	5	5	8	7	8	6	7	6	8	9	10	1	3	3	9	10
Solution 13	1	1	2	2	4	3	1	4	2	4	6	2	5	7	5	8	7	8	5	7	5	8	9	10	3	2	3	9	10
Solution 14	2	1	2	3	4	2	3	4	3	4	5	7	5	5	8	7	6	7	8	6	8	8	9	10	3	1	1	9	10
Solution 15	1	1	2	2	4	3	1	4	2	4	5	2	5	5	8	7	6	7	8	6	8	8	9	10	3	2	3	9	10

Again, by increasing the number of stations from 9 to 10, 15 feasible solutions are obtained. From Table 9 the best solution yields 327 sec of total time and a cycle time of 34 sec. By increasing the number of stations from 8 to 10, there in change in the number of feasible solutions and the same kind behavior is noticed when the number of stations further increased to 11, 12 and so on. Although the cost of installation of stations increases when the number of stations is increased, it provides the best flexibility in maintenance of the stations. If the cost of installation of the stations is given priority, it is the 8 station model, which suits best for this problem over 9 and 10 stations model.

Table 9: Total time taken by each of 10 stations for the Buxey 29 Tasks problem

Solution ->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Station 1	30	34	37	37	37	37	37	47	47	47	37	43	57	52	57
Station 2	39	39	35	36	52	42	49	36	42	39	53	53	39	47	39
Station 3	30	34	43	42	38	38	44	44	38	51	46	38	47	44	57
Station 4	32	30	35	48	54	58	54	51	56	44	43	50	54	54	54
Station 5	33	37	35	45	45	46	31	32	35	35	35	35	36	37	37
Station 6	34	34	32	34	34	34	37	36	33	32	36	36	36	32	32
Station 7	29	33	42	42	42	41	31	33	33	38	43	43	35	35	41
Station 8	35	39	44	44	44	35	38	39	40	35	39	39	39	35	35
Station 9	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
Station 10	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
Total Time	327	345	368	393	411	396	386	383	389	386	397	402	408	401	417

By executing the algorithm from steps r to z, the idle times for the each model are obtained for each iterations as shown in table 10. The procedure is same as mentioned and the least of the total idle times is obtained during iteration 7, for which the quantity of models are 10, 14, 12 and 10 respectively. The idle times for each model and the total idle time can be viewed from Figs. 6 and 7. The total idle time is 36.2 sec and this is slightly lower than that of 8 and 9 stations model, which are 38.8 and 46.8 sec respectively.

Table 10: Idle time of 10 stations and the quantity of models for each iteration

	Smoothing-Model 1	Smoothing-Model 2	Smoothing-Model 3	Smoothing-Model 4	Total Smoothing	Model 1 - Qty	Model 2 - Qty	Model 3 - Qty	Model 4 - Qty	Incrementing model
Iteration 1	14	108	53	1	176.0	10	10	10	10	0
Iteration 2	14	77.8	53	1	145.8	10	11	10	10	2
Iteration 3	14	47.6	53	1	115.6	10	12	10	10	2
Iteration 4	14	47.6	22.3	1	84.9	10	12	11	10	3
Iteration 5	14	17.4	22.3	1	54.7	10	13	11	10	2
Iteration 6	14	17.4	8.4	1	40.8	10	13	12	10	3
Iteration 7	14	12.8	8.4	1	36.2	10	14	12	10	2
Iteration 8	46.4	12.8	8.4	1	68.6	11	14	12	10	1
Iteration 9	78.8	12.8	8.4	1	101.0	12	14	12	10	1
Iteration 10	111.2	12.8	8.4	1	133.4	13	14	12	10	1
Iteration 11	143.6	12.8	8.4	1	165.8	14	14	12	10	1
Iteration 12	176	12.8	8.4	1	198.2	15	14	12	10	1
Iteration 13	208.4	12.8	8.4	1	230.6	16	14	12	10	1
Iteration 14	240.8	12.8	8.4	1	263.0	17	14	12	10	1
Iteration 15	273.2	12.8	8.4	1	295.4	18	14	12	10	1
Iteration 16	305.6	12.8	8.4	1	327.8	19	14	12	10	1
Iteration 17	338	12.8	8.4	1	360.2	20	14	12	10	1
Iteration 18	370.4	12.8	8.4	1	392.6	21	14	12	10	1
Iteration 19	402.8	12.8	8.4	1	425.0	22	14	12	10	1
Iteration 20	435.2	12.8	8.4	1	457.4	23	14	12	10	1

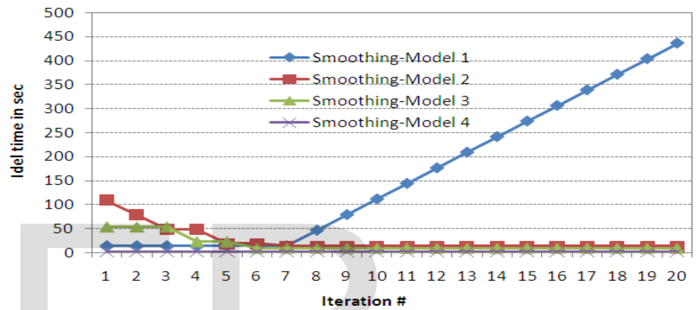

Figure 6: Idle time of stations for each iteration

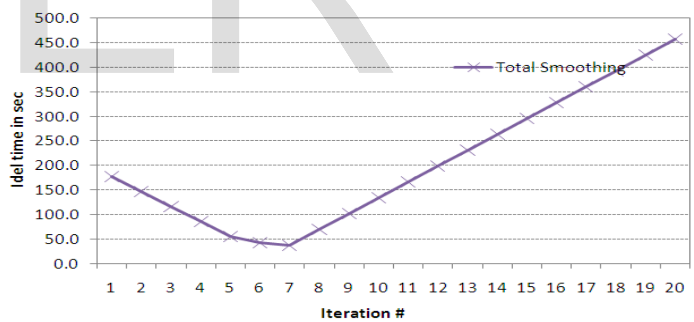

Figure 7: Total Idle time of stations for all the models for each iteration

The profile of the total idle time for the 10 station model is totally different than that of 8 and 9 stations model. The 8 and 9 station model has saw-teeth profile, where as the 10 station model has a V-shaped profile over 20 iterations. Hence it is difficult to obtain a mathematical prediction for the total idle time and only an iteration solution procedure like the one explained above can provide the best and fastest results.

In this work, multi mixed model assembly line problem is solved for best combination of number of station, number of models and quantity of each model by developing an algorithm. The algorithm has two parts, the first part considers the average time of each task for all the models to derive the feasible solutions. In this work, 8, 9 and 10 station models are solved which has resulted a least total cycle time of 327 sec in all the cases. Hence one can choose any of these models, but the cost of other resources in maintaining this number of stations to be given consideration. In second part of the algorithm, for each of these models, the total idle time is calculated. The results show that, the total idle time is 38.8, 46.8 and 36.2 sec for 8 station, 9 station and 10 station models respectively. However, if the cost of maintenance is given importance along with the lower idle time, one can choose the 8 station model since it has the second lowest idle time with least number of station. The quantity for each model on the 8 station model for least total idle time is 15, 15, 15, and 13 respectively. It is also studied about the number of iterations required to obtain the least total idle time. It is concluded that, the number of iterations required to obtain the least total idle time is dependent on type of problem, number of station, number of models and their quantity, which is difficult to pre-guess. Hence it is difficult to obtain a mathematical prediction for the total idle time and only an iteration solution procedure like the one explained above can provide the best and fastest results.

4.Conclusion

In this work, multi mixed model assembly line problem is solved for best combination of number of station, number of models and quantity of each model by developing an algorithm. The algorithm has two parts, the first part considers the average time of each task for all the models to derive the feasible solutions. In this work, 8, 9 and 10 station models are solved which has resulted a least total cycle time of 327 sec in all the cases. Hence one can choose any of these models, but the cost of other resources in maintaining this number of stations to be given consideration. In second part of the algorithm, for each of these models, the total idle time is calculated. The results show that, the total idle time is 38.8, 46.8 and 36.2 sec for 8 station, 9 station and 10 station models respectively. However, if the cost of maintenance is given importance along with the lower idle time, one can choose the 8 station model since it has the second lowest idle time with least number of station. The quantity for each model on the 8 station model for least total idle time is 15, 15, 15, and 13 respectively. It is also studied about the number of iterations required to obtain the least total idle time. It is concluded that, the number of iterations required to obtain the least total idle time is dependent on type of problem, number of station, number of models and their quantity, which is difficult to pre-guess. Hence it is difficult to obtain a mathematical prediction for the total idle time and only an iteration solution procedure like the one explained above can provide the best and fastest results.

Acknowledgment

I would like to express my deep sense of gratitude and obligations to my respected Principal **Dr. Ashok. H. Biradar** Guru Nanak Dev Engineering College, Bidar and my respected Guide **Dr.D.Choudhary**, Professor and H.O.D Mechanical Engineering Department, Guru Nanak Dev Engineering College Bidar for the timely support, guidance and suggestions in completing this paper.

References

[01] Salveson, M.E. (1955). The assembly line balancing problem. J. Ind. Engng., 6 (3), 18-25.
[02] Gutjahr, A., Nemhauser, G., 1964. An algorithm for the line balancing problem. Management Science 11, 308–315
[03] Kilbridge, M.D. and L. Wester (1961). A heuristic method for assembly line balancing. Journal of Industrial Engineering, 12, 292-298.
[04] Helgeson, W. B. and D. P. Birnie (1961). Assembly line balancing using the ranked positional weight technique, J. Ind. Engng, 12 (6), 394-398.
[05] Hoffmann,T.R., 1963. Assembly Line Balancing with a Precedence Matrix. Management Science. Vol. 9.
[06] Mansoor, E.M., 1964. Assembly Line Balancing - An Improvement on the Ranked Positional Weight.
[07] Arcus A.L., 1966. COMSOAL: A Computer Method of Sequencing Operations for Assembly Lines.International Journal of Production Research. Vol. 4, No.4: pp.259-277.
[08] Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing, management Science, 32, 909-932.
[09]Jackson, J.R. (1956). A computing procedure for the line balancing problem. Management Science, 2.
[10] Bowman, E.H. (1960). Assembly line balancing by linear programming, Operations Research, 8 (3).
[11] Van Assche, F. and Herroelen,W.S., 1978. An Optimal Procedure for the Single-Model Deterministic.
[12] Mamoud,K.I., 1989. A Generalised Assembly Line Balancing Algorithm. PhD. Dissertation, University.
[13] Sarin, S.C., E. Erel and E.M. Dar-El., 1999. A Methodology for Solving Single-Model, Stochastic Assembly.
[14] Boysen, N., Fliedner, M., 2007.A versatile algorithm for assembly line balancing. European Journal of Operational.
[15]Scholl, A. and Klein, R., 1999. ULINO : Optimally balancing U-shaped JIT assembly lines. International Journal of Production Research. Vol. 37, No.4: pp.721-736.
[16] Becker, C.; Scholl, A., (2006), A survey on problems and methods in generalized assembly line balancing. European Journal of Opera- tional Research, 168(3), 694-715.
[17] Boctor, F., (1995), A multiple-rule heuristic for assembly line balancing. Journal of the Operational Research Society, 46, 62-69.
[18] Amen, M., (2001), Heuristic methods for cost- ori-

ISSN 2229-5518
ented assembly line balancing: A comparison on solution quality and computing time. International Journal of Production Economic.

[19] Pinnoi, A.; Wilhelm, W. E., (1998), Assembly system design: A branch and cut approach. Management Science, 44, 103-118.

[20] Bukchin, Y.; Rabinowitch, I., (2006), A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem form minimizing stations and task duplication costs. European Journal of Operational Research, 174, 492-508.

[21] Peeters, A., (2006), Linear programming based lower bound for the simple assembly line balancing problem. European Journal of Operations Research, 168(3), 716-731.

[22] Grabau, M. R.; Maurer, R. A.; Ott., D. P., (1997), Using simulation to generate the data to balance an assembly line. Proceeding of the 1997 Winter Simulation Conference, WSC, (Atlanta, GA, December 7-10), 733-738.

[23] McMullen, P. R.; Frazier, G. V., (1998), Using simulated annealing to solve a multi objective assembly line balancing problem with parallel workstations. International Journal of Production Research, 36(10), 2717-2741.

[24] Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing, Management Science, 32, 909-932.

[25] Ghosh S. and R. J. Gagnon., 1989. A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems. International Journal of Production Research.

[26] Erel, E. and S.C. Sarin., 1998. A Survey of the Assembly Line Balancing Procedures. Production Planning.

[27] Scholl, A.1993. Data of assembly line balancing problems. Retrieved from http://www.wiwi.uni-jena.de/Entscheidung/alb/.