Contents lists available at GrowingScience

Management Science Letters

homepage: www.GrowingScience.com/msl

Optimum assembly line balancing by minimizing balancing loss and a range based measure for system loss

Dilip Roy[a*] and Debdip khan[b]

[a]*Department of Business Administration, The University of Burdwan, Burdwan, W.B., India*
[b]*Department of Business Administration, Burdwan Raj College, Golapbag. Burdwan, India*

ARTICLEINFO

Article history:
Received 30 June 2010
Received in revised form
28 October 2010
Accepted 29 October 2010
Available online
29 October 2010

Keywords:
Balancing loss
Idle time
System loss
Measure for System loss
Slackness

ABSTRACT

An assembly line is a sequence of workstations, connected together by a material handling system, which is used to assemble components into a final product. The economic importance of assembly as a manufacturing process has led to extensive efforts for designing an assembly line to ensure efficiency and cost effectiveness of assembly operations. These efficiency and effectiveness can be maximized by minimizing balancing loss and system loss. The current work offers a range based measure for System loss and describes a technique for jointly minimizing balancing loss and system loss. To arrive at the optimum solution of the joint minimization problem, the same has been reduced to sequential marginal minimization problems and simulation technique has been installed for obtaining the optimal solution. For demonstration purpose, one example has been worked out to indicate the strength of the suggested method.

1. Introduction

A production process can be classified into three types, namely mass production, batch production and jobbing, as may be seen from Ray Wild (2004). During the era of mass production, the products were more or less homogeneous in nature, and the basic objective of a manufacturing unit was to achieve maximum market share by undertaking minimum cost production. In that context, mass production assumed high priority.

In view of the importance of mass production a couple of researchers have worked on the problem of assembly line balancing for arriving at an optimum solution. The initial analytical treatment of assembly line balancing is available in Bryton (1954). The linear programming approach to arrive at an optimum solution for this problem was introduced by Salveson (1955) and Bowman (1960). However, in view of the complicated nature of the problem their optimum solutions turned out to be impractical and occasionally unstable. These difficulties led to introduction of heuristic methods as may be seen from the works of Kilbridge and Wester (1961). Helgerson and Birnie (1961), Moodie and Young (1965) and Mansoor (1968) worked on the heuristic ranked positional weight technique

* Corresponding author. Tel. +919434331371 Fax +913422559774
E-mail addresses: debdip.khan@gmail.com (D. Khan)

doi: 10.5267/j.msl.2010.01.009

for solving assembly line balancing problem. Later another heuristic technique, namely MALB, for balancing large but single-model assembly line was suggested by Dar-El (1973).

In a related problem Charlton and Death (1969) suggested a general method for machine scheduling. Assembly line balancing using Best Bud search by Nevins(1972) and using simulation to generate the data to balance an assembly line by Grabau et al. (1997) are the two interesting usage of simulation approach available for solving the line balancing problem. Hoffman (1963), Mansoor and Yadin (1971) and Geoffrion (1976) used mathematical programming to give a clear insight into this complex but important problem. Van Assche and Herroelen (1979) proposed an optimal procedure for the single-model deterministic assembly line balancing problem. An integer programming procedure for designing an assembly system was used by Graves and Lamer (1983). In fact, in the mid 80's some researchers gave emphasis on application part like application of operational research models and techniques in flexible manufacturing systems (Kusiak,1986) and application of a hierarchical approach for solving machine grouping and loading problems of flexible manufacturing systems (see Stecke, 1986). McMullen and Frazier (1998) studied simulation approach to solve a multi-objective assembly line balancing problem with parallel workstations. The branch and cut approach was used by Pinnoi and Wilhelm (1998) to deal with the problem of system design. Nicosia et al. (2002) suggested a different type of optimization procedure for optimally balancing assembly lines with different workstations. Thus, one can see that during the last few decades several researchers have mostly made use of mathematical programming, simulation technique and heuristic methods for solving an assembly line balancing problem. A few new approaches have been suggested in the literature as may be seen from the survey of algorithms for the simple assembly line balancing problems by Baybars (1986). Subsequently, Whitley (1989) suggested the "GENITOR algorithm" and Scholl and Becker (2003) suggested State-of-the-art to deal with the balancing problem. Roy and Khan (2010) suggested a generic approach for designing of an assembly line where, with a given number of workstations, one can efficiently arrive at the desired solution under different methods of search like simulation, heuristic etc. These are the markedly new approaches against the general trend of mathematical programming, simulation technique and heuristic method.

1.1 Problem description

Unfortunately, the balancing problems studied in all the above mentioned methods are oriented towards minimization of balancing loss. But assembly lines involving human elements have a different pressing problem of system loss. It was stated in the literature (see Ray Wild, 2004) that system loss is more important than balancing loss. Variations in the idle times in different work stations may lead to a behavioral problem. Also, there may be system loss due to stochastic nature of the time elements. Our objective in this current work is to design an assembly line where dual objectives of minimization of balancing loss and of system loss can be met. For this purpose, we first propose a range based measure for system loss (RMS) and then install our optimization method through simulation approach. The procedure we propose here will be generic in nature and can be used with different types of balancing methods. We present the proposed approach in section 2. The concerned algorithm is presented in section 3 and finally section 4 presents the worked out example.

2. Notation and Methodology

2.1. Notation

K	number of jobs	
N	number of workstations	
T_i	task time or assembly time of ith job,	$i = 1, 2, \ldots..K$
L_j	idle time of jth work station,	$j = 1, 2, \ldots..N$

N_{min}	minimum number of workstations
C	cycle time
C_t	trial cycle time
C_{min}	minimum cycle time
S	slackness
S_t	slackness for trial cycle time C_t
R	range of idle times
B	balancing loss
M	Range based Measure of System loss (RMS) = R / (minimum idle time)

2.2. Methodology

The concept of balancing loss offers a measure of the efficiency of an assembly line. It is defined as the loss resulting from allocation of work elements to workstations and is given by

$$B = \left\{ \left(NC - \sum_{i=1}^{K} T_i \right) / NC \right\} .100\%$$

Over the years, the basic consideration for designing any assembly line has always been the balancing loss, *B*. The underlying objective is to minimize *B* subject to precedence constraints. However, the objective of minimizing *B* generally leads to multiple solutions, which may not be equally efficient in the sense of system loss. The choice of system loss as the sole objective suffers from similar difficulties. Our proposed work is a multi-objective one where minimization of balancing loss is to be addressed along with minimization of system loss. But there is no concrete measure of system loss, reported in the literature so far. So, our objective is to first suggest a measure of system loss. After the identification of a measure of system loss, our objective will be to minimize both the balancing loss and the system loss. We propose to handle this problem sequentially. Given these twin objectives we first try to find the feasible solutions with minimum balancing loss and then we suggest a method for reducing the system loss along with the balancing loss so that the resultant line balancing solution will minimize both balancing loss and system loss and enjoy wider acceptability and greater applicability. According to Raywild (2004), workers' variable operation time is the main cause for system loss. The high disruption in the system takes place in configurations where one workstation has no idle time and another workstation has high idle time. In contrast, if the workstations have idle times of nearly equal length the chance of disruption will be less. We may then tend to consider the difference between maximum idle time and minimum idle time as a measure of the system loss with the belief that a system will be stable if the idle time of each workstation is more or less the same. However, in the extreme case when there is no idle time for any of the work stations this difference will be zero but that situation will lead to high system loss. Thus, a minimum idle time is needed in the system. As the minimum idle time increases system loss decreases. Keeping these views in mind, system loss can be measured through range with lower value preferred over higher value and can also be measured through minimum idle time with higher value preferred over lower value. To make them unidirectional and suggest a unit free measure, we consider a combined range based measure of system loss as *M* = Range / minimum idle time.

Since cycle time is externally fixed for each workstation, idle time may significantly vary if we look at balancing loss only. So, our target will be to minimize the system loss due to uneven idle times for a particular set up that meets the balancing loss requirement. In place of sequentially minimizing balancing loss and system loss if one proposes to simultaneously minimize these dual objectives the solution set becomes unnecessarily large. It is easy to observe that minimization of balancing loss reduces the solution set to a manageable one. So, the problem will be easier to handle if we divide the dual objective problem into two stages, each stage having a single objective. Given a cycle time, one can arrive at the minimum number of workstations. Then, given this minimum number of workstation, we can generate a set of feasible solutions to line balancing problem each optimizing the balancing loss within a range of cycle time. Then, on each of these solutions, the condition of system loss can be applied and optimum solution to the overall problem can be obtained.

By definition, the length of work time, or operating time, for which a component is available at each workstation is known as cycle time (C). Now, given a choice of C, it may be noted that the theoretical minimum number of workstations, N_{min}, must satisfy the following condition where $N_{\min}$ is an integer :

$$\sum_{i=1}^{K} T_i / C \le N_{\min} \le \sum_{i=1}^{K} T_i / C + 1. \tag{1}$$

It may not be always possible to attain $N_{min.}$ value due to precedence constraints. Then, to arrive at the first feasible solution, the number of workstation may be sequentially increased. The above inequalities can be reformulated in terms of C if N_{min} is given. From the left hand side of the inequality (1), we have,

$$\sum_{i=1}^{K} T_i / C \le N_{\min} \ ,$$

for which an equivalent condition on C is

$$C \ge \sum_{i=1}^{K} T_i / N_{\min} ,$$

so that, the minimum value of C, C_{min}, will be the highest integer contained in

$$\sum_{i=1}^{K} T_i / N_{\min} + 1,$$

i.e.,

$$C_{\min} = \left[\sum_{i=1}^{K} T_i / N_{\min} + 1 \right].$$

The optimality of the assembly line with respect of balancing loss or the N_{min} value will not be disturbed, if the choice of cycle time C varies from C_{min} to C. Now, given a cycle time, C, one may conceptually start from C_{min} and move up to C to arrive at the set of feasible workstation configurations with the same balancing loss. In this process if C_t is the trial cycle time, one has to add a slackness S_t to each workstation to maintain the same cycle time C. This slackness S_t is calculated as $S_t = C - C_t$, where C_t is the trial cycle time satisfying the condition $C_{min} \le C_t \le C$. This sequential approach can generate a reasonably good set of alternative solutions with the same balancing loss and forcefully induce similar idle times. From this solution set, we next calculate the range of idle time for each solution. Range is the difference between the maximum and the minimum idle times among workstations in a particular configuration or set up. Using minimization of M = (range / minimum idle time) as a choice mechanism, the solution set can be reduced to an optimal one where both balancing and system losses will be the least. We present an algorithm to make the suggested procedure computationally functional. This algorithm will meet the dual objective of minimizing both the balancing and range based measure of system losses for any assembly line balancing problem and will help us deal with large and complex problems.

3. The Algorithm

1. Set cycle time *C*, determine the minimum number of workstations N_{min} and calculate the C_{min} value

2. Set the cycle time at C_{min}
3. Prepare List *U* which is the list of all uncovered tasks
   From the tasks of List *U* with no immediate predecessor or whose immediate predecessors have been undertaken, prepare List *R*. These tasks are ready for selection and from the tasks of List *R* having task time / assembly time less than that of cycle time List *A* is prepared which are acceptable for immediate inclusion.

4. Select a task from the List *A* randomly and reset the cycle time as $\{C_t$ – assembly time$\}$
5. Repeat steps 3 – 4 if the cycle time is more than the elemental time
6. If cycle time less than the elemental time then open a new work station and repeat steps 3 – 5
7. Calculate the *M* value of idle times after getting the complete distribution of tasks to workstations
   After each run, the current *M* value is compared with the previous least *M* value. If the new *M* value is less than the previous least *M* value, the new solution is stored as the basis for next comparison. Otherwise the new solution is discarded.

8. Increase the trial cycle time by one unit and repeat the entire exercise until it crosses *C* value. If *C* value is crossed go to step 10
9. Repeat steps 3 to 8 and check whether all the work elements have been assigned to specified number of workstations, if not, increase the value of N_{min} by 1, recalculate C_{min} and go to step 2
10. Print the solution and this is the best solution in terms of minimum balancing loss *B*, and minimum system loss, *M*

We have converted the above algorithm into *C* language program for our numerical study. It may be noted that by considering a large number of runs for each trial, one can reasonably arrive at the optimum solution whose M is minimum within a set of solutions with minimum balancing loss.

4. Example

For demonstration purpose, we consider a standard assembly line balancing problem, as presented in Ray Wild (2004).

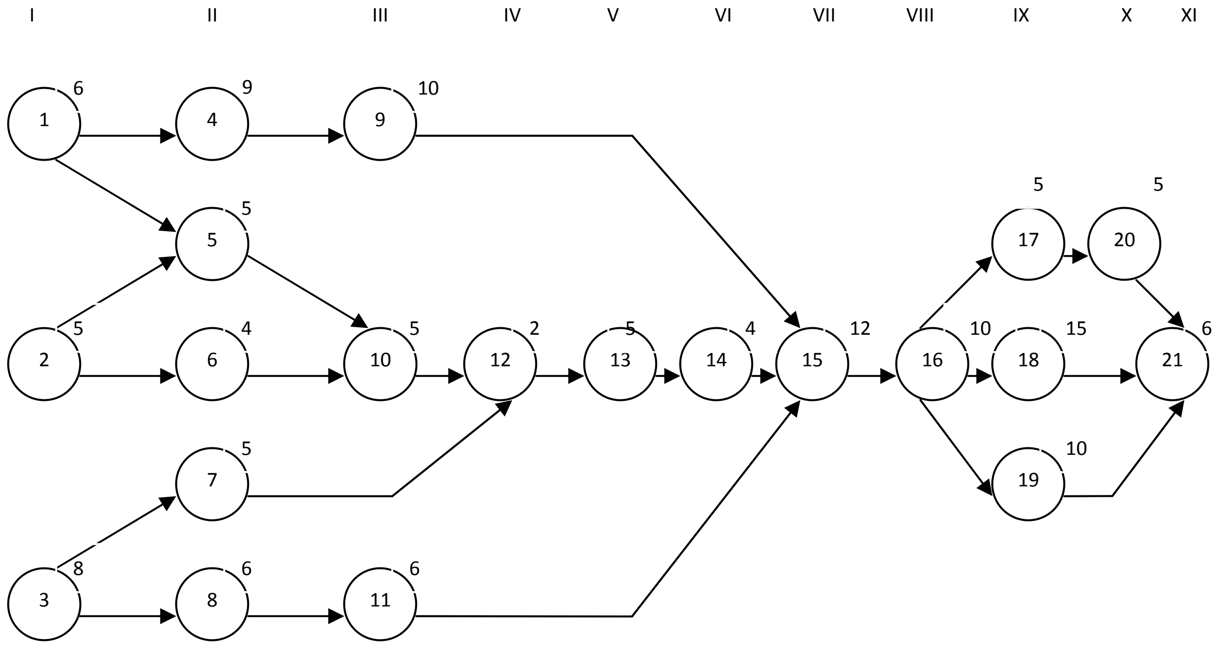

Fig. 1. Precedence diagram of work stations

There are 21 work elements in the example shown in Fig1. These work elements are governed by certain precedence constraints, as shown in the above figure. In the precedence diagram circles shows work elements and figures against them show task times. The underlying task is to arrive at an optimum configuration of the workstations, given a cycle time of 35 time units.

Then, for C=35, N_{min} works out as

$$N_{min} = [\ \sum T_i / C\ \ + 1\ ] = [\ 143/35 + 1] = 5.$$

For all these configurations, balancing loss will be the least as any configuration with *N*=5 will ensure the same *B* value. One can obtain C_{min} values in the following way to tackle the problem of system loss.

$$C_{min} = [\ \sum T_i / N_{opt} + 1\ ] = [\ 148/5\ ] = [\ 29.6\ ] = 29.$$

We consider a trial cycle time C_t which should start from C_{min} and proceed up to C = 35 for minimizing system loss. To ensure C = 35, we add to each work station a corresponding slack value $S_t = 35 - C_t$.

Thus, our iteration starts from C_t =29, with a slackness, S_{29} = 35 - 29 = 6. For C_t=29, no solution is available even after 25,000 runs. We next consider C_t=30 with S_t=5. There also, no feasible solution exists. This is consistent with our intuition that for very tight situations like C_t=29 and C_t=30, chance of generating a feasible solution is remote. We next consider C_t=31 with S_t=4. Here, we get 10 solutions in 25,000 runs. These solutions are described below in Table 1.

Table 1

Work station wise line balancing configurations with trial cycle time 31

	Work Station 1		Work Station 2		Work Station 3		Work Station 4		Work Station 5		Range	M
Sol^n	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time		
01	1,3,2,5,8	01	6,4,10,11,7,12	00	13,9,14,15	00	16,19,17, 20	01	18,21	10	10	2.50
02	1,3,8,2,7	01	6,11,5,10,4,12	00	13,9,14,15	00	16,17,19, 20	01	18,21	10	10	2.50
03	2,3,7,8,1	01	11,6,4,5, 10,12	00	9,13,14,15	00	16,18,17	01	20,19, 21	10	10	2.50
04	2,3,1,7,8,	01	5,11,6,10,12,13,14	00	4,9,15	00	16,18,17	01	20,19, 21	10	10	2.50
05	2,3,7,8,11	01	1,4,6,5,10,12	00	9,13,14,15	00	16,19,17, 20	01	18,21	10	10	2.50
06	3,1,7,8,11	00	2,6,5,10, 12,4	01	9,13,14,15	00	16,19,17, 20	01	18,21	10	10	2.50
07	3,7,8,1,2	01	5,6,10,12,13,14,11	00	4,9,15	00	16,19,17, 20	01	18,21	10	10	2.50
08	2,3,1,5,8	01	11,6,10,4,7,12	00	13,9,14,15	00	16,18,17	01	19,20 21	10	10	2.50
09	3,7,8,11,2	01	6,1,5,4,10,12	00	13,14,9,15	00	16,18,17	01	19,20, 21	10	10	2.50
10	1,2,3,8,11	00	4,6,7,5,10,12	01	13,14,9,15	00	16,17,18	01	20,19, 21	10	10	2.50

The minimum M value works out as 2.50. We next consider C_t=32 with S_t=3. For C_t=32 we get 7 solutions in 25,000 runs. These solutions are described in Table 2.

Table 2
Work station wise line balancing configurations with trial cycle time 32

	Work Station 1		Work Station 2		Work Station 3		Work Station 4		Work Station 5		Range	M
Sol^n	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time		
01	1,2,6,5,10	07	3,7,12,8,4	02	11,9,13,14	07	15,16,17, 20	00	19,18, 21	01	07	2.33
02	2,6,1,5,10	07	4,3,8,7,12	02	13,11,14,9	07	15,16,19	00	17,18, 20,21	01	07	2.33
03	2,6,1,5,10	07	3,8,11,4	03	7,12,9,13,14	06	15,16,19	00	18,17, 20,21	01	07	2.33
04	1,2,6,5,10	07	4,3,8,11	03	9,7,12,13,14	06	15,16,17, 20	00	19,18, 21	01	07	2.33
05	2,1,5,6,10	07	3,4,7,12, 13	03	14,8,11,9	06	15,16,17, 20	00	18,19, 21	01	07	2.33
06	2,6,3,7,1	04	5,10,12,13,8,14	05	4,11,9	07	15,16,17, 20	00	18,19, 21	01	07	2.33
07	2,1,6,3,5	04	7,10,12,13,14,8	05	11,4,9	07	15,16,19	00	18,17, 20,21	01	07	2.33

Here, we get the minimum of *M* value as 2.33. We next consider C_t=33 with S_t=2. For C_t=33 we get 17 solutions in 25,000 runs. These solutions are described in Table 3.

Table 3
Work station wise line balancing configurations with trial cycle time 33

	Work Station 1		Work Station 2		Work Station 3		Work Station 4		Work Station 5		Range	M
Sol^n	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time		
01	2,3,1,8,7	03	6,11,4,5, 10,12	02	9,13,14,15	02	16,18,17	03	19,20, 21	12	10	2.50
02	3,2,1,8,11	02	6,5,7,4,10,12	03	13,14,9,15	02	16,19,17, 20	03	18,21	12	10	2.50
03	3,7,8,2,11	03	6,1,5,10, 12,4	02	9,13,14,15	02	16,18,17	03	20,19, 21	12	10	2.50
04	3,1,7,8,2	03	4,11,5,6, 10,12	02	9,13,14,15	02	16,17,19, 20	03	18,21	12	10	2.50
05	1,2,5,3,8	03	6,4,7,11, 10,12	02	13,14,9,15	02	16,19,17, 20	03	18,21	12	10	2.50
06	3,8,1,2,11	02	7,4,6,5,10,12	03	9,13,14,15	02	16,19,17, 20	03	18,21	12	10	2.50
07	2,6,3,8,7	05	1,5,4,9	03	11,10,12, 13,14	11	15,16,17, 20	01	18,19, 21	02	10	3.33
08	3,7,2,6,8	05	1,4,9,5	03	11,10,12, 13,14	11	15,16,19	01	18,17, 20,21	02	10	3.33
09	3,8,1,4	04	9,2,5,6,10	04	11,7,12,13,14	11	15,16,19	01	18,17, 20,21	02	10	3.33
10	1,3,2,6,8	04	5,10,4,9	04	11,7,12,13,14	11	15,16,17, 20	01	19,18, 21	02	10	3.33
11	2,6,1,3,8	04	4,5,7,9	04	11,10,12, 13,14	11	15,16,19	01	18,17, 20,21	02	10	3.33
12	1,2,4,6,5	04	3,7,8,9	04	11,10,12,13,14	11	15,16,17, 20	01	19,18, 21	02	10	3.33
13	1,2,6,4,5	04	10,9,3,8	04	11,7,12,13,14	11	15,16,19	01	18,17, 20,21	02	10	3.33
14	2,1,4,5,6	04	10,9,3,4	04	7,11,12,13,14	11	15,16,17, 20	01	18,19, 21	02	10	3.33
15	2,6,3,8,1	04	4,5,7,9	04	10,11,12, 13,14	11	15,16,17, 20	01	19,18, 21	02	10	3.33
16	1,3,4,8	04	7,2,5,9,6	04	10,12,11, 13,14	11	15,16,17, 20	01	19,18, 21	02	10	3.33
17	2,1,4,6,5	04	3,7,10,12,13,14	04	9,8,11	11	15,16,19	01	18,17, 20,21	02	10	3.33

Here minimum of *M* works out as 2.50. We next consider C_t=34 with S_t=1. For C_t=34 we get 8 solutions in 25,000 runs. These solutions are described in Table 4.

Table 4
Work station wise line balancing configurations with trial cycle time 34

	Work Station 1		Work Station 2		Work Station 3		Work Station 4		Work Station 5		Range	M
Sol^n	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time		
01	3,7,8,1,11	03	2,6,5,4,10, 12	04	9,13,14,15	03	16,19,17, 20	04	18,21	13	10	2.50
02	3,7,8,11,1	03	2,5,6,10,4, 12	04	13,9,14,15	03	16,18,17	04	19,20, 21	13	10	2.50
03	3,2,8,1,11	03	7,5,6,4,10, 12	04	13,9,14,15	03	16,18,17	04	19,20, 21	13	10	2.50
04	2,3,8,11,1	03	6,5,7,10, 12,13,14	04	4,9,15	03	16,17, 20, 19	04	18,21	13	10	2.50
05	3,2,8,11,1	03	7,5,4,6,10, 12	04	13,9,14,15	03	16,19,17, 20	04	18,21	13	10	2.50
06	3,1,8,11,2	03	6,7,5,10, 12,13,14	04	4,9,15	03	16,17,18	04	20,19, 21	13	10	2.50
07	3,8,11,1,7	03	2,5,6,10, 12,13,14	04	4,9,15	03	16,19,17, 20	04	18,21	13	10	2.50
08	3,7,8,1,11	03	2,6,5,10, 12,13,14	04	4,9,15	03	16,18,17	04	20,19, 21	13	10	2.50

Here the minimum value of M is 2.50. We next consider C_t=35 with S_t=0. For C_t=35 we get 2 solutions in 25,000 runs. These solutions are described below in Table 5.

Table 5
Work station wise line balancing configurations with trial cycle time 35

	Work Station 1		Work Station 2		Work Station 3		Work Station 4		Work Station 5		Range	M
Sol^n	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time		
01	2,1,6,4,3	03	7,9,8,11,5	03	10,12,13, 14,15	07	16,19,17, 20	05	18,21	14	11	3.33
02	2,1,6,3,4	03	8,11,5,9, 10	03	7,12,13, 14,15	07	16,17,19, 20	05	18,21	14	11	3.33

Here the minimum value of *M* is 3.67. As stated earlier, our objective is to globally minimize the *M* value based on all such solutions. So the optimum configuration is given in Table 6 where cycle time is *C* = 35 and trial cycle time is C_t=32 with slack S_t=3.

Table 6
Work station wise line balancing final configurations

	Work Station 1		Work Station 2		Work Station 3		Work Station 4		Work Station 5		Range	M
Sol^n	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time	Elements	Idle time		
01	1,2,6,5,10	07	3,7,12,8,4	02	11,9,13,14	07	15,16,17, 20	00	19,18, 21	01	07	2.33
02	2,6,1,5,10	07	4,3,8,7,12	02	13,11,14,9	07	15,16,19	00	17,18, 20,21	01	07	2.33
03	2,6,1,5,10	07	3,8,11,4	03	7,12,9,13,14	06	15,16,19	00	18,17, 20,21	01	07	2.33
04	1,2,6,5,10	07	4,3,8,11	03	9,7,12,13,14	06	15,16,17, 20	00	19,18, 21	01	07	2.33
05	2,1,5,6,10	07	3,4,7,12, 13	03	14,8,11,9	06	15,16,17, 20	00	18,19, 21	01	07	2.33
06	2,6,3,7,1	04	5,10,12,13,8,14	05	4,11,9	07	15,16,17, 20	00	18,19, 21	01	07	2.33
07	2,1,6,3,5	04	7,10,12,13,14,8	05	11,4,9	07	15,16,19	00	18,17, 20,21	01	07	2.33

Above example shows that the proposed approach has led to simultaneous minimization of balancing and system losses. It may be noted that global minimum value of M is obtained when flexibility in the system is moderate and slackness is also moderate. Further, the M value increases in the extreme situations. Compared to the standard simulation approach proposed in COMSOAL and modified COMSOAL methods, our proposed method is better because it generates more feasible solutions by using the concept of varying cycle time and then adding slack time. This forcefully induces more uniformity among the idle times because a fixed slack value is added to each workstation.

5. Conclusion

The objective of this work was to develop an efficient solution for minimizing system loss and balancing loss of any assembly line balancing problem. For this purpose we have proposed a new measure for system loss, RMS, a Range based Measure of System loss. This solution aims to achieve a balanced distribution of work between different work stations. We have introduced the concept of trial cycle time, slack time in addition to cycle time so that more sets of feasible solutions can be obtained. Our approach is a generic one, which is capable of solving different line assembly problem with a reasonable computation time. From the trial solution sets, final choice is made based on optimum number of workstations and RMS value. In our approach we first consider minimization of balancing loss and then minimization of system loss. In this process we have reduced the simultaneous optimization problem to a sequential optimization problem. This has been done to keep the number of solution to a manageable form. This approach according to numerical study is giving a better set of configurations because we are using some amount of slackness in each workstation with trial cycle time varying from C_{min} to C.

References

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

Bowman, E. H. (1960). Assembly Line Balancing by Linear Programming. *Operations Research*, 8(3), 385-389.

Bryton, B. (1954). Balancing of a continuous production line. M.Sc.Thesis, North-Western University.

Charlton, J. M. & Death, C. C. (1969). A general method for Machine scheduling. *International Journal of Production Research*, 7, 207.

Dar-El (Mansoor), E.M. (1973). MALB - a heuristic technique for balancing large single model assembly lines. *AIIE Transaction*, 54, 343-356.

Geoffrion & Arthur M. (1976). The purpose of mathematical programming is insight, not numbers. *Interface*, 7(1), 81-92.

Grabau, M. R., Maurer, R. A., & Ott., D. P. (1997). Using Simulation to Generate the Data to Balance an Assembly Line. In Proceeding of the 1997 Winter Simulation Conference, WSC '97, Atlanta, GA, December 7-10, 733-738.

Graves, S. C. & Lamer, B. W. (1983). An integer programming procedure for assembly system design problems. *Operations Research*, 31, 522-545.

Helgerson, N.B. & Birnie, D.P. (1961). Assembly Line balancing using the ranked positional weight technique. *Journal of Industrial Engineering*, 11(6), 394.

Hoffmann, T.R. (1963). Assembly line balancing with precedence matrix. *Management Science*, 9, 551-562.

Keegan, W.J. (1995). *Global Marketing Management*. Prentice Hall of India, 351.

Kilbridge, K. & Wester, L. (1961). A heuristic method of assembly line balancing. *Journal of Industrial Engineering*, 11(4), 292.

Kusiak, A. (1986). Application of Operational Research models and techniques in flexible manufacturing systems. *European Journal of Operational Research*, 24, 336-345.

Mansoor, E. M. (1968). Assembly line balancing : A heuristic algorithm for variable operator performance levels. *Journal of Industrial Engineering*, 19, 618.

Mansoor, E. M. & Yadin, M. (1971). *On the problem of assembly line balancing. Development in Operations Research,* edited by B. Avi-ltzhak, Gordon and Breach, New York, 361.

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.

Moodie, C. L. & Young, H. H. (1965). A heuristic method of assembly line balancing assumptions of constant or variable work element times. *Journal of Industrial Engineering*, 16, 23-29.

Nevins, A.J. (1972). Assembly Line Balancing using Best Bud search. *Management Science*, 18(9), 530.

Nicosia, G., Paccarelli, D., & Pacifici, A. (2002). Optimally balancing assembly lines with different workstations. *Discrete Applied Mathematics*, 118, 99-113.

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

Wild, R. (2004). *Operation Management*. Thomson.

Roy, D. & Khan, D. (2010). Assembly line balancing to minimize balancing loss and system loss. *Journal of Industrial Engineering International*, 6(11), 1-5.

Roy, D. & Khan, D. (2010). Integrated model for line balancing with workstation inventory. *International Journal of Industrial Engineering Computation*, 1, 139-146.

Roy, D. & Khan, D. (2010). Optimum assembly line balancing: A stochastic programming approach. *International Journal of Industrial Engineering Computation*, 2, doi: 10.5267/j.ijiec.2010.04.001.

Salveson, M.E. (1955). The assembly line balancing problem. *Transaction of American Society of Mechanical Engineering*, 939.

Scholl, A. (1999). *Balancing and Sequencing of Assembly Lines*. 2nd Ed., Physica-verlag Heidelberg, ISBN 3-7908-1180-7.

Scholl, A. & Becker, C. (2003). State-of-the-art and heuristic solution procedures for simple assembly line balancing. *Jenaer Schriften zur Wirtschaftswissenschaft, FSU Jena*.

Stecke, K.E. (1986). A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems. *European Journal of Operational Research*, 24, 369-378.

Van Assche, F. & Herroelen, W.S. (1979). An optimal procedure for the single-model Deterministic assembly line balancing problem. *European Journal of Operational Research*, 3(2), 142.

Whitley, D. (1989). The GENITOR algorithm and selective pressure: why rank-based allocation of reproductive trials is best. In Proceeding of the Third International Conference on Genetic Algorithms and their Applications.