

# Line Balancing Techniques To Improve Productivity Using Work Sharing Method

Mahmud Parvez<sup>1</sup>, FariaBinta Amin<sup>2</sup>, Fahmida Akter<sup>2</sup>

<sup>1,2</sup> Department of Industrial Engineering and Management, Khulna University of Engineering & Technology, Bangladesh

---

**Abstract:** Line balancing is an effective tool to improve the throughput of assembly line while reducing bottleneck, cycle time. Line balancing is the problem of assigning operation to workstation along an assembly line, in such a way that assignment is optimal in some sense. This project mainly focuses on improving overall efficiency of single model assembly line by reducing the bottleneck activities, cycle time and distribution of work load at each work station by line balancing, using line balancing techniques mainly work sharing method. The methodology adopted includes calculation of cycle time of process, identifying bottleneck activities, calculating total work load on station and distribution of work load using code block (c++) software on each workstation also redesigning the layout by line balancing, in order to improve the efficiency of line and increase overall productivity.

**Keywords-** Bottleneck, Efficiency, Layout, Line Balancing, Work Sharing Method.

---

## I. Introduction

The production process of garments is separated into four main phases: designing or clothing pattern generation, fabric cutting, sewing, and ironing or packing. The most critical phase is the sewing phase, as it generally involves a great number of operations. The sewing line consists of a set of workstations in which a specific task in a predefined sequence is processed. In general, one to several tasks is grouped into one workstation[1]. Tasks are assigned to operators depending on the constraints of different labor skill levels. Finally, several workstations in sequence are formed as a sewing line. Shop floor managers are concerned with the balance of the lines by assigning the tasks to workstations as equally as possible. Unequal workload among workstations of a sewing line will lead to the increase of both WIP and waiting time, indicating the increase of both production cycle time and cost[2]. In practice, the sewing line managers or production controllers use their experience to assign tasks to workstations based on the task sequence, labor skill levels and the standard time required to complete each task. As a result, the line balance performance cannot be guaranteed from one manager to another with different assignment preference and/or work experience. Manufacturing a product always requires different types of sewing machines and different yarn colors, making it difficult to assign a worker to perform operations on just a single machine[3].

### 1.1 Concept of line balancing

The Line balancing is “design a smooth production flow by allotting processes to workers so as to allow each worker to complete the allotted workload within a given time. Line balancing is the problem of assigning various tasks to workstations, while optimizing one or more objectives without violating any restrictions imposed on the line.

### 1.2 Code blocks

Is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, code::Blocks is oriented towards C, C++, and Fortran. It has a custom build system and optional Make support.[4]

### 1.3 Workstation

It is an assigned location where a given amount of work is performed. Normally a workstation is manned by one operator only. Sometimes, work stations are manned by several operators, e.g. aircraft production line.

### 1.4 Cycle Time (CT)

Cycle Time may be defined as the ratio between the effective time available per period and the production volume per period. Effective time available = (Time per period) X per period (%Utilization of period) the cycle time may also be interpreted in the following ways: It is the time between consecutive releases of finished assembly's frail the last station of the line. It is the time between consecutive releases of semi - finished products between any two adjacent stations.[5]

### 1.5 Flexim Software

FlexSim is a powerful analysis tool that helps engineers and planners make intelligent decisions in the design and operation of a system. With FlexSim, you can build a 3-dimensional computer model of a real-life system, then study that system in a shorter time frame and for less cost than with the actual system. As a "what-if" analysis tool, FlexSim provides quantitative feedback on a number of proposed solutions to help you quickly narrow in on the optimum solution. With FlexSim's realistic graphical animation and extensive performance reports, you can identify problems and evaluate alternative solutions in a short amount of time. By using FlexSim to model a system before it is built, or to test operating policies before they are actually implemented, you will avoid many of the pitfalls that are often encountered in the startup of a new system. Improvements that previously took you months or years of trial-and-error experimentation to achieve can now be attained in a matter of days and hours using Flexim[6]

### 1.6 Why we use line balancing

All factories that have a line such as traditional assembly line and new assembly line such as heuristic and U-type and also mixed model used a few technique such as work sharing , genetic algorithms and fuzzy logic and also simulation method to improve a few parameter of line control in other hand ,manager like has a productivity and high yield in their factory and for this goal get help from previous technique to locate a machine ,employer ,assign employer to machine to select best choose for control and work by machine . In a few company one employer control 2 or more than 2 machines and this result is output of line balancing. In another word the company used line balancing for grow up the rate of production and decrease man power, idle time and buffer near machine, also used line balancing for produced more than 2 products.[7] Assembly lines are the most important components of mass production systems .The improved labor productivity is their essential significance for manufacturers who have to produce high volume products in a fast and cost effective manner. An assembly line consists of several successive workstations in which a group of assembly operations (tasks) are performed in a limited duration (cycle time). The productivity level of an assembly line generally depends on balancing performance. Assembly line balancing (ALB) is the problem of assigning tasks to successive workstations by satisfying some constraints and optimizing a performance measure. This performance measure is usually the minimization of the number of workstations utilized over the assembly line.

## II. Literature Review

Assembly line balancing is the problem of assigning various tasks to workstations, while optimizing one or more objectives without violating any restrictions imposed on the line. ALBP has been an active field of research over the past decades due to its relevancy to diversified industries such as garment, footwear and electronics. Ghosh and Gagnon (1989) as well as Erel and Sarin (1998) provided detailed reviews on these topics. Configurations of assembly lines for single and multiple products could be divided by three line types, single-model, mixed-model and multimodel. Single-model assembles only one product, and mixed-model assembles multiple products, whereas a multimodel produces a sequence of batches with intermediate setup operations (Becker & Scholl, 2006). This paper solves single-model line balancing problem with real application. Boysen, Fliedner, and Scholl (2007, 2008) classified ALBPs and pointed out that there were less than 5% articles explicitly solving line balancing of real world assembly systems. As a result, for practical consideration, this paper focuses on the real case of an assembly line in garment manufacturing. Configurations of assembly lines for single and multiple products could be divided by three line types, single-model, mixed-model and multi-model. Single -model assembles only one product, and mixed-model assembles multiple products, whereas a multi-model produces a sequence of batches with intermediate setup operations. A single-model line balancing problem with real application was solved in this project.[8]

ALBP with various objectives are classified into three types

- ALBP-I: Minimizes the number of workstations, for a given cycle time.
- ALBP-II: Minimizes the cycle time, for a given number of workstations.
- ALBP-III: Maximizes the workload smoothness, for a given number of workstations.

In type I problems, the ALBP of assigning tasks to workstations is formulated with the objective of minimizing the number of workstations used to meet a target cycle time. It can result in low labor costs and reduced space requirements. Type II problems maximize the production rate of an assembly line. Since this objective requires a predetermined number of workstations, it can be seen as the counterpart of the previous one. In general, shop managers are concerned with the workload equity among all workers. The issue of workload smoothing in assembly lines allocates tasks among a given number of workstations, so that the workload is distributed as evenly as possible. This problem is known as Type III problem.[9] Our project was focused on type-1 line balancing problem. Relevant data obtained from an apparel industry was used to formulate the solution. The objective of the project was to balance the cycle time for various operations and minimization of workstations.[10]

### III. Methodology

In order to balance a production line in sewing floor a line was chosen and necessary data was accumulated from the line.

#### 3.1 Work sharing method

An apparel garment order is chosen which was started in that line, knowing total amount of order, style description, fabric type and color. Two important attributes have been considered, one is possible standard method for each process and another is considerable time in between the input has been fed to the time study took to record the actual individual capacity of each worker. We have recorded the time to make each process for each and every worker to find out the number of operator and helper, type of machines and individual capacity. To find out the (standard minute value) S.M.V, process wise capacity has been calculated, in addition to that we have calculated the target, benchmark capacity, actual capacity line graph, labor productivity and line efficiency using code blocks(c++). After taking necessary data from the line we proposed a suitable line balancing technique for the line. At first we highlighted the bottleneck processes which were our prime concern and then seek solution to minimize the problem. In this project we proposed a method to balance the line by sharing workload among equally adept workers who has experience in both the bottleneck process and balancing process. Line has been balanced considering the bottleneck and balancing process where the balancing process has shared the excess time after the benchmark production in the bottleneck process. After balancing, new manpower has been proposed and final capacity of each worker has been reallocated. We have compared the line target after balancing the line, labor productivity and line efficiency. Finally a proposed production layout has been modeled with balanced capacity. And the layout is redesigned by using flexim software.

### IV. Equations

For Calculating Capacity:

$$\text{Capacity per hour} = 60 / \text{SMV} \dots\dots\dots (1)$$

Equation of Daily Production:

$$\text{Daily production} = \text{capacity per hour} * \text{working hour} \dots\dots\dots (2)$$

Tact Time of a Line:

$$\text{Tact time} = \text{Total SMV} \div \text{No. of operators} \dots\dots\dots (3)$$

Daily Target:

$$\text{Target} = (\text{Total manpower per line} * \text{total working minutes per day}) \div \text{SMV} \dots\dots\dots (4)$$

S.M.V. Calculation:

$$\text{Standard minute value (S.M.V)} = (\text{average cycle time} * \text{allowance}) \text{ in minute} \dots\dots\dots (5)$$

Manpower Requirement Is:

$$\text{Theoretical manpower} = \text{Benchmark target per hour} / \text{Process capacity per hour} \dots\dots\dots (6)$$

For Calculating Labour Productivity:

$$\text{Labor productivity} = \text{Total number of output per day per line} \div \text{number of workers worked} \dots\dots\dots (7)$$

Efficiency of a Line:

$$\text{Line efficiency} = (\text{Total output per day per lines} * \text{S.M.V}) \div (\text{Total manpower per line} * \text{total Working minutes per day}) * 100\% \dots\dots\dots (8)$$

### V. Data analysis & calculation

#### 5.1 collection of data

##### 5.1 collection of data

We collect the data for a single line of sewing process from Crony group of industries.

**Table 5.1:** data for a single

Seq.No.	operation	M/L Description	Cycle time(in sec)										Average in (min)	Performance rating	Normal Time (avg time * rating)	SMV (normal time * allowance)
			1	2	3	4	5	6	7	8	9	10				
1	F/B part match	HEL P	13	14	13	14	11	12	11	15	15	12	.22	.7	.15	.18
2	1st shoulder join	O/L	9	10	11	10	11	9	11	12	10	11	.17	.9	.15	.18

3	Neck binding	O/L	8	9	7	8	9	8	8	9	9	9	.14	.95	.13	.16
4	Thread cut& folding	HEL P	1 2	11	10	11	9	10	11	12	12	12	.18	.7	.12	.15
5	Neck top stitch	F/L	9	10	11	8	9	9	10	11	10	10	.16	.95	.15	.18
6	Shoulder cross tuck	S/N	8	9	10	8	9	9	10	8	9	10	.15	.9	.13	.16
7	2nd shoulder join	O/L	9	8	10	8	9	10	10	10	8	9	.15	.9	.13	.16
8	Shoulder out tuck	S/N	9	10	11	12	10	11	12	10	9	9	.17	.9	.16	.19
9	Sleeve pear	HEL P	8 9	7	8	9	8	8	9	9	9	9	.14	.9	.13	.14
10	Sleeve & body match	HEL P	1 3	13	14	14	12	15	13	14	13	12	.22	.7	.15	.18
11	Sleeve join	O/L	1 5	13	16	13	14	13	15	15	16	15	.24	.85	.20	.23
12	Thread cut & folding	HEL P	1 3	12	14	11	13	12	13	14	13	14	.28	.7	.15	.18
13	Side seam close	O/L	1 8	19	17	18	18	17	19	15	16	17	.3	.7	.21	.24
14	Thread cut & folding	HEL P	1 2	13	14	13	12	14	15	13	12	13	.21	.7	.15	.18
15	Label sewing	S/N	1 1	10	9	10	11	11	12	10	11	12	.17	.85	.14	.17
16	Bottom hem	F/L	8	9	8	9	10	10	9	9	10	10	.15	.95	.14	.17
17	Churi hem	F/L	1 5	13	16	13	14	13	15	15	16	15	.24	.95	.22	.27
18	Thread cut & folding	HEL P	1 5	13	16	13	14	13	15	15	16	15	.24	.7	.17	.20

### 5.2 Before line balancing

The first step of line balancing is to breakdown the operation into sequential logical order. The breakdown is done to better understand and implement the sequential order of product processing steps.

Taking cycle time for each operation is done manually and S.M.V is calculated from the average time with suitable allowance. Adding total S.M.V we can obtain target/hour. In this case 80% efficiency is the desired output level per hour. Before line balancing production scenario is illustrated in the figure 5.1

```

Capacity for Churi hem is :227.255
Capacity for Thread cut & folding is :308.417
SMU for Fb prt mtch is : 0.174417
SMU for Fst sldr join is : 0.1794
SMU for Neck binding is : 0.15295
SMU for Thread cut & fldng is : 0.147583
SMU for Neck top stitch is : 0.176621
SMU for Shoulder cross tuck is : 0.15525
SMU for Snd shoulder join is : 0.156975
SMU for Shoulder out tuck is : 0.177675
SMU for Sleeve pear is : 0.1449
SMU for Sleeve & body match is : 0.178442
SMU for Sleeve join is : 0.236229
SMU for Thread cut & folding is : 0.173075
SMU for Side seam close is : 0.23345
SMU for Thread cut & folding is : 0.175758
SMU for Label sewing is : 0.174321
SMU for Bottom hem is : 0.167517
SMU for Churi hem is : 0.264021
SMU for Thread cut & folding is : 0.194542
Total SMU : 3.26313
Pitch Time : 0.193889
Actual Output : 227
Line Balancing : 0.73437
Enter working hour : 8
Enter Organization Efficiency : .8
LineTarget : 2118.21
Line Efficiency : 68.663%
Bottleneck for Fb prt mtch is :70.0749
Bottleneck for Fst sldr join is :79.6305
Bottleneck for Neck binding is :21.7936
Bottleneck for Thread cut & fldng is :7.5287
Bottleneck for Neck top stitch is :74.3679
Bottleneck for Shoulder cross tuck is :27.6052
Bottleneck for Snd shoulder join is :31.8522
Bottleneck for Shoulder out tuck is :76.3834
Bottleneck for Sleeve pear is :0
Bottleneck for Sleeve & body match is :77.8343
Bottleneck for Sleeve join is :160.088
Bottleneck for Thread cut & folding is :67.4082
Bottleneck for Side seam close is :157.064
Bottleneck for Thread cut & folding is :72.7088
Bottleneck for Label sewing is :69.8857
Bottleneck for Bottom hem is :55.9054
Bottleneck for Churi hem is :186.824
Bottleneck for Thread cut & folding is :105.661
    
```

**Fig 5.1:** bench mark target, labor productivity and line efficiency before balancing line

**5.3 Bottleneck processes**

From Table 5.1 we have acknowledged some variations in process capacity from the bench mark target and the lower capacity with highest SMV from the bench mark target is the bottleneck process as production flow would stuck on the bottleneck point. Comparing total capacity of each process to the 80% bench mark target, we have identified the bottleneck processes named Thread cut & folding, Sleeve join, Churihem .

**5.4 Balancing Processes**

Sharing method is very crucial to make the production flow almost smoother compare to the previous layout. Considering working distance, type of machines and efficiency, workers who have extra time to work after completing their works, have been shared their work to complete the bottleneck processes. Previously identified three bottleneck processes have been discussed in the Table 4.2 .Thread cut & folding have been made by overlock machine and this have been shared by two overlock machine processes named shoulder join & neck binding. Sleeve join which also have been made by overlock machine and is sheared by shoulder cross tuck. Again Churi hem which has been made by flat lock machine is shared by other flat lock machine named bottom hem. Process 2 can work for 50min & 3 can work for 45 min& share with process 13 for last 10 &15 min respectively .Again process 6 can work for 45 min & share work with process 11 for last 15 min. And process 16 can work for 45 min & share work with process 17 for last 15 min

**Table 5.2:** balancing processes to equalize the bottleneck process

SI No.	Bottleneck process					Balancing process				
	Process name	Process No.	m/c type	Capacity /hr	Balanced capacity	Process name	Process No.	m/c type	Capacity /hr	Balanced capacity
1.	Thread cut & folding	13	O/L	257	300	1 <sup>st</sup> shoulder join	2	O/L	334	305
						Neck binding	3	O/L	392	332
#process 2 can work for 50min & 3 can work for 45 min& share with #process 13 for last 10 &15 min respectively										

SI No.	Bottleneck process					Balancing process				
	Process name	Process No.	m/c type	Capacity /hr	Balanced capacity	Process name	Process No.	m/c type	Capacity /hr	Balanced capacity
2.	11	Sleeve join	o/l	254	283	Shoulder Cross tuck	6	o/l	386	331
#process 6 can work for 45 min & share work with #process 11 for last 15 min										

SI No.	Bottleneck process					Balancing process				
	Process name	Process No.	m/c type	Capacity /hr	Balanced capacity	Process name	Process No.	m/c type	Capacity /hr	Balanced capacity
3	Churi hem	17	f/l	227	263	Bottom hem	16	f/l	358	294
#process 16 can work for 45 min & share work with #process 17 for last 15 min										

First column on both side of center table 5.2 shows the machine type and then followed by process no. process name, S.M.V , previous capacity and after balance capacity. After first process thread cut & folding has been come to process no. 2 named 1<sup>st</sup> shoulder join, then process no.3neck binding have been passed to process no. 13. Shoulder cross tuck then has been passed to process no. 11 and so on. In the proposed balancing process

machines of the same type are used for line balancing. Process no 2 and 3 are both overlock operations, process 13 and 11 are also done by overlock machine, process 16 and 17 are done by flat lock machine .So workers operating on the same machines are accustomed to the various operations done by the same machine. As a result they can share their work.

**VI. Result& Discussion**

**6.1 Proposed layout**

Using work-sharing method ,we have designed a proposed layout where process no. 13 have been shared by process no. 2&3.And process no. 11 has been shared by process no. 6.Again process no. 17 has been shared by process no. 16. We have denoted here the final layout of the line & the position of the machines in fig. 6.1 .

Balanced Capacity/ Hr	Previous capacity	SMV	Process name	Process No.	m/c		m/c	Process No.	Process name	SMV	Previous capacity	Balanced capacity
344	344	.18	f/b part match	1	Help		F/L	16	Bottom hem	.17	358	294
305	334	.18	1 <sup>st</sup> Shoulder Join	2	O/L		S/N	15	Label sewing	.17	344	344
333	392	.16	Neck binding	3	O/L		Help	14	Thread Cut& folding	.18	341	341
407	407	.15	Thread Cut & Folding	4	HELP		O/L			.24	257	300
340	340	.18	Neck Top Stitch	5	F/L		HEL P	12	Thread cut & folding	.18	347	347
331	386	.16	Shoulder Cross Tuck	6	S/N		O/L	11	Sleeve join	.23	254	283
382	382	.16	2 <sup>nd</sup> Shoulder Join	7	O/L		HEL P	10	Sleeve And Body match	.18	336	336
338	338	.19	Shoulder out tuck	8	S/N			9	Sleeve pear	.14	414	414

**Fig 6.1:** Result using code blocks

**6.2 Result using code blocks**

Using code blocks (c++), we show the calculation of actual output, line efficiency, line balancing that all are increased & the bottleneck of the line has been decreased. Before line balancing, our actual output were 227.So the production/day was: 1816.As our target is 2100, so it can't meet the target. Now we have got our final actual output 263 & daily production is: 2104 & it is increased by 288.

```

SMU for fb prt mch is : 0.174417
SMU for fst sldr join is : 0.19665
SMU for neck binding is : 0.180262
SMU for trd cut & folding is : 0.147583
SMU for neck top stch is : 0.176621
SMU for sldr x tuck is : 0.181125
SMU for nd sldr join is : 0.156975
SMU for sldr out tuck is : 0.177675
SMU for sleeve pear is : 0.1449
SMU for sleeve & body mch is : 0.178442
SMU for sleeve join is : 0.211792
SMU for trd cut & folding is : 0.173075
SMU for side seam close is : 0.199908
SMU for trd cut & folding is : 0.175758
SMU for label sewing is : 0.174321
SMU for bottom hem is : 0.203933
SMU for churi hem is : 0.227604
SMU for trd cut & folding is : 0.194542
Total SMU : 3.27558
Pitch Time : 0.193889
Actual Output : 263
Line Balancing : 0.851869
Enter working hour : 8
Enter Organization Efficiency : .8
LineTarget : 2110.16
Line Efficiency : 79.9532%
Bottleneck for fb prt mch is :70.0749
Bottleneck for fst sldr join is :108.968
Bottleneck for neck binding is :81.2307
Bottleneck for trd cut & folding is :2.5287
Bottleneck for neck top stch is :74.3679
Bottleneck for sldr x tuck is :82.8157
Bottleneck for nd sldr join is :31.8522
Bottleneck for sldr out tuck is :76.3834
Bottleneck for sleeve pear is :0
Bottleneck for sleeve & body mch is :77.8343
Bottleneck for sleeve join is :130.781
Bottleneck for trd cut & folding is :67.4082
Bottleneck for side seam close is :113.941
Bottleneck for trd cut & folding is :72.7008
Bottleneck for label sewing is :69.8857
Bottleneck for bottom hem is :119.865
Bottleneck for churi hem is :150.463
Bottleneck for trd cut & folding is :105.661
    
```

Fig 6.2: showing result using code blocks (C++)

### 6.3 proposed layout using flexim software

The layout we have acquired in work sharing method, is now redesigned by flexim software. We have run the line of each work station by flexim software to see from which station our product pass from other stations.

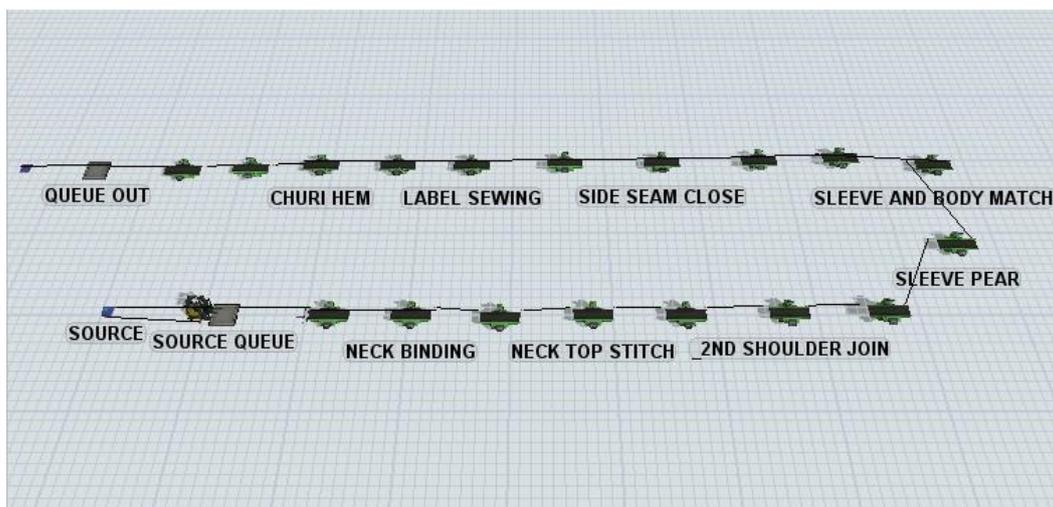


Fig 6.3: Proposed layout using Flexim

## VI. Conclusion

We consider the unbalancing problem in an industrial production line of production line. In order to recognize the bottleneck stations, the whole production line is calculated for different types of operations. In order to recognize the bottleneck stations, the whole production line is calculated for different types of operations by code blocks(c++) software. The calculation results provide a comprehensive insight for managers about backward and forward flows of the products in production line. Therefore, by changing the locations of stations and performing some improvements in activities of one station, the improvement scenarios are generated. Redesigning the improvement scenarios reveal that the confliction problems between backward and forward flows of the products are resolved. Moreover, the WIP products are greatly decreased and the bottleneck problem is eliminated. Our results show that the calculation using codeblocks(c++) based on statistical techniques is an effective method for balancing the complicated production lines in the real industrial cases.

### **7.1. Recommendations for Future Research**

Result would have been more effective if we would have taken some large quantity order and balancing the process is highly related to the type of machines as machine utilized in bottleneck and balancing process should be similar. Further improvements in the productivity can be achieved by considering large amount of order minimum 10000pieces. Table4.2 shows the new bench mark target which can be the further chance of improvements to balance the line with this new bench mark target. Proposed layout model has been followed the logic of modular system (one worker works more than two processes who is skilled on all processes and these combination of skilled workers finish their work in piece flow production) and traditional system (one worker works in one process and all the workers who may be skilled or not finish their work in bundle flow production) both together where only modular production system can be applicable with a series of skilled workers to achieve more productivity. On this occasion, skilled workers are eligible for the production processes and proper training and supervision is essential to achieve the optimum improvements on productivity. We can improve the flexim software to use it in many sections. We can also improve our code blocks programming for simulation of our line.

### **References**

- [1] James C. Chen , Chun-Chieh Chen , Ling-Huey Su , Han-Bin Wu, Cheng-Ju Sun “Assembly line balancing in garment industry” *Expert Systems with Applications* 39 (2012) 10073–10081.
- [2] Pereira “An enumeration procedure for the assembly line balancing problem based on branching by non-decreasing idle time” *European Journal of Operational Research* 229 (2013) 106–113.
- [3] Olga Battai ,AlexandreDolgui, “A taxonomy of line balancing problems and their solution approaches”, *Int. J. Production Economics* 142 (2013) 259–277.
- [4] Amardeep, T.M.Rangaswamy , Gautham j “Line balancing of single model assembly line” *Inter- national Journal of Innovative Research in Science, Engineering and Technology* Vol. 2, Issue 5, May 2013.
- [5] Md. RezaulHasanShumon, KaziArif-Uz-Zaman and Azizur Rahman “Productivity Improvement through Line Balancing in Apparel Industries” , *Proceedings of the 2010 International Conference on Industrial Engineering and Operations Man- agement* ,Dhaka, Bangladesh, January 9 – 10, 2010.
- [6] Agpak, K., &Gökçen, H. (2005). Assembly line balancing: Two resource constrained cases. *Inter- national Journal of Production Economics*, 96, 129– 140.
- [7] Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalizedassembly line balancing. *European Journal of Operational Research*, 168, 694–715.
- [8] Chen, J. C., Hsaio, M. H., Chen, C. C., & Sun, C. J. (2009). A grouping genetic algorithm for the assembly line balancing problem of sewing lines in garment industry. In *Proceedings of the eighth international conference on machine learning and cybernetics* (pp. 2811–2816).
- [9] J. Bautista, J. Pereira, Ant algorithms for atime and space constrained assembly line balancing problem, *European Journal of Operational Research* 177 (2006) 2016–2032.
- [10] M.D. Kilbridge and L. Wester, A Heuristic Method of Assembly Line Balancing, *J. Industrial Engineering*. 12 (1961) 292-298.