Systematic Pedagogy to Line Balancing with EXCEL

Thomas Lovejoy-Henkel[1], Jimin Lee[2], Donna Parsons[3], and Robert Yearout[1]

[1]Department of Management & Accountancy
[2]Department of Mathematics
The University of North Carolina at Asheville
One University Heights
Asheville, North Carolina 28804 USA

[3]Departmentof Business
Mars Hill University
Mars Hill, NC 28754

Corresponding author's Email: yearout@unca.edu

Author Note: Thomas Lovejoy-Henkel graduated from the University of North Carolina Asheville with a degree in Business Administration Management in May 2014. Jimin Lee, Assistant Professor of Statistics, has published many articles in statistics and bio-statistics. She has published in international industrial engineering journals and conference proceedings. Donna Parsons, Assistant Professor of Business, has published numerous business articles as well as industrial engineering journals and proceedings. Robert Yearout, Professor of Industrial Engineering and Management, has published a significant number of articles in national and international industrial engineering and occupational safety journals and proceedings.

Abstract: Over the past ten years, simple and inexpensive operations research software that is user friendly to the mentor, student, and instructor has become difficult to obtain. This is especially true since Emmons, Flowers, Khot, and Mathur's STORM 4.0 for Windows is obsolete for current 32 and 64 bit operating systems and no longer in print. After a diligent product and literature search, it appears there is no adequate inexpensive software that is easily available. Assembly line balancing algorithms are heuristic methods used for balancing operations or production lines. However, most methods employ complex calculations that are challenging to the mentor and mentee. This paper presents a pedagogy from a systems approach using Microsoft EXCEL. The object is to prepare a spreadsheet file with four separate worksheets that are linked to the first worksheet. The step-by-step systematic approach allows the entry on the main worksheet of data such as an annual demand, annual time available, and process times. When the user changes these data entry points, the efficiencies of each operating or production line are automatically re-computed for all three shifts. Worksheets use one of the several available heuristics to compute cycle times (required time between process activities) and transfers it to one, two, or three shifts (worksheets two, three, or four). Once the spreadsheet and accompanying worksheets were completed, the results were compared to several different heuristic algorithms. When authors were satisfied that the results were accurate and not significantly different from other examined algorithms, the final step was to develop a working pedagogy to efficiently describe the process. This allows the user an efficient analytical tool to illustrate and explain interactions within a given process. A local manufacturing facility used this method as a part of a monthly effort to increase line efficiency for individual workstations. The project's results were satisfactorily tested in a production operations class. The major advantage to the practitioner, engineer, instructor, and student is that EXCEL is readily available on all personal computers, easily understood, and is very practical. Students with very little exposure to line balancing were able to master the method within the first hour of exposure.

Keywords: Cycle Time, Efficiency, Line Balance, Pedagogy, EXCEL

1. Introduction

Line balancing is the process of assigning work to stations based upon the demand. Line balancing is an essential tool that assists the engineer, operations research specialist, consultant, and operations manager in optimizing a facility layout. It allows the user to optimize the procurement of equipment required for each task, the efficient use of available floor space, hiring of skilled workers, and the determination of the amount of annual time required to meet demand. For years, a multitude

of variations of software have been available to assist in performing this analysis. Emmons, Flowers, Khot, and Mathur's STORM 4.0 for Windows (2001) was adequate, user-friendly, and affordable. However, since STORM 4.0 (16 bit) is now obsolete for current 32 or 64 bit operating systems, no inexpensive, user-friendly, easily explained pedagogy software is available. This project presents pedagogy for line balancing using Microsoft EXCEL (2013). There are several different applications offered on-line that promote a systematic approach: however, the authors have found these to be relatively clumsy and difficult to explain to novices. In today's economic environment, demands over small time periods are becoming more common. The value of EXCEL is that it is a software package that is compatible with most computers, very user-friendly, easily adapted to an instructor's teaching style and pedagogy, and internationally accepted. In the workplace an easily defined EXCEL algorithm can be readily available to the engineer or operations manager.

1.1 Cycle Time

To perform a line balance, the cycle time must be calculated. The cycle time is the maximum amount of time that can be allowed to perform any specific task on any one workstation. It is not the time required to complete a task, but is the time required between units in a typical operation. If the time required to perform a specific task on any one workstation exceeds the cycle time, the work period demand will not be met. Thus, cycle time is expressed in time per unit as illustrated in Equation 1. Thus, cycle time is time between units, not time it takes to complete a task. Figure 1 illustrates Cycle Time.

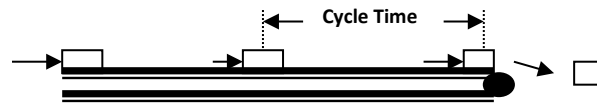Equation: *Cycle Time = Total Time Available/Demand for that Period*   (1)

Figure 1. Cycle Time Illustrated

Once the cycle time is computed and a sequence of tasks is completed, a precedence diagram is constructed (Figure 2). Then the cycle time is for the entire process is compared to the longest task time. If the time required to perform any one specific task exceeds the cycle time, then an additional component must be added to the workstation. There are many different algorithms that compute cycle time for any given operation. Computer-based techniques tend to give good, although not necessarily optimal results. Human judgment and pattern recognition often allow us to improve on computer-generated solutions. Thus manual methods are still the most prevalent in practice. The authors chose the *'Longest Work-Element Time Rule'* for the basic solution and picked the Largest Number of Followers Rule to determine if subsequent substations will allow improved efficiency (Krajewski and Ritzman, 2009 and Konz, 1985).

1.2 Balancing an Assembly Line

There are a multitude of potential solutions to an assembly line. However, the goals are to maximize efficiency, minimize idle time, minimize idle delay, and reduce bottle necks (Konz, and Johnson (2007); and Krajewski and Ritzman, 2009). Once the cycle time is calculated, processing times are determined for each workstation, and a precedence (process flow) diagram is completed. When the processing times are less than the cycle time the following workstations can be combined (Figure 2 and 3). Cycle Time for Station 1 and 2 is 60 seconds.

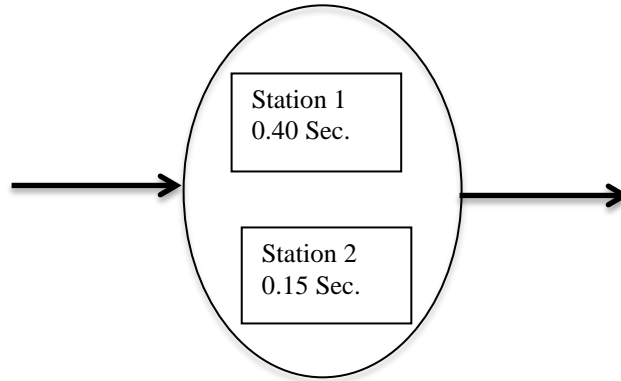Figure 2. Station 2 and Following Station 1 are Less Than Cycle Time

Figure 3. New Station 1 for Analysis

Note that there is a balance delay of 0.05 seconds.

When Process Time exceeds Cycle Time, the basic assembly line balancing algorithm must be modified. To modify the model, the analyst must make one of four decisions as follows.

a.   Increase the Cycle Time.
b.   Determine work methods that will reduce the processing times.
       Comment: If new machines are required one might attempt to locate more efficient machines.
c.   Add one or two shifts, which will increase the amount of total time available. Adding a shift will effectively increase the Cycle Time to a point where it exceeds Process Time.
d.   If the above mentioned options are not feasible, the analyst would increase the number of operators or current model machines in each workstation Figure 4). The figure 4 example has a cycle time of 60 seconds and the process time (PT) is 90seconds.

   ♦   At 60 seconds: Machine A has been in process for 60 seconds. Part B arrives at Machine B
   ♦   At 90 Seconds: Part leaves Machine A. Part in Machine B has been in process 30 seconds.
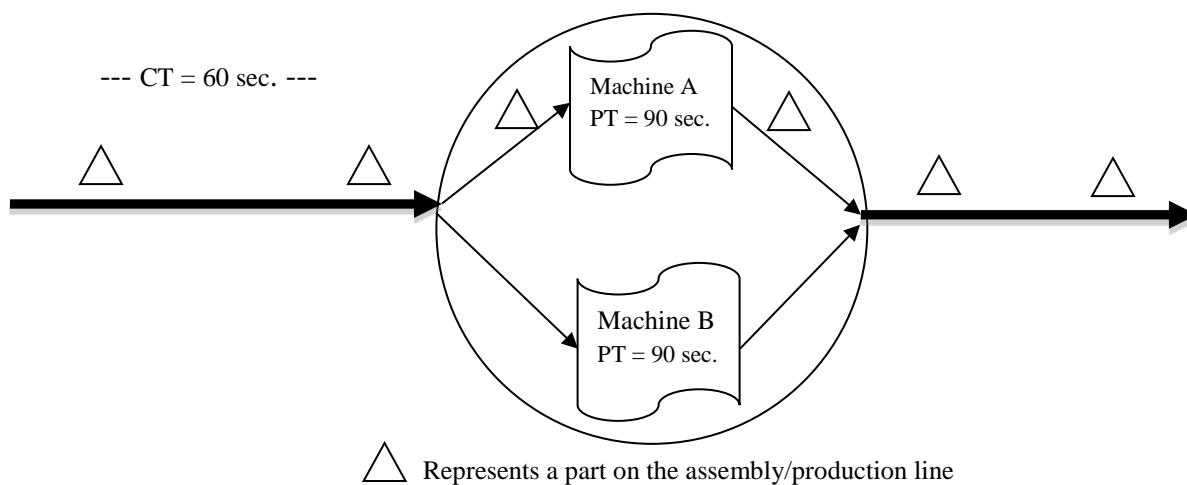   ♦   At 120 Seconds: Part leaves Machine B. This process continues.

Figure 4. Adding Machine where Cycle Time is Less than Processing Time

1.3 The Challenge

The challenge was to create a systematic algorithm and pedagogy that requires the user to enter inputs on a main worksheet (annual demand, workdays, hours worked per day, and task time in minutes or seconds) with the end goal of completing a working, functional line balancing model using EXCEL.

2. Methodology

This step-by-step procedure will address each cell that is contained in the four different spreadsheets. The interactive EXCEL file's first spreadsheet will be the first worksheet and is to be named *Main* (Lovejoy-Henkel, Lee, Parsons, and Yearout. 2010). Succeeding tabbed worksheets will be named as follows: Second worksheet is named *1 Shift,* third worksheet *2 Shifts,* and fourth worksheet *3 Shifts.* Comments in this paper will be addressed in the typical spreadsheet matrix where numeric values represent rows within the spreadsheet, while alphabetic characters represent columns. This project required four interlinking worksheets using standard EXCEL functions. A worksheet was completed for a main sheet and three different scenarios (1 shift per day, 2 shifts per day, and 3 shifts per day). Cycle time is calculated for a period demand and associated task times for an assigned operation. Next, each of the three worksheets were linked to the main worksheet, where inputs must be entered in order for the spreadsheets to produce results. Two scenarios will be examined; that of which process time is greater than cycle time (scenario 1(Krajewski and Ritzman, 2009)) and process times are less than cycle time (scenario 2). In each of the algorithms cell that are in *Red* requires user input. The spreadsheet is so designed that only minimum entry is required on each spreadsheet. The user must enter task times, demand, workweeks per year, and hours worked per day on the main worksheet. This allows the user to manually change any one input on the main worksheet which will result in the automatic re-calculation of the other three worksheets.

2.1 Algorithm Development (Main Tab Data Entry)

To construct the 'Main' algorithm, open an EXCEL spreadsheet and title its accompanying tab *Main*. Table 1 illustrates the Main tab where data entry will occur to allow proceeding worksheets to be linked to this worksheet. This will allow the user to manually change any one input on the main worksheet which will, result in the automatic re-calculation of the other three worksheets.

Table 1. Data Entry for Cycle Time Computations (Main Tab)

	A	B	C
1	Demand/Year =	5,000,000	
2	Number Of Weeks =	50	
3	Number of Days/Week =	5	
4	Hours per Day =	8	
5	Demand/Day =	20,000	
6	*[1]Meal Time on Employer (Yes (1) or No (0))	1	
7	Maintenance =	1%	
8	Set-Up Times =	2%	
9	*[2]Estimated Inefficiency =	6%	
10	Work Station	Task Time (Minutes)	Task Time (Seconds)
11	A	1.200	72.000
12	B	0.350	21.000
13	C	1.100	66.000
14	D	0.750	45.000
15	E	0.800	48.000
16	F	1.400	84.000
17	G	0.900	54.000
18	H	0.490	29.400
19	I	0.780	46.800

Table 1A. Table 1 (Continued) Results Summary for Cycle Time in Minutes

	A	**B**	**C**	**D**	**E**
21	Number of Shifts	Total Employees	Efficiency	Inefficiency	Cycle Time (Min)
22	1 Shift	411	98.83%	1.07%	0.01911
23	2 Shifts	207	98.21%	1.79%	0.03822
24	3 Shifts	140	96.81%	3.19%	0.05733

Note: If the cycle time is desired in seconds then minor alterations can be made in table 1.A

      The purpose of the results summary (Table 1 A) is to connect all results from worksheets 2, 3, and 4 to the Main tab so that all results are displayed on the Main tab. Included in the detailed results are total employees, efficiencies and inefficiencies of each assembly/production line, and cycle time dependent on number of shifts.

      To obtain a working line balance model, the user should enter each of the command and labels as referenced by its accompanying cell. For example, in cell A1 enter '*Demand per Year =*' in cell A2 enter '*Number Of Weeks =*'. Numeric values and reference equations of functions will be entered in the appropriate cells in 'Red'. Such as in cell B1 enter the numeric '*5,000,000*.

    Notes: *[1]If the company provides the meal and the 8 hours is not extended by 30 minutes enter '1'.
      *[2]This is a catchall such as absenteeism or other factors which will reduce the available time.

Table 1.B provides the required equations and entry algorithm for the Main Tab.

Table 1B. Algorithm Equations and Entry for Main Tab

B1: Enter numeric demand per year.	C11. Enter equation ' =(B11*60)'. Drag down to Cell 'C19'.
B2: Enter numeric number of operating weeks per year.	C22. Enter equation' =(1Shift!B28)'
B3: Enter numeric number of operating days per week.	C23: Enter equation '=(2Shifts!B28)'
B4: Enter numeric number of hours per day.	C24: Enter equation '=(3Shifts!B28)'.
B5. Enter equation ' =(B1)/(B2*B3)'.	D22. Enter equation '=(2Shifts!E28)'.
B6: Enter numeric '1' or '0' for '1'Meal.	D23. Enter equation '=(2Shifts!E28)'.
B7.: Enter numeric for maintenance times as a percent	D24. Enter equation ' =(3Shifts!E28)'.
B8. Enter numeric for set-up times as a percent.	E22. Enter equation '=(2Shifts!B11)'
B9: Enter numeric for estimated inefficiency as a percent.	E23: Enter equation '=(2Shifts!B11)'.
B11: Enter task time in minutes for each individual. Drag down to Cell 'B19'.	E24: Enter equation '=(2Shifts!B11)'.
B2: Enter equation '=(1Shift!D26)'	
B23. Enter equation ' =(2Shifts!D26)'.	
B24. Enter equation '=(3Shifts!D26)'.	

2.2 Algorithm Development for One Shift

      Once the Main tab has been completed, data entry can begin on the next tab (worksheet 2), which is to be named *1 Shift* (Table 2*).* This tab will include all cycle time calculations for 1 shift per day. For cells A1 through A28 enter labels as you did for the main menu. For example, in cell A1 enter '*1 Shift /Day'*. Numeric values and reference equations of functions will be entered in the appropriate cells in 'Red'. Table 2 is the calculations display.

Table 2. First Shift Calculations and Entry Display

	A	B	C	D	E	F	G
1	*1 Shift/Day*		**Break Type**	**# Of Breaks**	**Min/Break**	**Meal/Shift**	**Total**
2	Demand/Shift =	20000	**Breaks**	2	15	30	60
3			**Meal**	1	30	30	
4				**Maintenance**	**Set-Up**	**Inefficiency**	**Total**
5			**Efficiency**	0.01	0.02	0.06	0.10
6	**TIME**						
7	Hours/Day =	8	**Break Time**	**w/ Break**	**Efficiency**	**Available**	
8	Min/Day =	480	60	420	0.90	378	
9	Sec/Day =	28800	3600	25200	0.90	22680	
10	*Cycle Time*						
11	*Minutes*	*0.0189*	**Min/Unit**				
12	*Seconds*	*1.134*	**Sec/Unit**				
13							
14	**Work**	**Task**	**Number of**		**Idle Time**	**Percent (%)**	**Percent (%)**
15	**Station**	**(Min)**	**Machines**	**Assigned**	**(Min)**	**Efficiency**	**Idle**
16	A	1.200	62.794	63	0.004	99.674	0.326
17	B	0.350	18.315	19	0.013	96.395	3.605
18	C	1.100	57.561	58	0.008	99.244	0.756
19	D	0.750	39.246	40	0.014	98.116	1.884
20	E	0.800	41.863	42	0.003	99.674	0.326
21	F	1.400	73.260	74	0.014	99.000	1.000
22	G	0.900	47.096	48	0.017	98.116	1.884
23	H	0.490	25.641	26	0.007	98.619	1.381
24	I	0.780	40.816	41	0.004	99.552	0.448
25							
26	*Total*	**7.770**	**406.593**	**411**			
27							
28	*% Efficiency*	**98.928**		*% Inefficiency*	**1.072**		

Table 2.A provides the required equations and entry algorithm for the 1 Shift Tab.

2.3 Algorithm Development for 2 and 3 shifts

Now that you have completed worksheet 2 (1 Shift) copy worksheet 2 (1 Shift*)* in its entirety, and paste it to cell A1 on worksheet 3 (2 Shifts) and worksheet 4 (*3* Shifts*)*. Making these simple and quick changes will result in the entire assembly line balance change. Note how resulting values in Table 3 are different than those in Tables 1 and 2. The above four changes will automatically re-compute all calculations on both this page and on the Main page results. The changes are listed below.

For 2 Shifts (Table 3)

A1: Label Worksheet 3: 2 Shifts Per Day.
B2: Enter equation '=SUM(Main!B5)*(0.5)'.
B7. Enter equation '=SUM(Main!B4)*2.
D4. Enter the numeric '4'.
D5. Enter the equation '=(Main! B6*2)'.

For 3 Shifts (Table 3)

A1: Label Worksheet 4 : 3 Shifts Per Day
B2: Enter equation'=SUM(Main!B5)*(0.333)'.
B7. Enter equation '=SUM(Main! B4)*3.
D4. Enter the numeric '6'.
D5. Enter the equation '=(Main! B6*3)'.

Table 2A. Equations for the First Shift (1 Shift) Entry

B2: Enter equation '= (Main!B5)'.	E2: Enter break time '15' minutes.
B7: Enter equation '=(Main! B4)'.	E2: Enter break time '30' seconds.
B8: Enter equation '=(B7*60)'.	E5: Enter equation'=(Main! B8)'.
B9: Enter equation '=(B8*60)'.	E8: Enter equation '=(1-G5)'. Drag down to E9.
B11: Enter equation '=(F8/$B$2)'. Drag down to B12.	E16: Enter equation '=((D16*$B$11) – (B16))'. Drag down to E24.
B16: Enter equation ' =(Main! B11)'. Drag Down to B24.	E28: Enter equation '=(1 – B28)'.
B26: Enter equation '=SUM(B16:B24)'.	F2: Enter equation '=(D2*E2)'. Drag down to F3.
B28: Enter equation'=SUM((B26)/(D26*$B$11))'.	F5: Enter equation '=(Main!B9)'.
C8: Enter equation '=(G2)'.	F8: Enter equation '=(D8*E8)'.
C9: Enter equation '=(C8*60)'.	F16: Enter equation '=((B16)/(D16*$B$11)). Drag down to F24.
C16: Enter equation '=(B16/$B$11)'. Drag down to C24.	G2: Enter equation '=(F2+F3)'.
C26: Enter equation '=SUM(C16:C26)'.	G5: Enter equation 'SUM((D5:F5)'.
D2: Enter either a '2' for the number of breaks.	G16.Enter equation '=(1-F16). Drag down to G24'.
D3: Enter equation '=(Main! B6)'.	
D5: Enter equation'=(Main! B7)'.	
D8: Enter the equation '=(B8-C8)'. Drag down to D9.	
D16: Enter equation '=ROUNDUP(C16,0)'. Drag down to D24.	

Table 3. Minor Changes Required for 2 and 3 Shifts

	A	B	C	D	E	F	G
1	*2 Shifts/Day*		**Break Type**	**# Of Breaks**	**Min/Break**	**Meal/Shift**	**Total**
2	Demand/Shift =	10000	**Breaks**	4	15	60	120
3			**Meal**	2	30	600	
4				**Maintenance**	**Set-Up**	**Inefficiency**	**Total**
5			**Efficiency**	0.01	0.02	0.06	0.09
6	**TIME**						
7	Hours/Day =	16	**Break Time**	**w/ Break**	**Efficiency**	**Available**	
8	Min/Day =	960	120	840	0.91	764.4	
9	Sec/Day =	57600	7200	50400	0.91	45864.0	

3. Analysis

Listed below are two example scenarios, the first of which is a scenario that involves a cycle time being greater than process time. This example can be seen in section 3.1 (manual entry), where work elements have been grouped into stations. The manual entry calculations are listed in Table 4 and Table 5 gives the results of the spreadsheet.   After the calculations were completed, they were tested against the spreadsheet developed in this paper. Both models resulted in identical efficiencies and balance delays.

3.1 Scenario 1 (Process Time < Cycle Time)

3.1.1 Scenario 1 *Manual Entry Analysis*

For this scenario the cycle time is 60 seconds. The precedence diagram has been completed; Task Times for each Task are listed in the upper right in Figure 5. Based upon the precedence, specific operational requirements, space, and individual task times, the analyst has grouped the nine tasks into four new work stations S1 through S4. Note that the grouped process times for each work station do not exceed the cycle time (Krajewski and Ritzman, 2009).
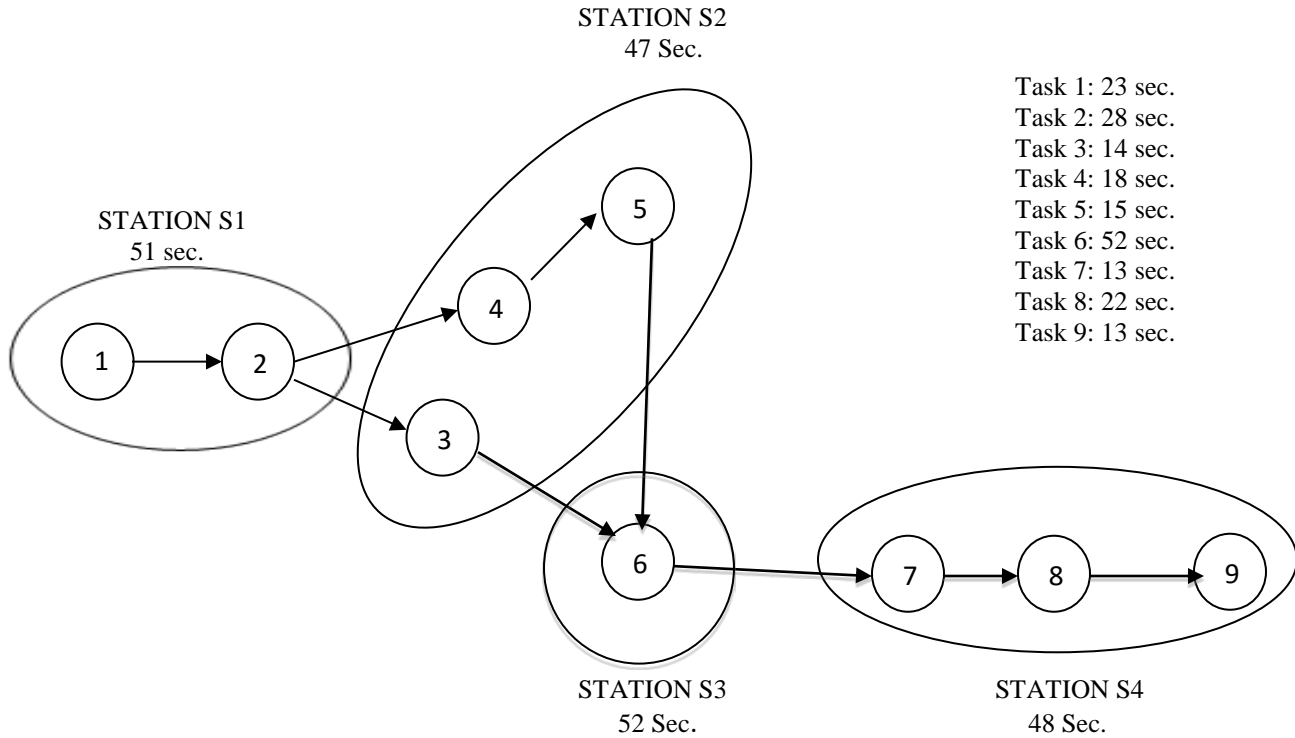

Figure 5. Grouped Work Stations with Embedded Precedence Diagram

Table 4. Manual Calculations

Station	Work Element Assigned	Time (seconds)	Station Slack
S1	1,2	51	9
S2	3,4,5	47	13
S3	6	52	8
S4	7,8,9	48	12
	Total	**198 (t)**	**42**
# Workstations	**Efficiency**	**82.5%**	
9 (n)	**Inefficiency**	**17.5%**	

Results for the process by individual work stations were calculated as follows;

a. Time in seconds (t) = 198 seconds.
b. Number of individual work stations (n) = 9
c. Cycle time (c) = 60 seconds.

Efficiency = ((t)/(n*c))*100 = ((198/9*60))*100 = 82.50%
Inefficiency = 100% – Efficiency = 100% – 82.50% = 17.50%

3.1.2 Spreadsheet Solution (Table5)

Table 5. Spreadsheet Results for Scenario 1

Station	Task Time (sec)	# of O/M	Assigned	Idle Time	% Efficiency	% Idle
S1	51.000	0.850	1	9.0000	85.000	15.000
S2	47.000	0.783	1	13.0000	78.333	21.667
S3	52.000	0.867	1	8.0000	86.667	13.333
S4	48.000	0.800	1	12.0000	80.000	20.000
Efficiency	**82.500**				**Inefficiency**	**17.500**

3.2. Scenario 2 (Process Time > Cycle Time)

To illustrate this scenario, the process is illustrated in Figure 6 and the times for each station are listed in in Table 6. Cycle Time remains at 1 Minute. Demand 405 unit per day will give a Cycle time of 1.00 minutes.
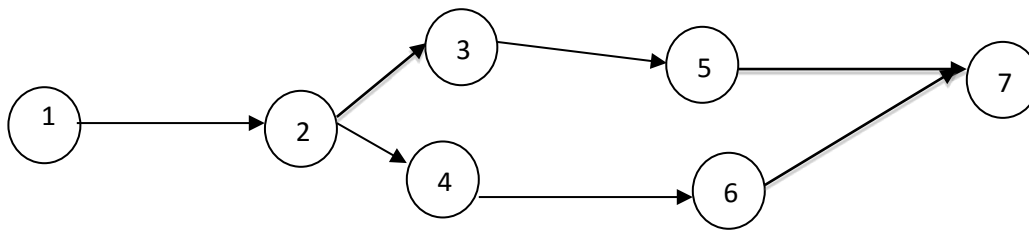

Figure 6. Precedence Diagram

Table 6. Spreadsheet Results for Scenario 2

	A	B	C	D	E	F	G
10	*Cycle Time*						
11	**Minutes =**	1.000	**Min/Unit**				
14	**Work**	**Task**	**# of Machines**		**Idle Time**	**Percent (%)**	**Percent (%)**
15	**Station**	**Minutes**	**Operators**	**Assigned**	**Minutes**	**Efficiency**	**Inefficiency**
16	1	1.200	1.20	2	0.800	60.000	40.000
17	2	2.250	2.25	3	0.750	75.000	25.000
18	3	3.000	3.00	3	0.000	100.00	0.000
19	4	0.750	0.75	1	0.250	75.000	25.000
20	5	1.980	1.98	2	0.020	99.000	1.000
21	6	3.250	3.25	4	0.750	81.250	18.750
22	7	2.500	2.50	3	0.500	83.333	16.667
24		14.930	14.93	18			
26	*% Efficiency*	82.944		**% Inefficiency**	17.056		

As seen in the above spreadsheet entry, the cycle time in minutes for this process is 1.000 minutes /unit. Looking at the Task Time (minutes) column of Table 6, the process time of activity 4 is less than the cycle time. All other times are greater than the cycle time. For this reason, in the *Assigned* column of Table 6, the spreadsheet automatically additional assigned operators/machines to each workstation where the process time exceeded the cycle time.

3.3. Bottlenecks

For more complicated processes one may be faced with problem where its layout dictates that a process time may exceed the cycle time. Steps can then be taken to eliminate potential bottlenecks. Remember, your workers may be the best consultant on how to eliminate this problem (Krajewski and Ritzman, 2009 and Konz, 1985). It is important to note the significance of choke points within a process layout. Looking at Figure 5, notice Stations S1, S2, and S3. Located in Station 3 is work element 6, with two work elements preceding (work elements 3 and 5). These choke points are the points within a process layout where the layout is most sensitive. Essentially speaking, there are two work elements within a process layout sending a part to one proceeding work element, which can cause problems in the long run. These points have the potential to slow down the entire system or force it to a complete halt if not handled correctly. It is incredibly important to have a process time significantly lower than the cycle time within choke points. Konz (1985) and Konz and Johnson (2007) offered solutions of buffers and work in progress (WIP) positioned in strategic areas within the process. Located at strategic locations, these buffers and WIP stations can greatly reduce the effects of bottle necks, shock and disturbance, and starvation.

4. Discussion

Calculating cycle time can be a long, arduous process. However, as this pedagogy illustrates, EXCEL is more than capable of completing these complex calculations, with a small amount of help from the user. As seen in Section 3.1 (Scenario 1 (Process Time < Cycle Time), calculating the efficiencies and inefficiencies (balance delay) of an assembly/production line can be calculated manually. When the results of the manual entry were compared against the results of the spreadsheet model developed in this paper, the results were essentially identical. Both models depicted an 82.5% efficiency rating, while assuming a balance delay of 17.5%.

When process time exceeds the cycle time (3.2. Scenario 2 (Process Time > Cycle Time), modifications must be made to the model. As detailed in Section 3.2, one of four things must be done; either add another shift, change the cycle time, procure more efficient machines, develop more optimal processes, or increase the number of operators/machines in each workstation. One major advantage of increasing the number of shifts is that no more machines are needed, thereby saving money. However, increasing the number of shifts will also inevitably increase labor costs. On the other hand, increasing the

number of operators/machines within a workstation will require the allocation of funds to purchase additional machines. Doing this will eliminate the need for additional shifts, which will keep labor costs low. Each option has its own advantages and disadvantages. It is up to engineer and management to decide which decision will be of the most beneficial. Often times, the individual environment and circumstance of the business itself assists in determining which decision will be selected.

5. Conclusion

Line balancing is a tool used by operations managers, industrial engineers, and instructors across the world. With user-friendly, affordable line balancing software becoming obsolete for current operating systems, demand exists for a reliable alternative. Microsoft EXCEL proves to be fully capable of these calculations. The major advantage to the practitioner, engineer, instructor, and student is that EXCEL is readily available on all personal computers, easily understood, and is very practical. Students with very little exposure to line balancing were able to master the method within the first hour of exposure.

Once the EXCEL spreadsheet and accompanying worksheets were completed, the results were compared to several different heuristic algorithms. After the authors were satisfied that the results were accurate and not significantly different from other examined algorithms, the final step was to develop a working pedagogy to efficiently describe the process. The results of this project were satisfactorily tested in a production operations class and a local manufacturing firm that conducts a line balance on each of its many processes every three months.

6. Acknowledgement

7. References

Emmons, Flowers, Khot, and Mathur. STORM 4.0 Quantitative Modeling for Decision Support for Windows. Lakeshore Communications Inc. Cleveland, 2001.

Konz, S. (1985). *Facility Design.* Hoboken, NJ: John Wiley & Sons Inc.

Konz, S., and Johnson, S. (2007). *Work Design: Occupational Ergonomics (7th Edition).* Scottsdale, AZ: Holcomb Hathway, Publishers.

Krajewski, L., Ritzman, L., and Malhotra, M. (2009). *Operations Management (9th Edition).* Upper Saddle River, NJ: Prentice Hall.

Lovejoy-Henkel, T.J., Lee, J., Parsons, D., and Yearout, R. (2014). *Systematic Pedagogy to Line Balancing with EXCEL.* Proceedings of the 3rd Annual World Conference of the Society for Industrial and Systems Engineering. San Antonio Texas, TX.

Microsoft. Microsoft Excel. Redmond, Washington: Microsoft, 2013. Computer Software.