



INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Interactive Excel-Based Procedure for Line Balancing

John F. Wellington, Stephen A. Lewis

To cite this article:

John F. Wellington, Stephen A. Lewis (2018) Interactive Excel-Based Procedure for Line Balancing. INFORMS Transactions on Education (): <https://doi.org/10.1287/ited.2017.0192>

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. You are free to download this work and share with others for any purpose, even commercially if you distribute your contributions under the same license as the original, and you must attribute this work as “*Informations Transactions on Education*. Copyright © 2018 The Author(s). <https://doi.org/10.1287/ited.2017.0192>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-sa/4.0/>.”

Copyright © 2018, The Author(s)

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Interactive Excel-Based Procedure for Line Balancing

John F. Wellington,^a Stephen A. Lewis^b

^a Indiana University—Purdue University Fort Wayne, Fort Wayne, Indiana 46805; ^b Mongrel Works, LLC, Columbus, Ohio 43209

Contact: wellingj@ipfw.edu (JFW); lewis.sa@mongrelworks.org (SAL)

Received: April 7, 2017

Revised: July 31, 2017; October 18, 2017; November 9, 2017

Accepted: November 27, 2017

Published Online in Articles in Advance: May 2, 2018

<https://doi.org/10.1287/ited.2017.0192>

Copyright: © 2018 The Author(s)

Abstract. The assignment of tasks to workstations that results in the assembly of an item or provision of a service is a common topic of discussion in operations management (OM) courses. In this paper, we present an interactive Excel spreadsheet assisted procedure for assigning the required tasks. The procedure actively engages the student in the process and serves as a tool for learning how to effectively and efficiently assign the tasks to a series (line) of connected workstations. The Excel implementation provides more flexibility in selecting tasks than textbook or other Excel based methods that rely on certain priority rules, and is considerably easier to use than math programming models. The procedure and its spreadsheet implementation are illustrated and discussed, with an example.

 **Open Access Statement:** This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. You are free to download this work and share with others for any purpose, even commercially if you distribute your contributions under the same license as the original, and you must attribute this work as “*Informs Transactions on Education*. Copyright © 2018 The Author(s). <https://doi.org/10.1287/ited.2017.0192>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-sa/4.0/>.”

Keywords: assembly line balancing • interactive mode • excel-based line balancing methodology

1. Introduction

The assembly line balancing (ALB) problem is a common topic of discussion in operations management (OM) courses. It is presented as an investigation of how a series of tasks required for the assembly of an item or the provision of a service can be assigned to the fewest workstations in a balanced timely manner that is consistent with product or service demand. In the discussion, OM students are introduced to the activity or completion times of the necessary tasks and their sequencing requirements referred to as precedence relations in this paper. In orientation to the ALB problem, students learn the importance of assigning the tasks to a series of connected workstations so that the sum of the activity times for the tasks assigned from workstation to workstation is as uniform (balanced) as possible. Student attention is drawn to how, through the balancing process, the use of workstation resources is improved and bottlenecks associated with widely varied task completion times become less disruptive. In addition, it is generally pointed out to students that the assignment of the tasks to workstations is in practice often driven by a targeted cycle time or completion rate (e.g., finished units per hour). In summary, the line balancing problem is typically framed as the search for an assignment of tasks to workstations so that: “(1) each workstation can complete its assigned set of tasks within the desired cycle time; (2) the precedence constraints among the tasks are satisfied; and (3) the number of workstations is minimized” Ragsdale and Brown (2004, p. 45).

Contributions to the line balancing literature include math programming formulations of the problem, heuristic solution strategies, and many other treatments of the ALB problem. Treatments vary significantly in the depth of background material required for comprehension and ease of implementation and in turn their appeal for teaching and learning the ALB process. In this regard, Ragsdale and Brown (2004, p. 45) note that “integer programming formulations are possible, [however] they quickly become unwieldy and increasingly difficult to solve as problem size increases. As a result, many authors recommend heuristic approaches to solving the line balancing problem (Meredith and Schafer (2003), Sabuncuoglu et al. (2000), Suresh et al. (1996)).” The proposed procedure with its spreadsheet implementation does not require a mathematical model of the ALB situation of interest. Much like heuristics, the procedure does not guarantee an optimal solution but allows the student, through interaction, to explore the ALB solution space with the expectation of finding at least a satisfying assignment of tasks to workstations. For assistance, the student can apply rules commonly found in textbooks for ordering eligible tasks for assignment consideration. In applying those rules, the OM student has some confidence that this is a good way to proceed in balancing a line.

This paper presents a spreadsheet tool for learning how to balance a simple assembly line. The tool is interactive, easy to use, adaptable to problems of varying sizes, not subject to the problems noted by Ragsdale

and Brown (2004) and others, and does not have the overhead of formulating and solving a math programming problem using special purpose solution software.

2. The Framework of the Spreadsheet Tool

The Excel spreadsheet implementation of the proposed line balancing procedure consists of four worksheets, see Figure 1. In worksheet Data, the student inputs the task references using identifiers (letters, numbers or combinations of both) of choice, their activity times and precedence requirements, and the desired cycle time. In worksheet Evaluator, the student selects, evaluates, and assigns tasks to workstations one at a time, one per row. This is similar to the ALB methods provided in most OM textbooks. In each row of this worksheet, the workstation designation, the assigned tasks, and the unassigned tasks at that point in the configuration are displayed. Also, the task-to-workstation time fit and precedence feasibilities of all tasks are made available to help the student identify the next task to assign. This includes identification of the feasible task with the largest completion time, which is a popular textbook rule for prioritizing eligible tasks for assignment consideration. The cumulative record of one-task-at-a-time assignments is maintained in the rows of this worksheet. The Calculator worksheet is where the task-to-workstation time fit and precedence eligibilities of the unassigned tasks are determined. The Results worksheet records the assignment of tasks to workstations and provides measures of the efficiency of the configured line. After inputting the data for a situation of interest in the Data worksheet, the student operates in the Evaluator worksheet.

Moving from row to row of the Evaluator worksheet making one assignment per row requires the student to input the workstation identifier in each row and select the task to assign to the current workstation, i.e., only two cell entries per row. Within each row, the student can investigate the merit of any eligible unassigned task. Students can also examine the feasibility of multiple task assignments to a workstation using cell Evaluator!H3.

The use of electronic spreadsheets for solving the ALB problem is not new. Our approach departs from previous spreadsheet contributions such as Ragsdale and Brown (2004) and Weiss (2013) in that it does not require Excel's Solver or Scenario Manager or a mathematical formulation of the ALB problem. The proposed method is interactive and as such opens the solution space to allow the student to order eligible tasks

for assignment consideration using priority rules discussed by Weiss (2013, p. 115). One rule has not been demonstrated to be universally superior. The same is true for the use of one rule within a given problem. Interaction also allows the student to examine task-to-workstation assignments that arise from visual inspection and through trial and error. In these ways, the student interacts with and governs the assignment process.

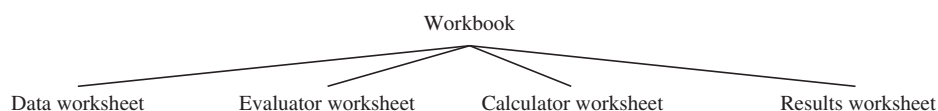
Regarding the use of an electronic spreadsheet for teaching line balancing, consider the following remark of Weiss (2013, p. 114). "Because I teach an operations course, I do not ask the students to build the model but rather use it as a template for solving problems with the same number of tasks as the template. I do point out to the students the inclusion of Excel's array formulas in the model and the use of Excel's Scenario Manager." Like Weiss, the authors teach operations courses, use an Excel spreadsheet to demonstrate line balancing, and do not require students to build mathematical models of ALB situations. However, the Excel spreadsheet described in this paper accommodates as many as fifty tasks without altering any cell formula.

Two other considerations governed the framing of the procedure and the worksheets. One is related to the Genikomsakis and Tourassis (2008, p. 1626) observation that the ALB problem is a "NP-hard combinatorial problem and for large-scale lines the objective is often reduced to finding feasible solutions." The other consideration is due to Wang et al. (2011, p. 4317), who state that "...in practice decision makers want a method that examines alternatives and provides them with a subset of alternatives that are close to the best, so that they can choose the final decision from the 'best-subset,' instead of unconditionally trusting the best solution provided by a computer." The spreadsheet-assisted procedure described in this paper leads the student to only feasible assignments and provides the means for discovering alternative assignments in a student-controlled interactive environment. Discovery of alternative solutions prompts discussions of their relative merits that enrich the learning experience. Through the illustrations in Section 5, we show how the worksheets facilitate this. Hereafter, the terms workbook and spreadsheet should be considered equivalent and understood to collectively refer to worksheets Data, Evaluator, Calculator, and Results.

3. Relevant Literature

How to frame and solve the ALB problem is the subject of many articles in the literature. The discussions include modeling formalizations, solutions strategies,

Figure 1. The Spreadsheet Workbook



heuristics, and other treatments of the problem. The contributions span a wide breadth of methodologies. Some are challenging to present and implement for instructional purposes. Reviews of line balancing methodologies are provided in Erel and Sarin (1998), Driscoll and Thilakawardana (2001), Scholl and Becker (2006), and Boysen et al. (2007, 2008). Becker and Scholl (2006) is a frequently cited reference for this topic. Battaia and Dolgui (2013, p. 272) provide an excellent taxonomy of the literature's treatment of the ALB problem. Fish (2005) presented a novel way to teach line balancing using Duplo® blocks.

Pedagogical software for solving the ALB problem appears in Chang (2003), Weiss (2006), and Ragsdale (2018). The OM student may be interested to know that commercial ALB software used by companies includes Proplanner Line Balancing (2017) and Timer Pro Professional Line Balancing Module (2017) as well as the IBM COMSOAL (Computer Method for Sequencing Operations for Assembly Lines) and the GE ASYBL (Assembly Line Configuration Program).

As noted earlier, Excel spreadsheet based methods for treating the ALB problem are discussed in the literature. Ragsdale and Brown (2004) used Excel's array formula feature and its evolutionary search engine Premium Solver for Education. More recently, Weiss (2013) presented an Excel framework that addressed some of the perceived shortcomings of the Ragsdale and Brown (2004) spreadsheet method. According to Weiss (2013, p. 114), his spreadsheet "solves the problem as soon as the data is entered as opposed to using Excel's evolutionary search engine as proposed by Ragsdale and Brown (2004)." Weiss (2013, p. 114) states that use of the evolutionary search engine in Solver Excel 2010 "was very time consuming because it required thousands of subproblems to be solved for Ragsdale and Brown's (2004) small example." The Weiss (2013) spreadsheet uses Excel's Scenario Manager and array formula features. Mason (2013, pp. 45–46) addressed the limitations of using Excel's Solver add-in for optimization problems.

The Excel template presented in this paper and its use in interactive mode is similar to the Nurre and Weir (2017) interactive Excel-based Gantt Chart Schedule builder that encourages the student to experiment with solution building and discovery.

4. The Step-by-Step Interactive Procedure

Next, we describe the procedure for interacting with the assignment of tasks to workstations using the four worksheets of the workbook titled *template.xlsx* (available at <http://www.mongrelworks.org/ALB2017>). In the procedure, r is the current row of the Evaluator worksheet, s is the evolving number of workstations,

and t is the number of tasks to be assigned. User input is limited to the blue filled cells of worksheets Data and Evaluator. Winston (2014) is a good reference for understanding the cell formulas of the worksheets.

4.1. Populating Worksheet Data

In the Data worksheet, task identifiers, completion times, and required predecessor tasks are entered in the blue filled cells of columns B, C, and D where information for task one is entered in row 2, for task two in row 3, etc., concluding with entry of the information for the last task in row $t + 1$. The headings in cells B1:D1 indicate where to enter this information. Task identifiers may be letters (a, b, ...), numbers (1, 2, ...) or combinations of both (an11, za2n, ...) without intervening spaces. The number of characters for any task identifier should not exceed four. The targeted cycle time should be entered in cell F2. Cells Am:Fm and below should be cleared of entries before proceeding, $m = t + 2$. Observe the note in cell H2 of this worksheet.

4.2. Using Worksheet Evaluator

In this worksheet, workstation identifiers must be the consecutive unit-spaced positive integers beginning with 1. The assignments are made one task per row in consecutive rows beginning with row $r = 11$. The student interacts with the task-to-workstation assignments using the provided template as follows:

Step 0: (Initialize). Open *template.xlsx* and enter data in the blue filled cells of worksheet Data. Next, move to worksheet Evaluator where the blue-filled cells C11–C60 and F11–F60 should be blank. Steps 1–3 are performed in worksheet Evaluator. Set $r = 10$ and $s = 1$. Go to Step 1.

Step 1: (Row identification). Increment r by 1. Enter the current workstation identifier s in cell Cr. Go to Step 2.

Step 2: (Workstation identification). In row r , select any task identifier from cell Er with its preceding comma and enter them in cell Fr. If no task identifier is available for selection from cell Er, stop. All tasks have been allocated and tasks-to-workstations assignments are available in worksheet Results. Otherwise, if the display of cells A11:B11 and below show that no tasks are feasible for assignment and cell G1 displays #N/A, increment s by 1, enter the result in cell Cr. Go to Step 3.

Step 3: (Task identification). Consult the displays of cells A11:B11 and below for the currently available time and precedence feasible unassigned tasks as well as cells E1:H1 for the feasible task with the largest completion time. Among them, identify the preferred task to assign and enter its identifier with a preceding comma in cell Fr by writing over entries there or appending the identifier and its preceding comma as the rightmost cell entries. Go to Step 1.

The first action of Step 2, entry of any valid task identifier in cell Fr, prompts display of the feasibility status

of all tasks in cells A11:B11 and below. Note that the entered identifier may not be the final choice for assignment. The latter is determined in Step 3. Step 2 sets up Step 3 so that only feasible tasks are considered for assignment. In Step 3, the student may select the identifier for any task associated with “yes” in cells A11:B11 and below. To revisit the assignment of tasks to workstations, the history of the feasible tasks available at the time of assignment in any row r may be recorded in cell Fr. Appending the identifiers of the then feasible tasks with their preceding commas in cell Fr records the history for later reference. This is helpful in searching for alternative workstation assignments. In any row r , the worksheet considers the task indicated by the rightmost identifier in cell Fr to be the assigned task of choice.

Since the feasibility status of all tasks is available with each assignment consideration (row r), a student can apply rules for prioritizing feasible unassigned tasks, in addition to the task with the largest completion time displayed in cells E1:H1. An example is the task with the largest number of follower tasks. Visual inspection may also reveal a task for consideration. Students are sometimes adept at identifying a subset of unassigned tasks that collectively fit the current workstation. Cell H3 allows the student to examine the feasibility of multiple task assignments.

4.3. Worksheets Calculator and Results

Execution of the step-by-step procedure conducted in worksheet Evaluator does not call for student

interaction with worksheet Calculator. The outcome of the step-by-step procedure is displayed in worksheet Results with simple efficiency measures associated with the final assignments.

5. An Illustration

The proposed spreadsheet procedure for balancing a simple line is illustrated with an example from EDRAW (2016) with $t = 20$ tasks.

5.1. The First Solution

Figure 2 is the image of the task identifiers, times, and precedence requirements that were inputted to worksheet Data for this example according to the discussion in Section 4.1. The desired cycle time of 100 was entered in cell F2. The workstation identifiers were the unit spaced positive integers beginning with 1. Since the sum of task times in column C of Figure 2 is 710 and the desired cycle time is 100, at least eight workstations were expected.

In the following discussion, unless otherwise noted, the cell references relate to worksheet Evaluator. The following was observed in implementing the procedure described in Section 4.2. With each new row r , the entry of the workstation identifier s in cell Cr was immediately followed with entry of the leftmost task identifier of cell Er (locus of unassigned task identifiers) and its preceding comma in the rightmost position of cell Fr (locus of current task under assignment consideration). In each case, this prompted display of

Figure 2. Image of the Data Worksheet of Workbook illustration.xlsx

	B	C	D	E	F
1	Task	Task time	Predecessor task(s)	No. of predecessor tasks	Desired cycle time
2	a	60.00		0	100
3	b	70.00	a	1	
4	c	10.00	b	1	
5	d	20.00		0	
6	e	10.00	d	1	
7	f	20.00	k	1	
8	g	60.00		0	
9	h	60.00		0	
10	i	15.00		0	
11	j	30.00	g, h, i	3	
12	k	15.00	j	1	
13	l	30.00	k	1	
14	m	60.00	f, l	2	
15	n	30.00		0	
16	o	30.00	n	1	
17	p	50.00	o	1	
18	q	40.00	c	1	
19	r	30.00	e, k	2	
20	s	20.00	r	1	
21	t	50.00	m, p, q, s	4	

information in cells A11:B30 and E1:H1, which assisted in the execution of Steps 2 and 3. In each row r , the task selected for assignment was the feasible task with the largest completion time identified as such in cells E1:H1. Figures 3–6 show the results of assigning the first three tasks to workstations. Cells A11:B30 changed with each row $r = 11, \dots, 30$. In each row, the required entries were the workstation designation in cell Cr and in cell Fr the identifier(s) of the task(s) selected for evaluation and among them the assigned task of choice appearing in the rightmost position of cell Fr.

The procedure began with the population of row $r = 11$ and entry of the workstation identifier 1 in cell C11 with Step 1 and was followed by entry of (, a) in cell F11 with Step 2, see Figure 3. This resulted in the task feasibility display of cells A11:B30, which showed that tasks a, d, g, h, i, n were time and precedence feasible for assignment to workstation 1. Cells E1:H1 showed that among them tasks a, g, and h had the largest completion time (=60). Based on this selection criterion, task a was chosen for assignment. Because (, a) was previously entered in cell F11 with Step 2, Step 3 concluded. Cells F2:F7 display the propriety of the assignment. The first task-to-workstation assignment was completed.

The procedure continued with the population of the next row $r = 12$ where workstation identifier 1 was entered in cell C12 with Step 1 and (, b) in cell F12 with Step 2; see Figure 4. As a result, cells A11:B30 showed that tasks d, i, n were the only time and precedence feasible tasks eligible for assignment with task n having the largest task completion time; see cells E1:H1 in Figure 4. Consequently, (, n) was appended rightmost in cell F12, thus concluding Step 3. Cells F2:F7 display the propriety of the assignment. Note that cell F7 indicated 10 units of remaining time for any subsequent task assignment to workstation 1. The second task-to-workstation assignment was completed.

The population of row $r = 13$ was next with the third assignment. When station identifier 1 was entered in cell C13 with Step 1 and followed by (, b) in cell F13 with Step 2, cells A11:B30 showed that there were no feasible tasks available for assignment; see Figure 5. Note also that cell G1 displayed #N/A. This is attributed to insufficient remaining workstation 1 time (=10, cell F7 of Figure 4) to accommodate any additional task. According to Step 2 of the step-by-step procedure, the workstation identifier s was incremented by 1 and the result (=2) entered in cell C13. Thereafter, cells A11:B30 displayed tasks b, d, g, h, i, o as the time and precedence feasible tasks; see Figure 6. Among them, task b had the largest task completion time; see cells E1:H1 in Figure 6. Task b was the choice for assignment. Cells F2:F7 display the propriety of the assignment. Because (, b) was previously entered in cell F13 with Step 2, Step 3 concluded. The third task-to-workstation assignment was completed.

In row $r = 13$ as possible actions of Step 3, any of the task identifiers b, d, g, h, i, o could have been appended one-at-a-time with the preceding comma as the rightmost characters of cell F13 in Figure 6. In each case, cells F2:F7 would display the consequence of the assignment including remaining workstation time. Appending in this manner may occur in any row r and facilitates inspection of alternatives to assigning the task with the largest completion time identified in cells E1:H1. Also, as noted in Section 4.2, the history of feasible tasks available for assignment in any row r may be appended to cell Fr for latter reference when searching for alternative assignments from row r forward. If the cell contents of F13 in Figure 6 read b, d, g, h, i, o, b, they would indicate that task b was first considered (outcome of Step 2), followed by Step 3 entries and inspections for tasks d, g, h, i, o. The examinations concluded with task b as the task of choice for the third assignment. In each case, cells F2:F7 would indicate the propriety of the prospective assignment.

The assignments for all twenty tasks are shown in Figure 7. The assignments, taken from worksheet Results, are summarized as Solution 1 in Figure 8. Note that in Figures 3–7 all entries made to cell Fr are reported. This was done to record the complete actions of Steps 2 and 3. In each row r , the identifier for the task selected for final assignment and its preceding comma could have overwritten any previous entries in cell Fr that resulted from Steps 2 and 3.

5.2. Analysis and Discussion

Given the desired cycle time (=100) and the sum of the 20 tasks times (=710), fewer than eight workstations were not possible. Furthermore, the idle time totality of 90 for Solution 1 displayed in Figure 8 appears to be a zero-sum phenomenon, i.e., moving tasks around the workstations to balance idle times has a net effect of zero. If this is true, the best that could be discovered through further investigation would be a solution with less than the largest workstation idle time of 50 found so far; see workstation 8 of Solution 1 in Figure 8. One way to proceed in the search for such a configuration is to first select one task as the only task for assignment to any one of the eight workstations. Then use the step-by-step procedure to verify that no more than eight workstations are populated, that total idle time for the eight workstations does not exceed 90, and that the largest idle time among the configured workstations is no greater than 50. Candidates to consider for the one-task-to-workstation assignment are those with the greatest completion times. Tasks a, g, h, m require 60 time units each; task b calls for 70 time units. These tasks as sole workstation assignments could result in maximum idle workstation time of 30 (cycle time 100-70 with task b) or 40 (cycle time 100-60 with task a, g, h, or m). In this way, a target or bound

Figure 3. Partial Image of the Evaluator Worksheet for the First Task Assignment in Row $r = 11$

	A	B	C	D	E	F	G	H
1	Evaluator sheet:				Max eligible task time →	60.00	a	← Task
2					Workstation →	l		
3			Desired cycle time		Current task candidate →	a		
4			100.00		Candidate predecessor(s) assigned?	Yes		
5			a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t		Candidate task time fit?	Yes		
6					Task candidate previously assigned?	No		
7					Remaining station time with task candidate assigned →	40.00		
10	Task	Eligible for assignment?	Workstation	Tasks previously assigned	Tasks unassigned	Task(s) considered for assignment	Task to be assigned	Task's time
11	a	Yes	l		, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	, a	a	60
12	b	No						
13	c	No						
14	d	Yes						
15	e	No						
16	f	No						
17	g	Yes						
18	h	Yes						
19	i	Yes						
20	j	No						
21	k	No						
22	l	No						
23	m	No						
24	n	Yes						
25	o	No						
26	p	No						
27	q	No						
28	r	No						
29	s	No						
30	t	No						
31								

Figure 4. Partial Image of the Evaluator Worksheet for the Second Task Assignment in Row $r = 12$

	A	B	C	D	E	F	G	H
1	Evaluator sheet:				Max eligible task time →	30.00	n	← Task
2					Workstation →	l		
3			Desired cycle time		Candidate task candidate →	n		
4			100.00		Candidate predecessor(s) assigned?	Yes		
5			a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t		Candidate task time fit?	Yes		
6					Task candidate previously assigned?	No		
7					Remaining station time with task candidate assigned →	10.00		
10	Task	Eligible for assignment?	Workstation	Tasks previously assigned	Tasks unassigned	Task(s) considered for assignment	Task to be assigned	Task's time
11	a	No	l		, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	, a	a	60
12	b	No	l	, a,	, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	b, n	n	30
13	c	No						
14	d	Yes						
15	e	No						
16	f	No						
17	g	No						
18	h	No						
19	i	Yes						
20	j	No						
21	k	No						
22	l	No						
23	m	No						
24	n	Yes						
25	o	No						
26	p	No						
27	q	No						
28	r	No						
29	s	No						
30	t	No						
31								

Figure 5. Partial Image of the Evaluator Worksheet for the Initial Third Task Assignment in Row $r = 13$

	A	B	C	D	E	F	G	H
1	Evaluator sheet:				Max eligible task time →	0	#N/A	← Task
2					Workstation →	1		
3			Desired cycle time		Candidate task candidate →	b		
4			100.00		Candidate predecessor(s) assigned?	Yes		
5			a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t		Candidate task time fit?	No		
6					Task candidate previously assigned?	No		
7					Remaining station time with task candidate assigned →	–60.00		
10	Task	Eligible for assignment?	Workstation	Tasks previously assigned	Tasks unassigned	Task(s) considered for assignment	Task to be assigned	Task's time
11	a	No	1		, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	, a	a	60
12	b	No	1	, a,	, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	, b, n	n	30
13	c	No	1	, a, n,	, b, c, d, e, f, g, h, i, j, k, l, m, o, p, q, r, s, t,	, b	b	70
14	d	No						
15	e	No						
16	f	No						
17	g	No						
18	h	No						
19	i	No						
20	j	No						
21	k	No						
22	l	No						
23	m	No						
24	n	No						
25	o	No						
26	p	No						
27	q	No						
28	r	No						
29	s	No						
30	t	No						
31								

Figure 6. Partial Image of the Evaluator Worksheet for the Final Third Task Assignment in Row $r = 13$

	A	B	C	D	E	F	G	H
1	Evaluator sheet:				Max eligible task time →	70.00	b	← Task
2					Workstation →	2		
3			Desired cycle time		Candidate task candidate →	b		
4			100.00		Candidate predecessor(s) assigned?	Yes		
5			a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t		Candidate task time fit?	Yes		
6					Task candidate previously assigned?	No		
7					Remaining station time with task candidate assigned →	30.00		
10	Task	Eligible for assignment?	Workstation	Tasks previously assigned	Tasks unassigned	Task(s) considered for assignment	Task to be assigned	Task's time
11	a	No	1		, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	, a	a	60
12	b	Yes	1	, a,	, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	b, n	n	30
13	c	No	2	, a, n,	, b, c, d, e, f, g, h, i, j, k, l, m, o, p, q, r, s, t,	, b	b	70
14	d	Yes						
15	e	No						
16	f	No						
17	g	Yes						
18	h	Yes						
19	i	Yes						
20	j	No						
21	k	No						
22	l	No						
23	m	No						
24	n	No						
25	o	Yes						
26	p	No						
27	q	No						
28	r	No						
29	s	No						
30	t	No						
31								

Figure 7. Partial Image of the Evaluator Sheet for the Complete History of Task Assignments¹

	A	B	C	D	E	F	G	H
1	Evaluator sheet:				Max eligible task time →	0	#N/A	← Task
2					Workstation →	8		
3			Desired cycle time		Candidate task candidate →	t		
4			100.00		Candidate predecessor(s) assigned?	Yes		
5			a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t		Candidate task time fit?	Yes		
6					Task candidate previously assigned?	Yes		
7					Remaining station time with task candidate assigned →	0		
10	Task	Eligible for assignment?	Workstation	Tasks previously assigned	Tasks unassigned	Task(s) considered for assignment	Task to be assigned	Task's time
11	a	No	1		, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	, a	a	60
12	b	No	1	, a,	, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t,	, b, n	n	30
13	c	No	2	, a, n,	, b, c, d, e, f, g, h, i, j, k, l, m, o, p, q, r, s, t,	, b	b	70
14	d	No	2	, a, n, b,	, c, d, e, f, g, h, i, j, k, l, m, o, p, q, r, s, t,	, c, o	o	30
15	e	No	3	, a, n, b, o,	, c, d, e, f, g, h, i, j, k, l, m, p, q, r, s, t,	, c, g	g	60
16	f	No	3	, a, n, b, o, g	, c, d, e, f, h, i, j, k, l, m, p, q, r, s, t,	, c, d	d	20
17	g	No	3	, a, n, b, o, g, d,	, c, e, f, h, i, j, k, l, m, p, q, r, s, t,	, c, i	i	15
18	h	No	4	, a, n, b, o, g, d, i,	, c, e, f, h, j, k, l, m, p, q, r, s, t,	, c, h	h	60
19	i	No	4	, a, n, b, o, g, d, i, h,	, c, e, f, j, k, l, m, p, q, r, s, t,	, c, j	j	30
20	j	No	4	, a, n, b, o, g, d, i, h, j,	, c, e, f, k, l, m, p, q, r, s, t,	, c	c	10
21	k	No	5	, a, n, b, o, g, d, i, h, j, c,	, e, f, k, l, m, p, q, r, s, t,	, e, p	p	50
22	l	No	5	, a, n, b, o, g, d, i, h, j, c, p,	, e, f, k, l, m, q, r, s, t,	, e, q	q	40
23	m	No	5	, a, n, b, o, g, d, i, h, j, c, p, q,	, e, f, k, l, m, r, s, t,	, e	e	10
24	n	No	6	, a, n, b, o, g, d, i, h, j, c, p, q, e,	, f, k, l, m, r, s, t,	, f, k	k	15
25	o	No	6	, a, n, b, o, g, d, i, h, j, c, p, q, e, k,	, f, l, m, r, s, t,	, f, l	l	30
26	p	No	6	, a, n, b, o, g, d, i, h, j, c, p, q, e, k, l,	, f, m, r, s, t,	, f, r	r	30
27	q	No	6	, a, n, b, o, g, d, i, h, j, c, p, q, e, k, j, r,	, f, m, s, t,	, f	f	20
28	r	No	7	, a, n, b, o, g, d, i, h, j, c, p, q, e, k, l, r, f,	, m, s, t,	, m	m	60
29	s	No	7	, a, n, b, o, g, d, i, h, j, c, p, q, e, k, l, r, f, m,	, s, t,	, s	s	20
30	t	Yes	8	, a, n, b, o, g, d, i, h, j, c, p, q, e, k, l, r, f, m, s,	, t,	, t	t	50
31			8	, a, n, b, o, g, d, i, h, j, c, p, q, e, k, l, r, f, m, s, t,	,			

¹Figure 7 shows worksheet Evaluator at the conclusion of the step-by-step interactive procedure.

Figure 8. Solutions¹ 1–4

	Solution 1		Solution 2		Solution 3		Solution 4	
Workstation	Assigned tasks	Workstation idle time	Assigned tasks	Workstation idle time	Assigned tasks	Workstation idle time	Assigned tasks	Workstation idle time
1	a, n	10	a, n	10	a, n	10	a, n	10
2	b, o	0	b	30	g, o	10	g, o	10
3	g, d, i	5	g, o, c	0	b, d, c	0	b, i	15
4	h, j, c	0	h, q	0	h, q	0	h, j	10
5	p, q, e	0	p, d, i, e,	5	p, i, j	5	p, k, f	15
6	k, l, r, f	5	j, k, l, f	5	k, l, f, e	25	l, m	10
7	m, s	20	m, r	10	m, r	10	c, d, e, r, s	10
8	t	50	s, t	30	s, t	30	q, t	10
Total (max) idle time		90 (50)		90 (30)		90 (30)		90(15)

¹The solution to the integer programming (IP) formulation of the illustration problem in the form given in Chapter 6 Ragsdale (2018) has eight workstations and total workstation idle time of 90. This solution was obtained from the Lingo coded form of the IP problem with 160 0/1 variables and 20 general integer variables with nearly 90 constraints plus Lingo required integer declarative statements.

for the largest workstation idle time can be specified. Of course, task b with the possibility of a workstation idle time of 30 is the most attractive for first consideration. Experimentation with these requirements within the step-by-step procedure produced Solution 2 shown in Figure 8. Solution 2 has task b as the sole assignment for workstation 2 (earliest possible) with a maximum workstation idle time of 30. Unintentionally, the same idle time occurred with workstation 8. The search was successful and the outcome should prompt discussion of which solution (1 or 2) in Figure 8 is more appealing, i.e., perceived as a more balanced line. Except for the assignment of task b to workstation 2, the evolution of Solution 2 proceeded by assigning the feasible task with the largest completion time.

Although the largest idle time for Solution 2 is 30 and fewer than the 50 of Solution 1, there are two workstations (2, 8) with an idle time of 30. Investigation may continue for workstation assignments with more evenly balanced idle times. Discussion of the appeal (or lack thereof) of such solutions enriches student understanding of ALB. In this regard, consider what other workstation configurations may be available. For example, Solution 3 in Figure 8, resulted from maintaining workstation 1 assignments of Solution 2, making task g the first assignment to workstation 2, and thereafter assigning tasks with the largest completion time as noted in cells E1:H1. Solution 3 has workstation 8 as the only workstation with the maximum idle time of 30. At this point, the student may allow trial and error to guide further discovery of possible task assignments with less varied workstation idle times. Several trial and error changes to the assignments of Solution 3 produced Solution 4 in Figure 8. These changes consisted of the following: moving task q to workstation 8 and task s to workstation 7; then moving tasks c, d, and e to workstation 7 and task m to workstation 6; moving tasks k and f to workstation 5, and assigning tasks i and j to workstations 3 and 4, respectively. The reassignments shown

in Solution 4 have a maximum idle time of 15 for workstations 3 and 5. Each of the other six workstations has an idle time of 10. Unlike Solutions 1–3, no workstation has zero idle time. Observe in Solution 4 that cycle time could be reduced to 90 time units with idle times 0, 0, 5, 0, 5, 0, 0, and 0 for workstations 1–8, respectively. Discovery of this solution prompts discussion of the consequence and the desirability of more product resulting from a cycle time of 90 compared to 100, which guided Solutions 1–4. Although Solution 4 appears attractive, instructors should remind students that cycle time is commonly derived from a targeted hourly output based on product demand and other considerations.

For reproducibility of Solutions 1–3 in Figure 8, linking the identifiers noted under the column heading “Assigned tasks” left to right beginning with workstation 1 and proceeding similarly with workstations 2 to 8 is the order in which the tasks were assigned one-at-a-time to the eight workstations in each solution. The Excel workbooks for Solutions 1–3 in Figure 8 are available at <http://www.mongrelworks.org/ALB2017>.

Note the following in interactively stepping through the assignments. Given the specified precedence requirements, task completion times, and targeted cycle time of 100 for the illustration, students can conclude that tasks l, m, r, and s cannot be assigned before workstation 3, and tasks b, c, f, j, k, p, q before workstation 2. Knowing the earliest (and latest) possible workstation assignments for tasks is helpful in identifying alternative task-to-workstation configurations. It also facilitates understanding of the impediments to an ideal configuration.

For students who wish to prioritize eligible tasks for assignment consideration based on the largest number of follower tasks, information is provided in the appendix.

The above demonstrations show that useful discoveries of what is possible in balancing a line can be made through the interactive mode using the worksheets described in this paper.

6. Teaching Experiences with the Spreadsheet

The authors have used the worksheets in various evolving forms in recent undergraduate and MBA level courses. The worksheets were used to demonstrate the line balancing process to first time ALB learners and to students engaged in ALB project work at both levels. In a recent undergraduate OM course, two student groups successfully altered the worksheets to accommodate their investigations of project related ALB situations with more than fifty tasks. A group of MBA students did the same for a course project in the Fall 2016 semester. Both levels of students used the worksheets to solve ALB problems that appeared on in-class examinations. In these course related situations, students were provided with the Excel workbook template and downloaded it to their Macintosh or Windows computers. To the instructors' knowledge, there were no platform issues.

Most students are not good at developing their own Excel formulas of the kind required to interactively solve a simple ALB problem. They are better at adapting given cell formulas to fit the ALB problem of interest. For example, students have used the information of cells Evaluator!A11:B11 and below to prioritize eligible unassigned tasks for assignment consideration. Some have added efficiency measures for comparing alternative solutions and eye-appealing displays of workstation configurations to worksheet Results. For students interested in altering the workbook for more than fifty tasks, instructors can point out the row boundary (=60) of cell formulas in the Evaluator worksheet, the row (=67) and column (=BC) boundaries in worksheet Calculator, and row (=53) boundary in worksheet Results.

From the authors' experiences, most students were unfamiliar with the unary Excel operator "--". However, when it was explained, they understood its use in the formulas appearing in cell Evaluator!F3, i.e., =OFFSET(G10,ROWS(F11:F60)-SUMPRODUCT(--ISBLANK(F11:F60)),0) and in cell Evaluator!F6, i.e., =IF(ISNUMBER(FIND("--"&F3&--,OFFSET(D10,ROWS(C11:C60)-SUMPRODUCT(--ISBLANK(C11:C60)),0))), "yes","no"). Although students perceived them as complicated cell formulas, as long as they understood how to use the Data, Evaluator, Calculator, and Results worksheets described in Section 4 and illustrated in Section 5, this level of complexity in formula construction did not appear to intimidate them. Students clearly benefitted from a walk-through of spreadsheet operations. (See illustration in Section 5). Of course, some students were frustrated with populating the worksheets: This was brought to the authors' attention to simplify student interactions. The evolving form of the offered worksheets reflects the authors' efforts in this matter. Note also that users of the worksheets were all business students.

Some students were unfamiliar with Excel's SUBSTITUTE(), OFFSET(), FIND(), and SUMPRODUCT() functions and the Excel operator &. Some used Google to explore unfamiliar Excel functions and operators, which seemed to help and which encouraged student experimentation with the worksheets. To the students' credit, the utility of the leading comma in referencing tasks was understood as well as the confusion it avoided, for example, in discerning task identifier 1 from the 1 in identifiers 11, 12, and 21. Students also came to appreciate the workbook organization into the four worksheets, and the clutter it avoided. The authors noted that use of multiple Excel functions within a single Excel cell formula was new to some students.

Recently, the authors experimented with use of the spreadsheet in a classroom setting in which student groups competed in identifying good solutions to an ALB problem much like the example in Section 5. Although there was no jury or prize, students had to publicly advocate their solutions to classmates. This exercise was an adaptation of a suggestion from Snider et al. (2017, p. 43) for enhancing student engagement in ALB learning through an experience in which student-discovered solutions are "publicly peer-reviewed by the rest of the class." In the opinion of the authors, this setting worked well. It engaged students and broadened their understanding of ALB through immersion in the search for more than one solution and through critiquing of solutions advanced by their classmates. In the two instances of this exercise, the group solutions differed. Consequently, there was much learned from the exercise.

7. Limitations and Summary

The dimensions of the worksheets described in this paper are of order t , the number of tasks to be assigned. As such, the dimensions are manageable for ALB situations considered in most OM courses, assignments, and projects. When a large number of tasks is considered, some may find stepping through line balancing in interactive mode using the worksheets to be a limitation. Although the authors have not found this to be the case, the perception should not overshadow the learning value of interactive mode using the spreadsheet environment, especially for first-time ALB learners. Using the worksheets can facilitate student discovery of alternative solutions and features that might, otherwise, remain unknown, e.g., the solutions in Figure 8.

The procedure presented here does not guarantee an optimal solution. However, it has minimal startup overhead. Unlike math modeling mode, it does not require the student to first formulate the ALB situation as an optimization problem and then translate it to a form required by special purpose solution software such as Excel's Solver or Lingo (see footnote to

Figure A.1. Precedence Diagram for Illustration of Section 5

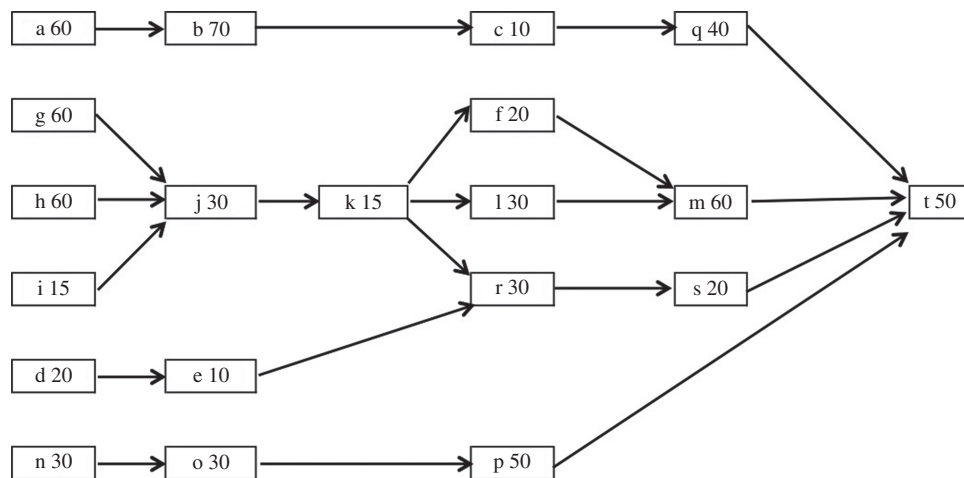


Figure 8). Furthermore, the spreadsheet-assisted procedure presented here is similar to the line balancing treatment in OM textbooks.

In summary, the procedure presented in this paper immerses the student in a visual and hands-on experience with line balancing using the ubiquitous problem-solving environment of Excel. The student sees the solution forming, is in control of its evolution, and is not blind to the merit of other solution possibilities. The line balancing task is in the hands of the student and not the software.

Note that our presentation is not intended to make the case for superiority of the interactive mode using the worksheets, but to make the case for what it can add to student learning of the ALB process.

Acknowledgments

We thank the editor, associate editor, and the anonymous reviewers for suggestions that significantly improved this paper and the Excel worksheets.

Appendix

The precedence diagram for the illustration in Section 5 is provided below as Figure A.1. The number of followers for each task a – t is noted as a(4), b(3), c(2), d(4), e(3), f(2), g(8), h(8), i(8), j(7), k(6), l(2), m(1), n(3), o(2), p(1), q(1), r(2), s(1), t(0).

References

- Battaia O, Dolgui A (2013) A taxonomy of line balancing problems and their solution approaches. *Internat. J. Production Econom.* 142(2):259–277.
- Becker C, Scholl A (2006) A survey on problems and methods in generalized assembly line balancing. *Eur. J. Oper. Res.* 168(3):694–715.
- Boysen N, Fliedner M, Scholl A (2007) A classification of assembly line balancing problems. *Eur. J. Oper. Res.* 183(2): 674–693.
- Boysen N, Fliedner M, Scholl A (2008) Assembly line balancing: Which model to use when? *Internat. J. Production Econom.* 111(2):509–528.
- Chang Y (2003) *Win QSB*, Version 2.0 (John Wiley & Sons, Hoboken, NJ).

- Driscoll J, Thilakawardana D (2001) The definition of assembly line balancing difficulty and evaluation of balance solution quality. *Robotics Comput. Integrated Manufacturing* 17(1–2):81–86.
- EDRAW (2016) Accessed March 18, 2018, <https://www.edrawsoft.com/precedence-diagram.php>.
- Erel E, Sarin S (1998) A survey of assembly line balancing procedures. *Production Planning and Control* 9(5):414–434.
- Fish L (2005) Teaching assembly line balancing: A mini-demonstration with DUPLO® blocks or “the running of the dogs.” *Decision Sci. J. Innovative Ed.* 3(1):169–176.
- Genikomsakis K, Tourassis V (2008) A simulation-based assessment of alternative assembly line configurations. 2008 *IEEE Internat. Conf. Systems, Man and Cybernetics*, 1626–1631.
- Mason A (2013) SolverStudio: A new tool for better optimization and simulation modelling in excel. *INFORMS Trans. Ed.* 14(1):45–52.
- Meredith J, Shafer S (2003) *Introducing Operations Management* (John Wiley & Sons, New York).
- Nurre S, Weir J (2017) Interactive excel-based gantt chart schedule builder. *INFORMS Trans. Ed.* 17(2):49–57.
- Proplanner Line Balancing (2017) Accessed March 18, 2018, <https://www.proplanner.com/solutions/assembly-process-planning/line-balancing/>.
- Ragsdale C (2018) *Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Business Analytics*, 8th ed. (Cengage Learning, Boston, MA).
- Ragsdale C, Brown E (2004) On modeling line balancing problems in spreadsheets. *INFORMS Trans. Ed.* 4(2):45–48.
- Sabuncuoglu I, Erel E, Tanyer M (2000) Assembly line balancing using genetic algorithms. *J. Intelligent Manufacturing* 11(3):295–310.
- Scholl A, Becker C (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur. J. Oper. Res.* 168(3):666–693.
- Snider B, Southin N, Weaver S (2017) Student peer evaluated line balancing competition. *INFORMS Trans. Ed.* 17(2):43–48.
- Suresh G, Vivod S, Sahu S (1996) A genetic algorithm for assembly line balancing. *Production Planning and Control* 7(1):38–46.
- Timer Pro Professional Line Balancing Module (2017) Accessed March 18, 2018, <http://www.acsco.com/LineBalancing.htm>.
- Wang Y, Luangkesorn L, Shuman L (2011) Best subset selection procedure. Jain S, Creasey R, Himmelspace P, White K, Fu M, eds. *Proc. 2011 Winter Simulation Conf.* (IEEE, Piscataway, NJ), 4315–4323.
- Weiss H (2006) *POM-QM for Windows*, Version 3 (Pearson Prentice Hall, Upper Saddle River, NJ).
- Weiss H (2013) Teaching note—Implementing line balancing heuristics in spreadsheets. *INFORMS Trans. Ed.* 13(2):114–125.
- Winston W (2014) *Microsoft Excel 2013: Data Analysis and Business Modeling* (O’Reilly Media, Sebastopol, CA).