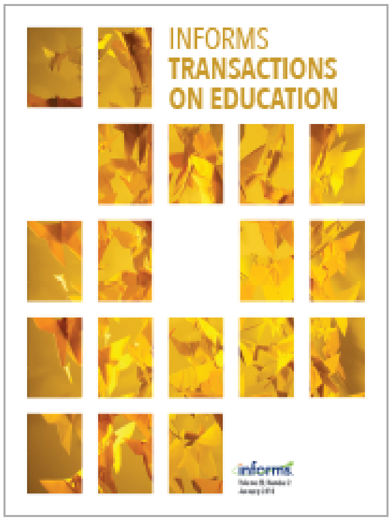


INFORMS Transactions on Education

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

A Tale of Two Linear Programming Formulations for Crashing Project Networks

Collin Huse, Michael J. Brusco

informs
http://pubsonline.informs.org/journal/ited

INFORMS TRANSACTIONS ON EDUCATION
Vol. 21, No. 2, January 2021, pp. 82–95
ISSN 1532-0545 (online)

A Tale of Two Linear Programming Formulations for Crashing Project Networks

Collin Huse,[a] Michael J. Brusco[b]

[a] Nextstage Design, New Haven, Connecticut 06510; [b] Florida State University, Tallahassee, Florida 32306

Contact: collinhuse@gmail.com (CH); mbrusco@fsu.edu, https://orcid.org/0000-0002-1465-6233 (MJB)

Received: August 10, 2019
Revised: June 9, 2020
Accepted: Jun 10, 2020
Published Online in Articles in Advance: October 7, 2020

https://doi.org/10.1287/ited.2019.0236

Abstract. Problems associated with time–cost trade-offs in project networks, which are commonly referred to as crashing problems, date back nearly 60 years. Many prominent management science textbooks provide a traditional linear programming (LP) formulation for a classic project crashing problem, in which the time–cost trade-off for each activity is continuous (and linear) over a range of possible completion times. We have found that, for students who are being introduced to time–cost trade-offs and the principles of project crashing, an alternative LP formulation facilitates a greater conceptual understanding. Moreover, the alternative formulation uses only half of the decision variables in the traditional formulation and has fewer constraints for many problems encountered in management science textbooks. Results from an MBA section of operations management suggest that students prefer the alternative formulation. Additionally, we have developed an Excel workbook that generates all possible paths for a network, allows students to manually evaluate crashing decisions, and generates the alternative LP formulation. We demonstrate the workbook using a small synthetic example and a larger, real-world network from the literature. We also show that the alternative formulation can be adapted easily to accommodate discrete project crashing problems for which the time–cost trade-offs for activities are not necessarily linear.

Supplemental Material: The e-companion is available at https://doi.org/10.1287/ited.2019.0236.

Keywords: project networks • crashing • linear programming • teaching project management

1. Introduction

Time–cost trade-offs in project networks have been studied for nearly six decades (Fulkerson 1961, Kelley 1961). Although there are different versions of time–cost trade-off problems, the general principle is to shorten the completion time of some (or all) of the project activities so as to either (i) complete the project at minimum cost subject to a project completion time constraint or (ii) complete the project in the minimum amount of time subject to a budget constraint. Shortening the completion time of an activity is known as *crashing*. For each activity, there is a *normal* completion time that represents the time for the activity under "normal" conditions as well as a *crash* time that represents the minimum possible completion time for the activity if additional resources are committed. It is commonly assumed that the time–cost trade-off for each activity is a continuous linear function over the interval from crash to normal time; however, discrete (nonlinear) time–cost trade-offs can also be modeled (De et al. 1995).

Many introductory textbooks for management science (Eppen et al. 1993, Hillier and Hillier 2003, Anderson et al. 2019, Taylor 2019), operations research (Winston 1994), and quantitative methods (Balakrishnan et al. 2007, Render et al. 2009) provide the same *traditional* linear programming (LP) formulation for minimizing total crashing cost subject to a completion time constraint under the assumption that the time–cost trade-off is a continuous linear function. This formulation has two sets of decision variables. The first set corresponds to the number of time periods each activity is shortened, whereas the second pertains to the early finish time for each activity (we note that some authors define the second set in terms of the early *start* time for each activity). There are constraint sets to ensure (i) the project completion time condition is satisfied, (ii) no activity is shortened by more than the difference between its normal and crash time, and (iii) the early finish time variable for each activity is linked to its corresponding crashing variable as well as the early finish times of its immediate predecessors.

Although the traditional LP formulation for project crashing is both interesting and useful, many students struggle to obtain a conceptual understanding of the formulation. In large part, this is attributable to the constraints that link the crashing decisions to the early finish times. We propose an alternative LP formulation that facilitates a more straightforward conceptualization of project network crashing decisions. This alternative formulation is grounded in the examination of all paths through the network. In addition to providing a more concrete understanding of the problem, the alternative formulation has the following advantages: (i) it requires only half the number of decision variables as the traditional formulation, (ii) it requires fewer constraints for most project networks found in introductory textbooks, and (iii) it establishes the "all-possible-paths" concept that is sometimes used for both discrete (nonlinear) project crashing problems and the computation of project completion time probabilities in PERT analyses (see, for example, Stevenson 2018).

Section 2 provides a formal statement of the traditional and alternative LP formulations for project network crashing as well as a comparison of the two formulations using a small synthetic network as an example. Section 3 presents the results of an experimental study that was designed to assess the relative preferences of students (in an MBA operations management class) for the two formulations. In light of the merits of spreadsheets for project network analysis (Seal 2001, Ragsdale 2003), Section 4 describes an Excel workbook that (a) generates all paths in the network and reports their length, (b) allows students to manually evaluate the effects of crashing decisions on cost and completion time, and (c) generates a template conducive to the LP formulation of the alternative formulation. We illustrate the worksheets of the workbook using the network from Section 2. Section 5 reports results obtained using the workbook for a larger, real-world network from the literature. Section 6 demonstrates the extension of the alternative formulation to the discrete (nonlinear) project network crashing problem. The paper concludes in Section 7 with a summary and discussion of the strengths and limitations of the two formulations.

2. Alternative LP Formulations for Project Network Crashing

The traditional and alternative formulations for the project network crashing problem share much of the same notation. We begin this section with precise definitions of the relevant parameters, sets, and decision variables. This is followed by the presentation of the traditional and alternative formulations, respectively. We conclude the section with a small numerical example to facilitate comparison of the formulations.

2.1. Parameters and Sets

n: the number of activities, indexed $j = 1, 2,\ldots, n$

τ_{Nj}: the normal time for activity j (for $1 \le j \le n$)

τ_{Cj}: the crash time (i.e., minimum possible completion time) for activity j (for $1 \le j \le n$)

ω_{Nj}: the cost for activity j at the normal time (for $1 \le j \le n$)

ω_{Cj}: the cost for activity j at the crash time, such that $\omega_{Cj} \ge \omega_{Nj}$ (for $1 \le j \le n$)

m_j: the maximum number of time periods that activity j can be shortened

$$m_j = (\tau_{Nj} - \tau_{cj}) \quad \forall\, 1 \le j \le n, \tag{1}$$

c_j: the per-time period crashing cost for activity j (for $1 \le j \le n$), which is

$$c_j = \begin{cases} \dfrac{(\omega c_j - \omega N_j)}{m_j} & \text{if } m_j > 0 \\ 0 & \text{if } m_j = 0 \end{cases} \quad \forall\, 1 \le j \le n, \tag{2}$$

T: the target project completion time

Q_j: the set of activities that are immediate predecessors for activity j (for $1 \le j \le n$)

G: the set of activities with no successors

2.2. Decision Variables

x_j: the early finish time for activity j (for $1 \le j \le n$)

y_j: the number of time periods activity j is crashed/shortened (for $1 \le j \le n$)

2.3. Traditional Formulation (F1)

With the parameters, sets, and decision variables in place, the traditional formulation for project network crashing (F1) can be stated as follows:

$$\min: Z_1 = \sum_{j=1}^{n} c_j y_j, \tag{3}$$

subject to

$$y_j \le m_j \qquad \forall\, 1 \le j \le n, \tag{4}$$

$$x_j \le T \qquad \forall\, j \in G, \tag{5}$$

$$x_j \ge (\tau_{Nj} - y_j) + x_k \qquad \forall\, 1 \le j \le n,\, k \in Q_j, \tag{6}$$

$$x_j \ge 0 \qquad \forall\, 1 \le j \le n, \tag{7}$$

$$y_j \ge 0 \qquad \forall\, 1 \le j \le n. \tag{8}$$

The objective function, Z_1, of F1 in Equation (3) is the total crashing cost, which is to be minimized. Constraint set (4) ensures that the number of time periods each activity is crashed does not exceed its maximum possible value. Constraint set (5) requires that all terminal activities (i.e., those with no successors) are finished within the target project completion time (T). Constraint set (6) guarantees, for each activity j and each of its immediate predecessors ($k \in Q_j$), that the early completion time for activity j equals or exceeds the early completion time for its predecessor plus its own normal completion time minus the number of time periods it is crashed. Constraint sets (7) and (8) are nonnegativity restrictions on the decision variables.

2.4. Alternative Formulation (F2)

When transitioning from the traditional formulation (F1) to the alternative formulation (F2), the x_j variables and constraint sets (5)–(7) are eliminated. We define p as the number of paths through the network, L_i as the length (total completion time based on normal times) for path i (for all $1 \leq i \leq p$), and S_i as the set of all activities on path i (for all $1 \leq i \leq p$). Formulation F2 seeks to minimize (3) subject to constraint sets (4), (8), and the following constraint set

$$\sum_{j \in S_i} y_j \geq L_i - T \quad \forall\, 1 \leq i \leq p : L_i > T. \tag{9}$$

Formulation F2 has one half the variables of F1. It also replaces constraint set (6), which is arguably the most conceptually challenging for students to grasp, with constraint set (9). The interpretation of constraint set (9) is as follows: For each path (i) such that the length of path i exceeds T (i.e., $L_i > T$), the activities on path i must be shortened by enough time to reduce the length of the path from L_i to T. For most project networks in management science textbooks, the number of constraints in (9) is fewer than the number of constraints in (6). Although it is easy to synthetically construct high-density (i.e., high interdependence among activities) networks in which the number of constraints in (9) greatly exceeds the number of constraints in (6), such networks are not particularly relevant for pedagogical purposes. Moreover, many real-world networks are low-to-moderate density, and accordingly, F2 is applicable for such networks. The equivalence of the F1 and F2 formulations is demonstrated in the appendix.

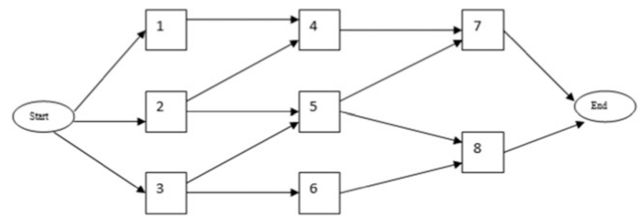
For completeness, as is the case for F1, F2 can be modified to minimize completion time subject to a budget constraint as opposed to minimizing cost subject to a time constraint. This is accomplished by changing T to a decision variable, changing the objective function to min: T, and limiting the amount spent on crashing to a specified budget value (B) via the constraint

$$\sum_{j=1}^{n} c_j y_j \leq B. \tag{10}$$

2.5. A Small Numerical Example

To compare formulations F1 and F2, we provide an example for a small network comparable to the size of those commonly encountered in introductory management science textbooks. The precedence diagram for the example network is displayed in Figure 1.

Figure 1. Project Network Diagram for the Small Numerical Example



The time and cost parameters, sets, and paths are provided in Table 1. The LP formulations for the example are provided in Figure 2.

The F2 formulation in the bottom panel of Figure 2 is clearly more compact than the F1 formulation in the top panel. Although the objective functions (3) and upper bound constraints (4) for crashing activities are the same for the two formulations, the F1 formulation model requires 14 additional constraints (12 of these are to link the early finish times to the crashing decisions), whereas the F2 formulation requires only five (one for each path of length greater than $T = 20$). Although there are seven paths in the network, two are excluded in Figure 2 (i.e., paths 3 and 5 from Table 1) because they are already 20 weeks or less. Moreover, it is evident that the F2 formulation does not require the x_j variables.

3. An Experiment to Study Students' Relative Preferences for F1 and F2

3.1. Experimental Setting

We conducted an experimental study to assess the relative preferences of students for the F1 and F2 formulations. The setting for our study was an on-campus MBA operations management course with 43 students (three students missed the exam, so the sample size was actually 40). We provided the students with a document that described the basic problem of crashing project networks, and the F1 and F2 formulations were presented using an example. The F1 formulation was presented first in the document. A second worked example was provided at the end of the document, again demonstrating the F1 formulation followed by the F2 formulation.

We told the students that the two formulations are equivalent and that either formulation could be used for exam purposes. We made no attempt to bias the students toward either formulation in the document. However, the instructor of the course (the second author of this manuscript) answered any student questions about the formulations in the classroom. The instructor gave the students an LP crashing problem on the exam that covered the topic of project scheduling. We counted the number of students using each formulation and the number of students who got the formulation 100% correct.

Table 1. Project Time and Cost Data for the Small Numerical Example

Activity (j)	1	2	3	4	5	6	7	8
Normal cost (ω_{Nj})	3,000	1,200	1,800	4,200	1,600	1,800	1,400	3,000
Crash cost (ω_{Cj})	5,800	2,000	3,000	6,000	2,500	2,400	3,000	5,000
Normal time in weeks (τ_{Nj})	10	7	8	9	7	5	5	9
Crash time in weeks (τ_{Cj})	6	5	4	6	6	3	3	5
Maximum crash time in weeks (m_j)	4	2	4	3	1	2	2	4
Per week crash cost (c_j)	700	400	300	600	900	300	800	500

$n = 8$, $p = 7$, $T = 20$
$G = \{7, 8\}$, $Q_1 = Q_2 = Q_3 = \varnothing$
$Q_4 = \{1, 2\}$, $Q_5 = \{2, 3\}$, $Q_6 = \{3\}$
$Q_7 = \{4, 5\}$, $Q_8 = \{5, 6\}$
Path $i = 1$: $S_1 = \{1, 4, 7\}$, $L_1 = 24$
Path $i = 2$: $S_2 = \{2, 4, 7\}$, $L_2 = 21$
Path $i = 3$: $S_3 = \{2, 5, 7\}$, $L_3 = 19$
Path $i = 4$: $S_4 = \{2, 5, 8\}$, $L_4 = 23$
Path $i = 5$: $S_5 = \{3, 5, 7\}$, $L_5 = 20$
Path $i = 6$: $S_6 = \{3, 5, 8\}$, $L_6 = 24$
Path $i = 7$: $S_7 = \{3, 6, 8\}$, $L_7 = 22$

3.2. Experimental Results

Two of the 40 students (5%) did not provide an answer to the formulation question. Three students (7.5%) used formulation F1. The vast majority of the students, 35 of 40 (87.5%), used formulation F2. Thus, it was unequivocally clear that students exhibited a preference for F2 as the ratio of its selection to that of F1 was more than 10:1.

Because of the limited usage of the F1 formulation, it is not possible to make a statistical comparison of the percentages of students who produced a correct formulation using the F1 and F2 formulations. Only one of the three students (33%) using the F1 formulation produced a correct formulation. Contrastingly, 28 of the 35 (80%) students using the F2 formulation provided a perfectly correct answer. Of the seven students who used the F2 formulation but did not produce a perfectly correct answer, (i) one student merely reversed the signs on the path constraints, (ii) one student merely reversed the signs on the crashing limit constraints, (iii) two students failed to identify one of the paths, and (iv) three students provided a formulation that had major problems with the objective function and one or more constraint sets.

4. Spreadsheet Implementation of F2

4.1. Spreadsheet Overview

We have developed an Excel workbook for implementing F2. The workbook makes use of a VBA macro (comppath) for identifying all of the paths in the network, allowing students to manually investigate crashing decisions, and for preparing input for the LP formulation. For networks that have multiple activities with no immediate predecessors, a dummy start node is added at the beginning of the network. Likewise, for networks that have multiple activities with no immediate successors, a dummy end node is included at the end of the network. The example in Figure 1 is illustrative of a situation in which both a dummy start activity and a dummy ending activity

Figure 2. LP Formulations for the Small Numerical Example

Min: $700y_1 + 400y_2 + 300y_3 + 600y_4 + 900y_5 + 300y_6 + 800y_7 + 500y_8$

Subject to:

$y_1 \le 4 \quad y_2 \le 2 \quad y_3 \le 4 \quad y_4 \le 3 \quad y_5 \le 1 \quad y_6 \le 2 \quad y_7 \le 2 \quad y_8 \le 4$

$x_7 \le 20 \quad x_8 \le 20$

$x_1 \ge (10 - y_1) \quad x_4 \ge (9 - y_4) + x_1 \quad x_5 \ge (7 - y_5) + x_3 \quad x_7 \ge (5 - y_7) + x_4 \quad x_8 \ge (9 - y_8) + x_5$

$x_2 \ge (7 - y_2) \quad x_4 \ge (9 - y_4) + x_2 \quad x_6 \ge (5 - y_6) + x_3 \quad x_7 \ge (5 - y_7) + x_5 \quad x_8 \ge (9 - y_8) + x_6$

$x_3 \ge (8 - y_3) \quad x_5 \ge (7 - y_5) + x_2$

Min: $700y_1 + 400y_2 + 300y_3 + 600y_4 + 900y_5 + 300y_6 + 800y_7 + 500y_8$

Subject to:

$y_1 \le 4 \quad y_2 \le 2 \quad y_3 \le 4 \quad y_4 \le 3 \quad y_5 \le 1 \quad y_6 \le 2 \quad y_7 \le 2 \quad y_8 \le 4$

$y_1 + y_4 + y_7 \ge 4 \quad y_2 + y_4 + y_7 \ge 1 \quad y_2 + y_5 + y_8 \ge 3 \quad y_3 + y_5 + y_8 \ge 4 \quad y_3 + y_6 + y_8 \ge 2$

Note. The top panel contains the F1 formulation and the bottom panel the F2 formulation.

are necessary. The purpose of these dummy activities is to facilitate the enumeration of paths by having a single starting activity and a single ending activity.

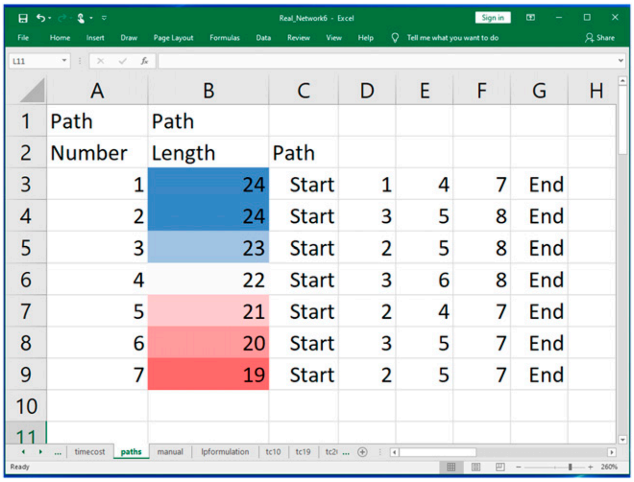
The workbook contains several project networks, each in their own separate worksheet. One of these, worksheet tc10, is the small synthetic network that we described in Section 2. It is noteworthy that, because this network had multiple activities with immediate predecessors and multiple activities with no successors, both a dummy starting activity and a dummy ending activity were added, thus increasing the total number of activities from 8 to 10. Two of the networks are medium-sized networks (approximately 20 activities) that we obtained from a service operations management textbook (Murdick et al. 1990): (1) worksheet tc19, an auditing project network originally published by Krogstad et al. (1977), and (2) worksheet tc20, a computer installation network (Moscove and Simkin 1987). We also include a 68-activity network pertaining to the structural/electrical update of the Blackhawk helicopter (worksheet tc68), which we obtained from a master's thesis by Fortier (2006).

To apply the comppath macro to one of these networks, the user only needs to select the worksheet containing the desired network and click the button that says "Generate Paths." The comppath macro reads the following data from the time–cost worksheet: (i) n: the number of activities, (ii) T: the target completion time, (iii) τ_{Nj}: the normal completion time for each activity, (iv) m_j: the maximum number of time periods that each activity can be crashed, (v) c_j: the per time period crashing costs for each activity, (vi) the number of immediate predecessors for each activity, and (vii) the set of immediate predecessors for each activity. The macro writes output to three worksheets: paths, manual, and lpformulation.

4.2. Path Generation (the `paths` Worksheet)

The comppath macro generates all p paths in the project network using a depth-first search algorithm. The length of each path (L_i) is also computed and stored. A bubble sort is used to rank the paths in descending order of length. To enable the analyst to have a frame of reference for all of the paths in the network, each of the p paths and their corresponding lengths are written to the worksheet `paths`. Color (red, white, blue) shading is used to distinguish the paths by length with the shortest paths in dark red and the longest paths in dark blue. The paths worksheet associated with the example discussed in Section 2 is provided in Figure 3. All seven paths, which are now numbered longest to shortest, and their lengths are reported.

Figure 3. The `Paths` Worksheet for the Network Associated with Figure 1 and Table 1



	A	B	C	D	E	F	G	H
1	Path	Path						
2	Number	Length	Path					
3	1	24	Start	1	4	7	End	
4	2	24	Start	3	5	8	End	
5	3	23	Start	2	5	8	End	
6	4	22	Start	3	6	8	End	
7	5	21	Start	2	4	7	End	
8	6	20	Start	3	5	7	End	
9	7	19	Start	2	5	7	End	
10								
11								

4.3. Manual Crashing Analysis (the `manual` Worksheet)

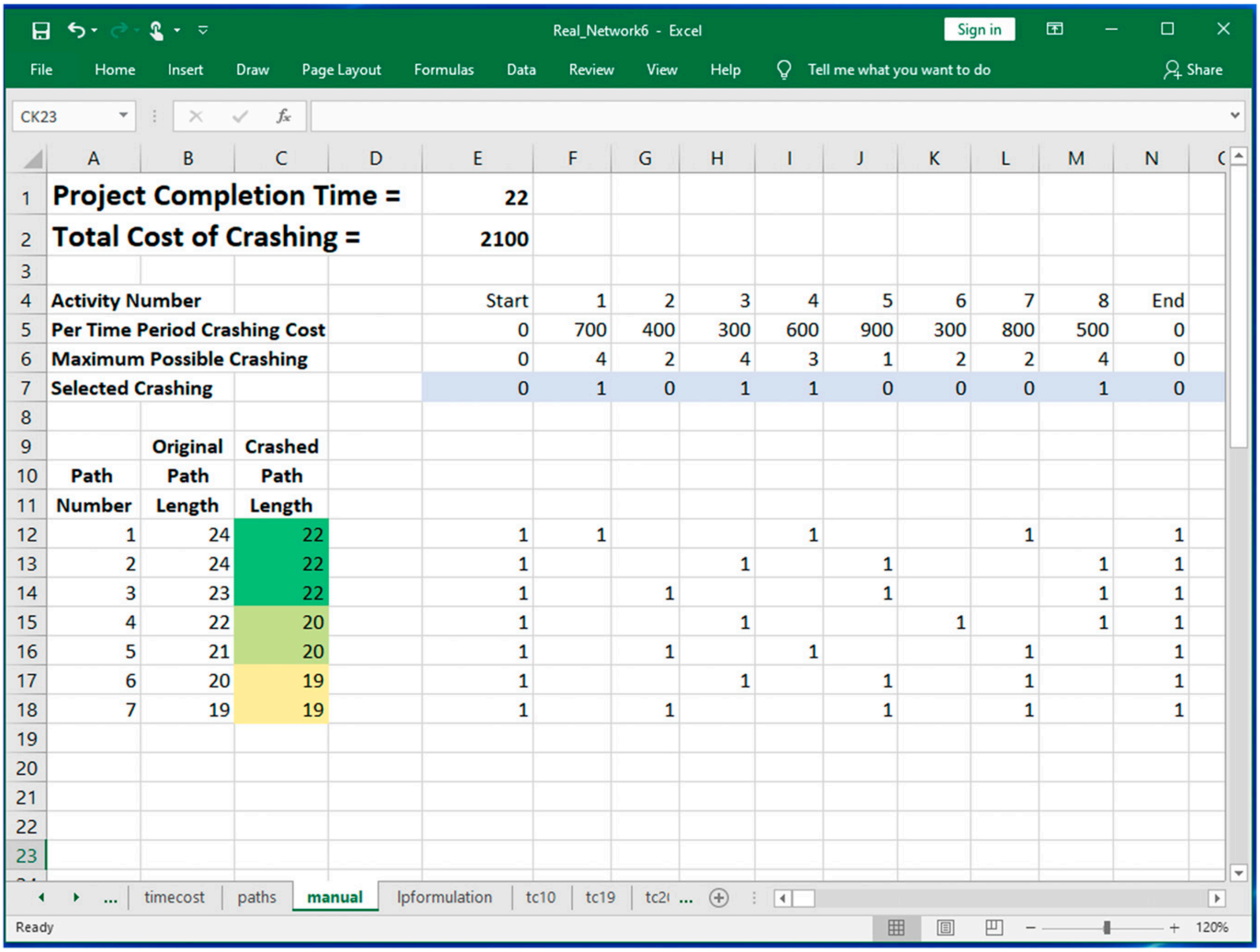
The `manual` worksheet also contains all of the paths; however, this worksheet also has rows that contain, for each activity, the activity number (row 4), the per time period crashing cost (row 5), the maximum possible crashing (row 6), and the selected crashing decision (row 7). Students can manually alter the crashing decisions in row 7 to evaluate their implications for total crashing cost (cell E2) and project completion time (cell E1). The path lengths are dynamically changed as students alter the cells in row 7, and as in the paths worksheet, color-coding of the cells is used to differentiate the lengths of the paths. Figure 4 presents the manual worksheet for the example from Section 2.

4.4 The LP Formulation for F2 (the `lpformulation` Worksheet)

The `lpformulation` worksheet contains a basic template for the F2 formulation, including rows containing column labels, per period crashing costs, maximum crashing, and the y_j variables. In addition, the worksheet contains the constraint right-hand sides and constraint coefficients for all of the paths for which $L_i > T$. Thus, whereas the paths worksheet contains all of the paths and their lengths, the `lpformulation` worksheet contains only those paths necessary for the formulation. For most applications, the number of paths with $L_i > T$ is appreciably smaller than p.

Figures 5 and 6 provide, respectively, for the example from Section 2, screenshots of the `lpformulation` worksheet and the corresponding Solver dialog box. Figure 5 reveals that the optimal solution to the F2 formulation is $y_1^* = y_3^* = 1$, $y_4^* = y_8^* = 3$, and all

Figure 4. The `manual` Worksheet for the Network Associated with Figure 1 and Table 1



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Project Completion Time =				22									
2	Total Cost of Crashing =				2100									
3														
4	Activity Number				Start	1	2	3	4	5	6	7	8	End
5	Per Time Period Crashing Cost				0	700	400	300	600	900	300	800	500	0
6	Maximum Possible Crashing				0	4	2	4	3	1	2	2	4	0
7	Selected Crashing				0	1	0	1	1	0	0	0	1	0
8														
9		Original	Crashed											
10	Path	Path	Path											
11	Number	Length	Length											
12	1	24	22		1	1			1			1		1
13	2	24	22		1			1		1			1	1
14	3	23	22		1		1			1			1	1
15	4	22	20		1			1			1		1	1
16	5	21	20		1		1		1			1		1
17	6	20	19		1			1		1		1		1
18	7	19	19		1		1			1		1		1

other $y_j^* = 0$. The total crashing cost is $Z_1^* = \$4{,}300$. The template for the LP formulation contains all seven paths even though two of the paths already satisfied the target completion time of $T = 20$.

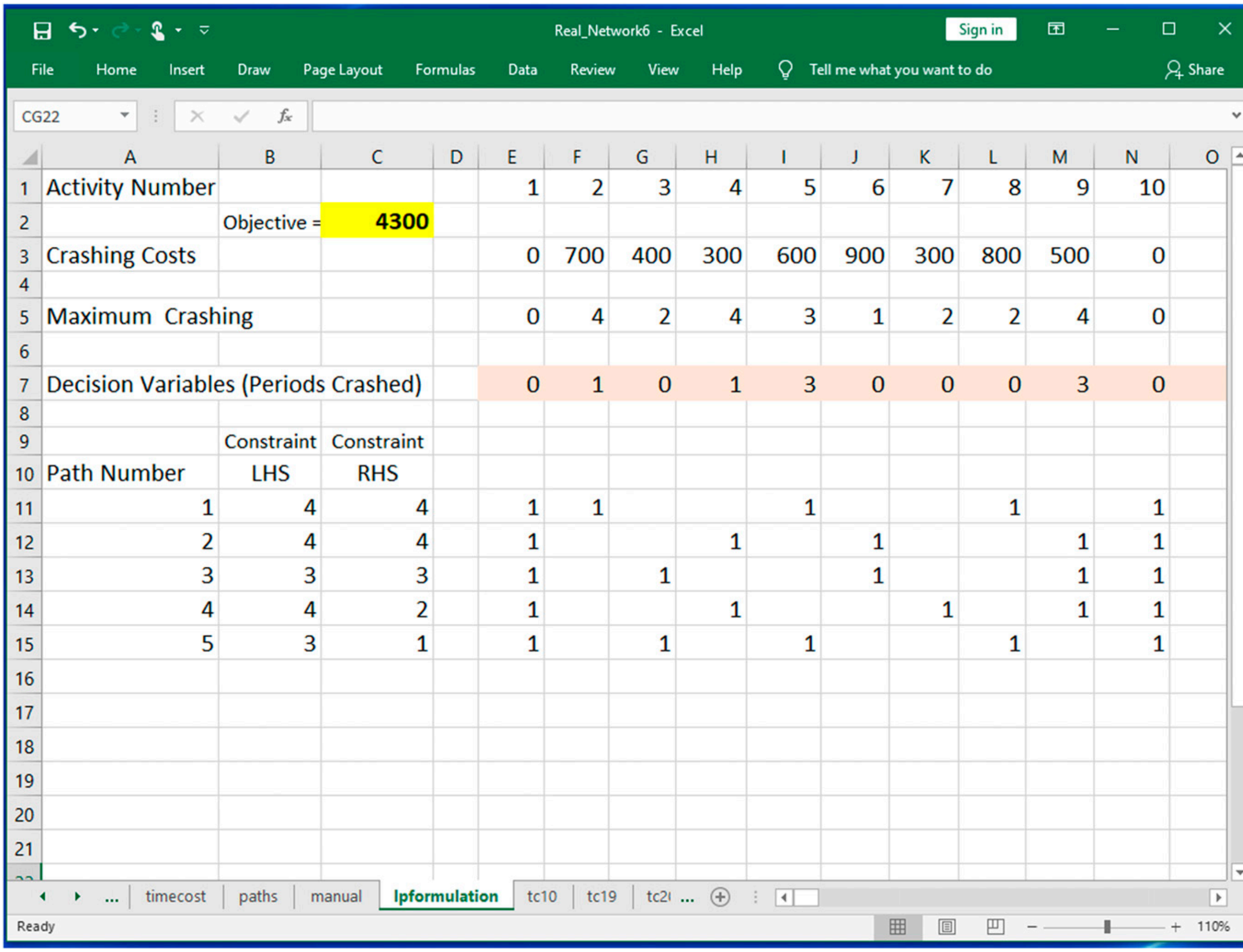
5. Application to a Larger (Real-World) Network

5.1. The Blackhawk Helicopter Upgrade Network (Fortier 2006)

For our second example, we consider a larger, real-world network published by Fortier (2006) that pertains to the structural/electrical upgrade of the UH-60A (Blackhawk) helicopter. The network contains $n = 68$ activities. Fortier (2006) provides a display of the precedence diagram for the network. The network is rather sparse as the maximum number of immediate predecessors is six (for activity 58), and most activities have only one immediate predecessor. We selected crashing cost scenario 1 (Fortier 2006) for demonstration purposes, which is one of several scenarios described by the author.

Fortier (2006) presents the constraint set associated with an F1 formulation of the project network crashing problem for scenario 1. There are several inconsistencies associated with the constraint set. First, the completion time constraint for activity 1 is missing. Because x_1 cannot be crashed, a constraint should be added such as $x_1 \geq 8$ or $x_1 = 8$. Second, the "TAT" constraint (i.e., constraint (4) of the F1 formulation) to ensure that the project is completed within 640 hours should be $x_{68} \leq 640$ rather than $y_{68} \leq 640$. Third, and most importantly, there is some discord between the constraint set and the precedence diagram (cf. Fortier 2006). For example, there is a constraint pertaining to activity 21 as an immediate predecessor for activity 27 (i.e., $x_{27} - x_{21} + y_{27} \geq 24$); however, the precedence diagram shows that activity 22 (not 21) is the immediate predecessor of activity 27. Similarly, there is a constraint for activity 25 as an immediate predecessor for activity 29 (i.e., $x_{29} - x_{25} \geq 104$); however, the precedence diagram shows that activity 21 (not 25) is the immediate predecessor of activity 29.

Figure 5. The `lpformulation` Worksheet for the Network Associated with Figure 1 and Table 1



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Activity Number				1	2	3	4	5	6	7	8	9	10	
2		Objective =	4300												
3	Crashing Costs				0	700	400	300	600	900	300	800	500	0	
4															
5	Maximum Crashing				0	4	2	4	3	1	2	2	4	0	
6															
7	Decision Variables (Periods Crashed)				0	1	0	1	3	0	0	0	3	0	
8															
9		Constraint	Constraint												
10	Path Number	LHS	RHS												
11	1	4	4		1	1			1			1		1	
12	2	4	4		1			1		1			1	1	
13	3	3	3		1		1			1			1	1	
14	4	4	2		1			1			1		1	1	
15	5	3	1		1		1		1			1		1	

Our reason for raising the inconsistencies between the project network diagram (Fortier 2006) and the constraint set (Fortier 2006) is not to diminish the interesting and important contribution provided by the study. However, it is essential to note the inconsistencies because our "consistent" analysis, which follows the precedence diagram, yields slightly different results, and the differences are likely attributable to the inconsistencies.

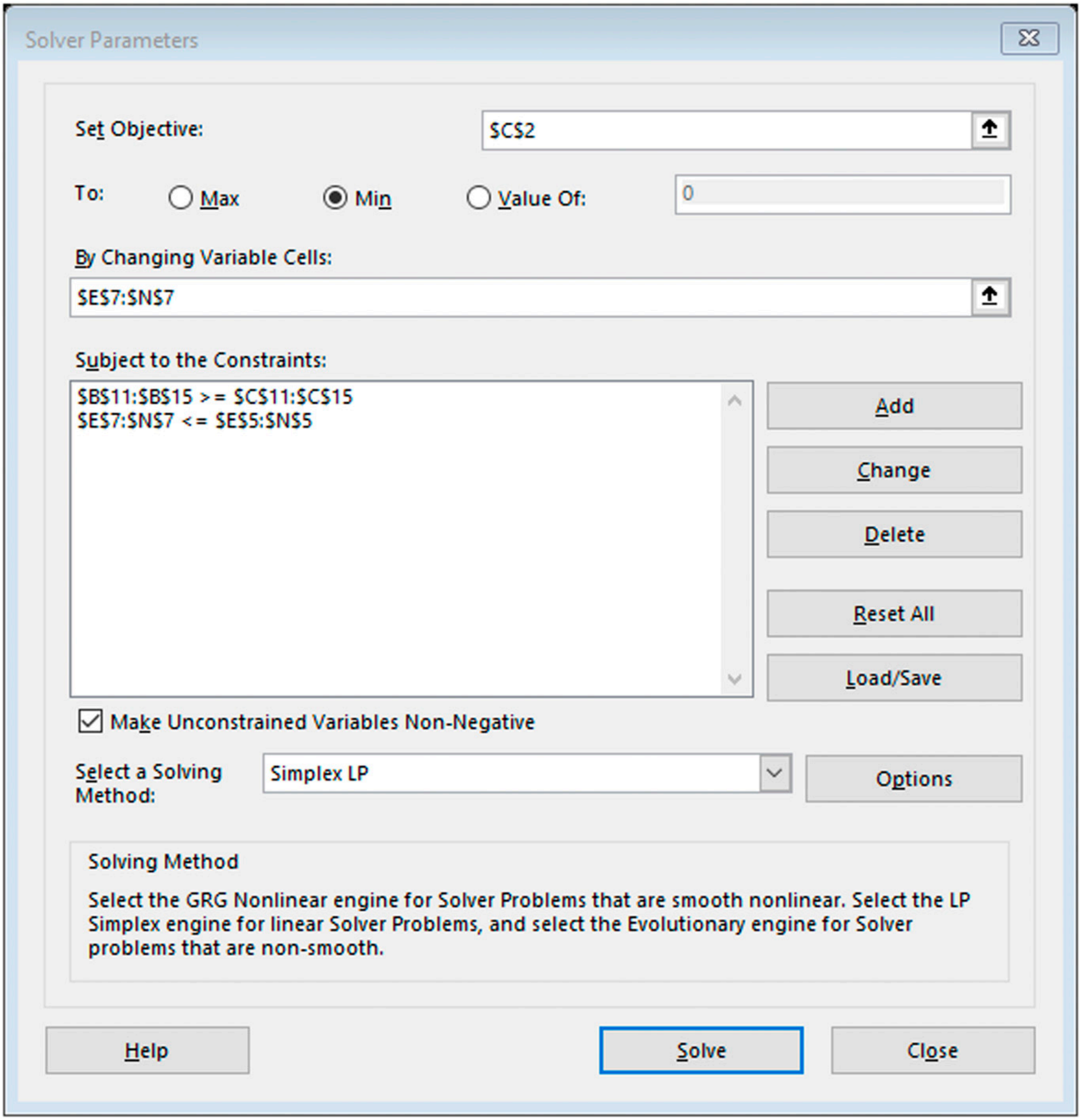
5.2. Spreadsheet Analysis of the Blackhawk Helicopter Upgrade Network

We use the spreadsheet described in Section 4 to analyze the Blackhawk helicopter upgrade network. The paths worksheet in Figure 7 reveals that there are $p = 48$ paths in the network. The fact that $p < n$ is attributable to the sparsity of the network. There are two paths that result in the maximum length of 644 hours. The two paths differ only with respect to whether activity 64 or activity 65 is the immediate successor to activity 63. It is also important to observe that 24 of the 48 paths are of a duration of 400 hours or fewer. Even with maximal crashing, it would be impossible to reduce the completion time to 400 hours, and accordingly, at least half of the paths are not necessary for the F2 formulation.

The manual worksheet in Figure 8 shows that crashing activity 4 by three hours reduces completion time from 644 to 641 hours at a cost of \$44,366.61. Although manually crashing activities might seem tedious for a project with 68 activities, the process is simplified because only 24 of the 68 activities can be crashed, and several of these have no impact on the longest paths of the network.

The `lpformulation` worksheet in Figure 9 shows the results obtained when using the F2 LP formulation to reduce project completion time from 644 hours to $T =$ 592 hours. Only 17 of the 48 paths are necessary for the LP formulation because the other 31 paths are already less than 592 hours. The optimal solution to the F2

Figure 6. Solver Dialog Box for the Network Associated with Figure 1 and Table 1

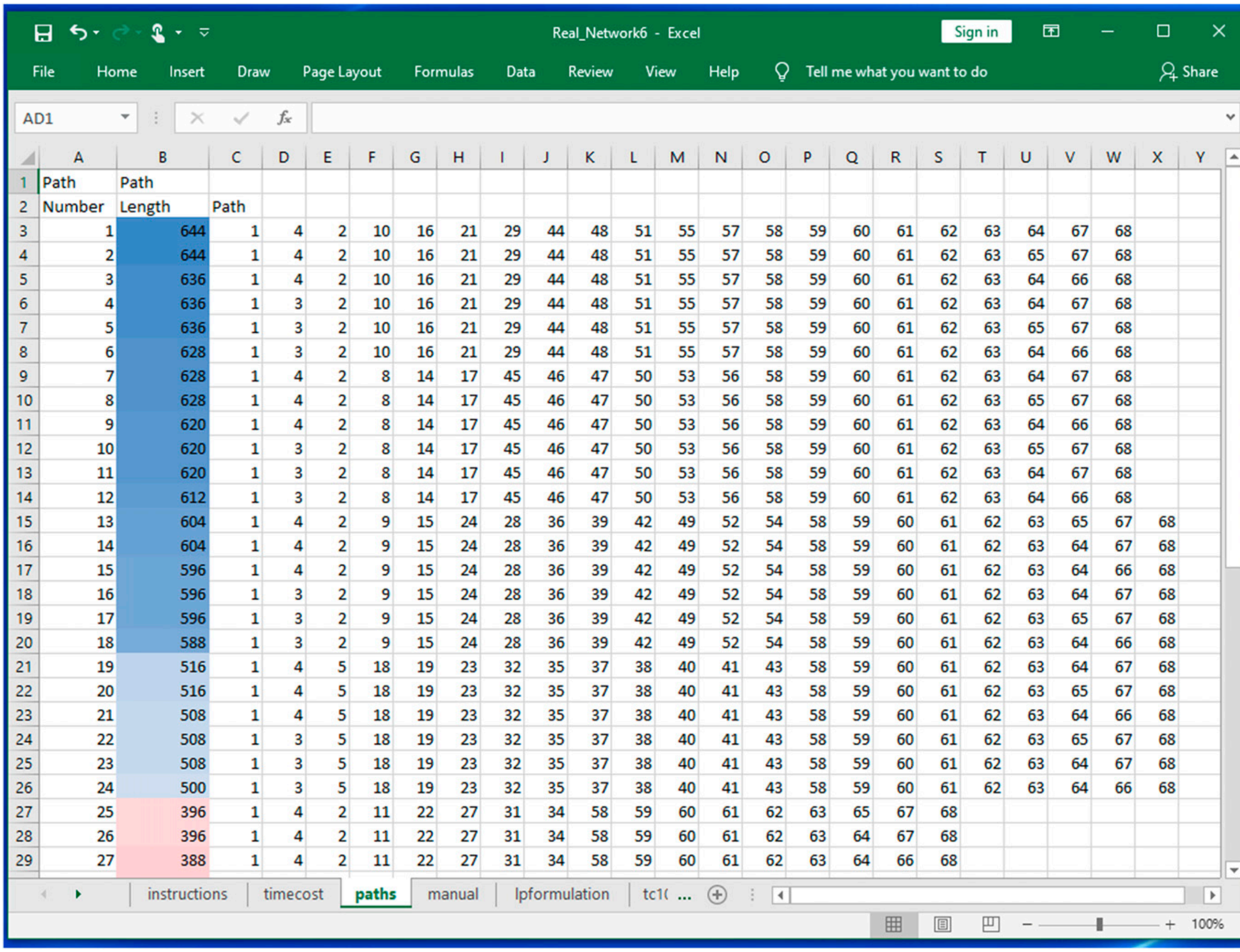


formulation is $y^*_{63} = 20, y^*_{55} = 8, y^*_{57} = y^*_{59} = 6, y^*_4 = y^*_{67} = 4, y^*_{58} = 2$, and $y^*_{60} = y^*_{62} = 1$. The total crashing cost is \$696,949.24.

Fortier (2006) reports that he obtained a feasible solution for a target completion time of $T = 584$ hours. Using the manual worksheet and setting the crashing decision variable for each activity to its maximum possible value, it is easy for us to establish that the minimum possible completion time for our "consistent" version of the formulation is $T = 590$ hours. The optimal solution for $T = 590$ hours is the same as the one for $T = 592$ hours except that $y^*_{59} = 8$ (its maximum value) rather than $y^*_{59} = 6$. The extra two hours of crashing for activity 59 costs \$43,992.14 (\$21,996.07 per hour), which increases the total crashing cost to \$696,949.24 + \$43,992.14 = \$740,931.38.

INFORMS Transactions on Education, 2021, vol. 21, no. 2, pp. 82–95, © 2020 The Author(s)

Figure 7. The `paths` Worksheet for the Blackhawk Helicopter Update Network



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Path	Path																							
2	Number	Length	Path																						
3	1	644	1	4	2	10	16	21	29	44	48	51	55	57	58	59	60	61	62	63	64	67	68		
4	2	644	1	4	2	10	16	21	29	44	48	51	55	57	58	59	60	61	62	63	65	67	68		
5	3	636	1	4	2	10	16	21	29	44	48	51	55	57	58	59	60	61	62	63	64	66	68		
6	4	636	1	3	2	10	16	21	29	44	48	51	55	57	58	59	60	61	62	63	64	67	68		
7	5	636	1	3	2	10	16	21	29	44	48	51	55	57	58	59	60	61	62	63	65	67	68		
8	6	628	1	3	2	10	16	21	29	44	48	51	55	57	58	59	60	61	62	63	64	66	68		
9	7	628	1	4	2	8	14	17	45	46	47	50	53	56	58	59	60	61	62	63	64	67	68		
10	8	628	1	4	2	8	14	17	45	46	47	50	53	56	58	59	60	61	62	63	65	67	68		
11	9	620	1	4	2	8	14	17	45	46	47	50	53	56	58	59	60	61	62	63	64	66	68		
12	10	620	1	3	2	8	14	17	45	46	47	50	53	56	58	59	60	61	62	63	65	67	68		
13	11	620	1	3	2	8	14	17	45	46	47	50	53	56	58	59	60	61	62	63	64	67	68		
14	12	612	1	3	2	8	14	17	45	46	47	50	53	56	58	59	60	61	62	63	64	66	68		
15	13	604	1	4	2	9	15	24	28	36	39	42	49	52	54	58	59	60	61	62	63	65	67	68	
16	14	604	1	4	2	9	15	24	28	36	39	42	49	52	54	58	59	60	61	62	63	64	67	68	
17	15	596	1	4	2	9	15	24	28	36	39	42	49	52	54	58	59	60	61	62	63	64	66	68	
18	16	596	1	3	2	9	15	24	28	36	39	42	49	52	54	58	59	60	61	62	63	64	67	68	
19	17	596	1	3	2	9	15	24	28	36	39	42	49	52	54	58	59	60	61	62	63	65	67	68	
20	18	588	1	3	2	9	15	24	28	36	39	42	49	52	54	58	59	60	61	62	63	64	66	68	
21	19	516	1	4	5	18	19	23	32	35	37	38	40	41	43	58	59	60	61	62	63	64	67	68	
22	20	516	1	4	5	18	19	23	32	35	37	38	40	41	43	58	59	60	61	62	63	65	67	68	
23	21	508	1	4	5	18	19	23	32	35	37	38	40	41	43	58	59	60	61	62	63	64	66	68	
24	22	508	1	3	5	18	19	23	32	35	37	38	40	41	43	58	59	60	61	62	63	65	67	68	
25	23	508	1	3	5	18	19	23	32	35	37	38	40	41	43	58	59	60	61	62	63	64	67	68	
26	24	500	1	3	5	18	19	23	32	35	37	38	40	41	43	58	59	60	61	62	63	64	66	68	
27	25	396	1	4	2	11	22	27	31	34	58	59	60	61	62	63	65	67	68						
28	26	396	1	4	2	11	22	27	31	34	58	59	60	61	62	63	64	67	68						
29	27	388	1	4	2	11	22	27	31	34	58	59	60	61	62	63	64	66	68						

6. Extension to Discrete (Nonlinear) Crashing Costs

6.1. Modified F2 Formulation for Discrete Crashing Costs (F2-D)

Next, we describe how the F2 formulation can be adapted to accommodate discrete (nonlinear) crashing costs. The extended formulation (F2-D) requires the definition of the following cost parameters, which replace the c_j parameters:

b_{jh}: the additional cost that is incurred when increasing the number of time periods that activity j is crashed from h-1 to h (for $1 \le j \le n$ and $1 \le h \le m_j$)

In addition, it is necessary to replace the y_j variables with the following binary crashing variables:

v_{jh}: one if the number of time periods that activity j is crashed is increased from $h-1$ to h and $v_{jh} = 0$ otherwise (for $1 \le j \le n$ and $1 \le h \le m_j$)

With these new parameters and decision variables in place, the extended formulation for discrete project network crashing (F2-D) can be stated as follows:

$$\min : Z_2 = \sum_{j=1}^{n} \sum_{h=1}^{m_j} b_{jh} v_{jh}, \tag{11}$$

subject to

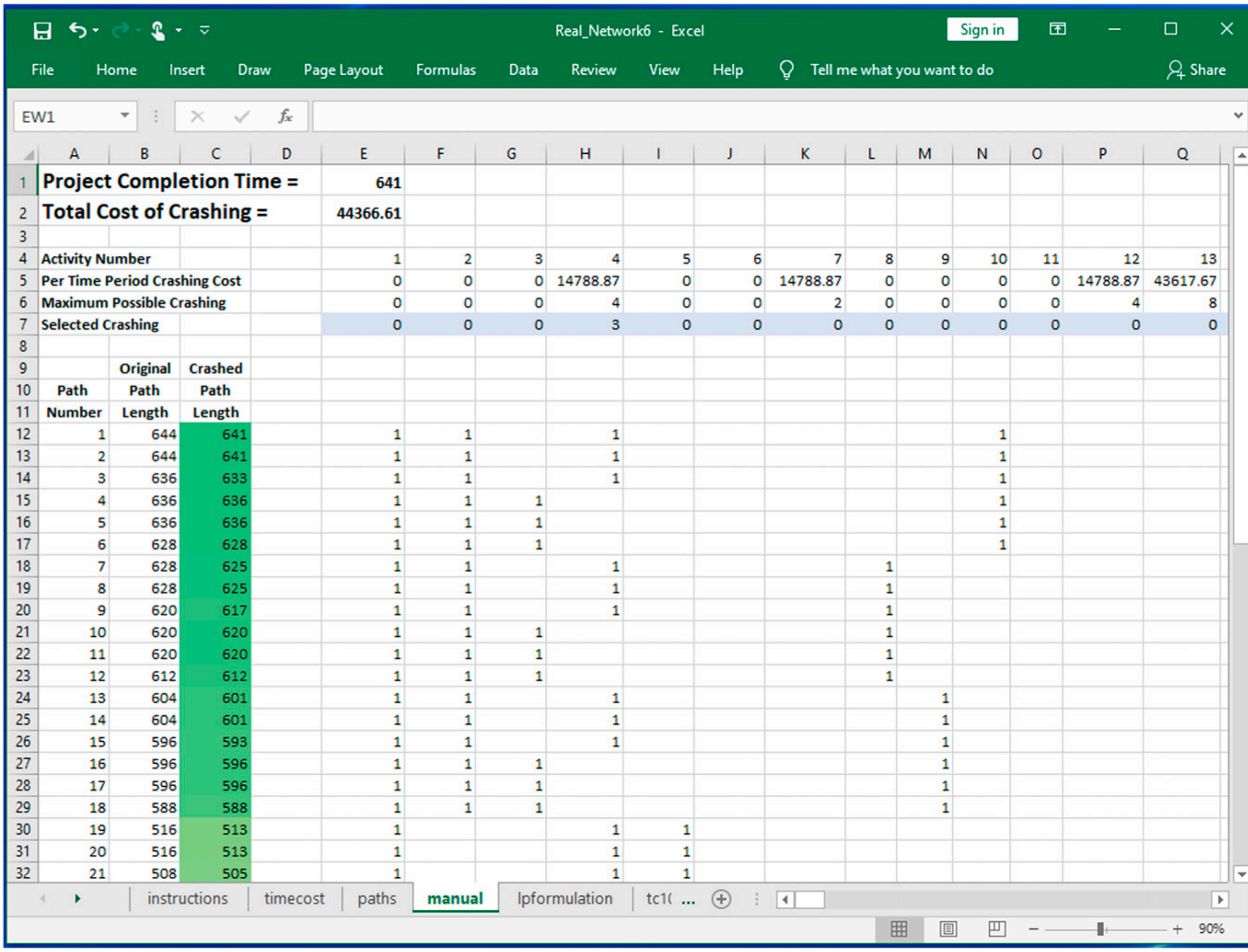
$$\sum_{j \in S_i} \sum_{h=1}^{m_j} v_{jh} \ge L_i - T \quad \forall\, 1 \le i \le p : L_i > T, \tag{12}$$

$$v_{jg} \le v_{jh} \qquad \forall\, 1 \le j \le n,\ 1 \le g > h \le m_j, \tag{13}$$

$$v_{jh} \in \{0,1\} \qquad \forall\, 1 \le j \le n,\ 1 \le h \le m_j. \tag{14}$$

The objective function, Z_2, of F2-D to be minimized in Equation (11) is the total crashing cost, which is computed as the product of the binary crashing variables and their corresponding incremental costs for shortening an activity by one additional time period.

Figure 8. The `manual` Worksheet for the Blackhawk Helicopter Update Network



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Project Completion Time =				641												
2	Total Cost of Crashing =				44366.61												
3																	
4	Activity Number				1	2	3	4	5	6	7	8	9	10	11	12	13
5	Per Time Period Crashing Cost				0	0	0	14788.87	0	0	14788.87	0	0	0	0	14788.87	43617.67
6	Maximum Possible Crashing				0	0	0	4	0	0	2	0	0	0	0	4	8
7	Selected Crashing				0	0	0	3	0	0	0	0	0	0	0	0	0
8																	
9		Original	Crashed														
10	Path	Path	Path														
11	Number	Length	Length														
12	1	644	641		1	1		1						1			
13	2	644	641		1	1		1						1			
14	3	636	633		1	1		1						1			
15	4	636	636		1	1	1							1			
16	5	636	636		1	1	1							1			
17	6	628	628		1	1	1							1			
18	7	628	625		1	1		1				1					
19	8	628	625		1	1		1				1					
20	9	620	617		1	1		1				1					
21	10	620	620		1	1	1					1					
22	11	620	620		1	1	1					1					
23	12	612	612		1	1	1					1					
24	13	604	601		1	1		1					1				
25	14	604	601		1	1		1					1				
26	15	596	593		1	1		1					1				
27	16	596	596		1	1	1						1				
28	17	596	596		1	1	1						1				
29	18	588	588		1	1	1						1				
30	19	516	513		1			1	1								
31	20	516	513		1			1	1								
32	21	508	505		1			1	1								

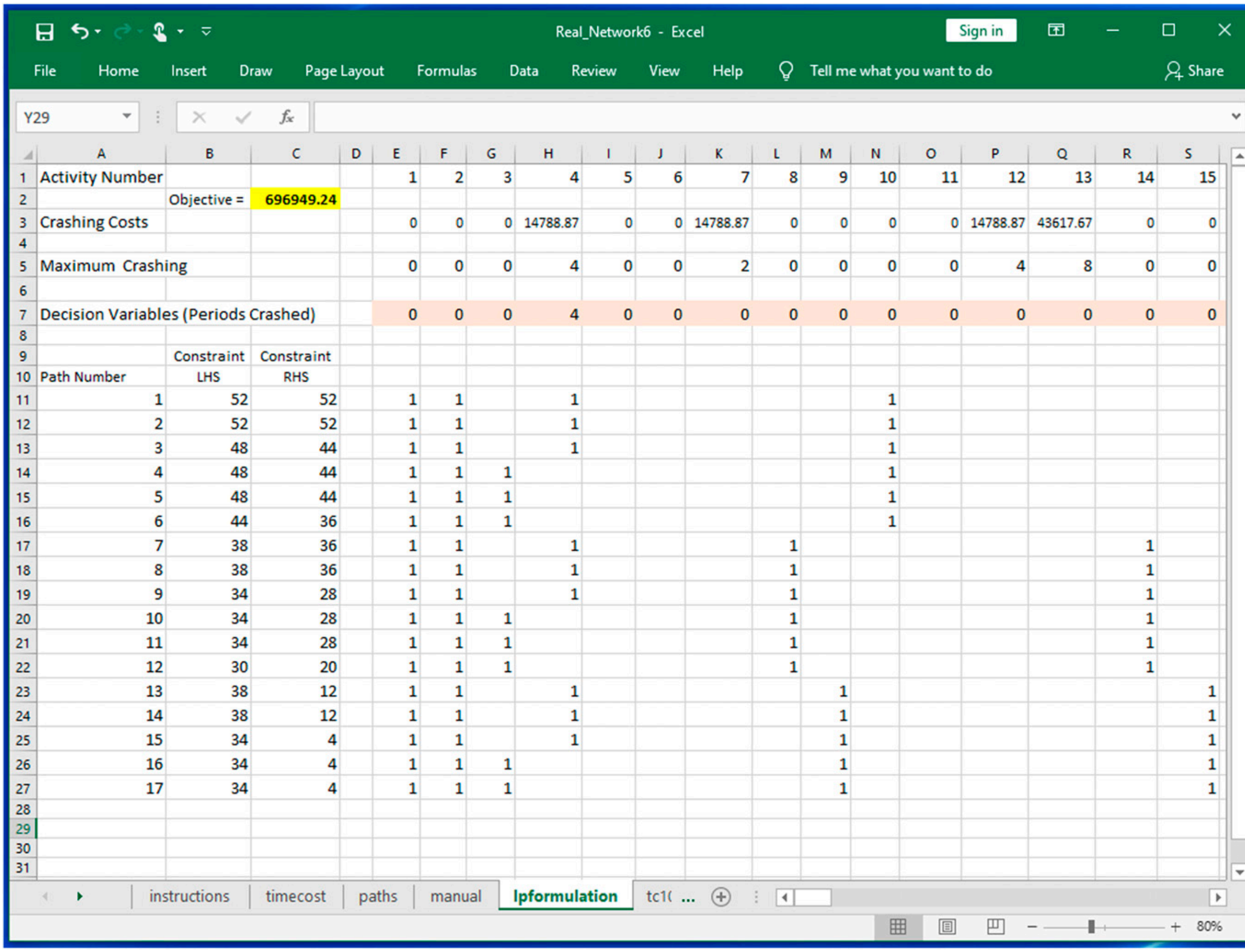
Constraint set (12) is analogous to (9) but now captures the crashing of each activity as the sum of binary variables. Constraint set (13) ensures that binary crashing variables are selected "in order." For example, suppose that the cost of crashing activity 1 is b_{11} = \$700 for the first week, and the cost is b_{12} = \$600 (additional) to shorten the activity by a second week. Without the constraint $v_{12} \leq v_{11}$, the solution to the model could shorten activity by only one week yet do so by selecting the variable corresponding to the second week (instead of the first) because it is cheaper. It should be noted that, if $b_{jg} > b_{jh}$, then the corresponding $v_{jg} \leq v_{jh}$ constraint in (13) can be safely removed. Constraint set (14) places binary restrictions on the decision variables.

6.2. A Modified Numerical Example for the Discrete Problem

The data, for example, for the discrete crashing problem are identical to the data provided in Table 1 for the example in Section 2.5 with one exception: the c_j cost parameters are replaced with b_{jh} parameters, which are displayed in rows 15–18 of the Excel template displayed in Figure 10. To illustrate, in the small numerical example in Section 2.5, the cost of crashing activity 1 was \$700 per week, and it was possible to shorten activity 1 by up to four weeks. In the modified example for the discrete crashing problem, rows 15–18 of column D in Figure 10 show that the cost of crashing activity 1 is \$700 for the first week. However, to shorten the activity by a second week costs only an additional \$600. To shorten the activity by a third week costs an additional \$1,200, and a fourth week adds another \$500. Accordingly, formulation F2-D allows for modeling of nonlinear time–cost trade-offs using a piecewise linear function. The Solver dialog box for the template shown in Figure 10 is displayed in Figure 11.

The maximum that any activity can be crashed is four weeks, and therefore, we defined four weeks of v_{jh} variables for each activity in rows 20–23 even though most activities allow fewer weeks of crashing. For example, activity 2 can be crashed by up to only two weeks. To ensure that three or four weeks of

Figure 9. The `lpformulation` Worksheet for the Blackhawk Helicopter Update Network



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Activity Number				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2		Objective =	696949.24																
3	Crashing Costs				0	0	0	14788.87	0	0	14788.87	0	0	0	0	14788.87	43617.67	0	0
4																			
5	Maximum Crashing				0	0	0	4	0	0	2	0	0	0	0	4	8	0	0
6																			
7	Decision Variables (Periods Crashed)				0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
8																			
9		Constraint	Constraint																
10	Path Number	LHS	RHS																
11	1	52	52		1	1		1						1					
12	2	52	52		1	1		1						1					
13	3	48	44		1	1		1						1					
14	4	48	44		1	1	1							1					
15	5	48	44		1	1	1							1					
16	6	44	36		1	1	1							1					
17	7	38	36		1	1		1				1						1	
18	8	38	36		1	1		1				1						1	
19	9	34	28		1	1		1				1						1	
20	10	34	28		1	1	1					1						1	
21	11	34	28		1	1	1					1						1	
22	12	30	20		1	1	1					1						1	
23	13	38	12		1	1		1					1						1
24	14	38	12		1	1		1					1						1
25	15	34	4		1	1		1					1						1
26	16	34	4		1	1	1						1						1
27	17	34	4		1	1	1						1						1

crashing are not selected, we imposed prohibitively large cost penalties of 99,999 on these variables (see cells E17 and E18 in Figure 10). We also compute, in row 3, "Total Crashing," the y_j variables as the sum of the v_{jh} variables. For example, cell D3 in Figure 10 is the sum of cells D20:D23.

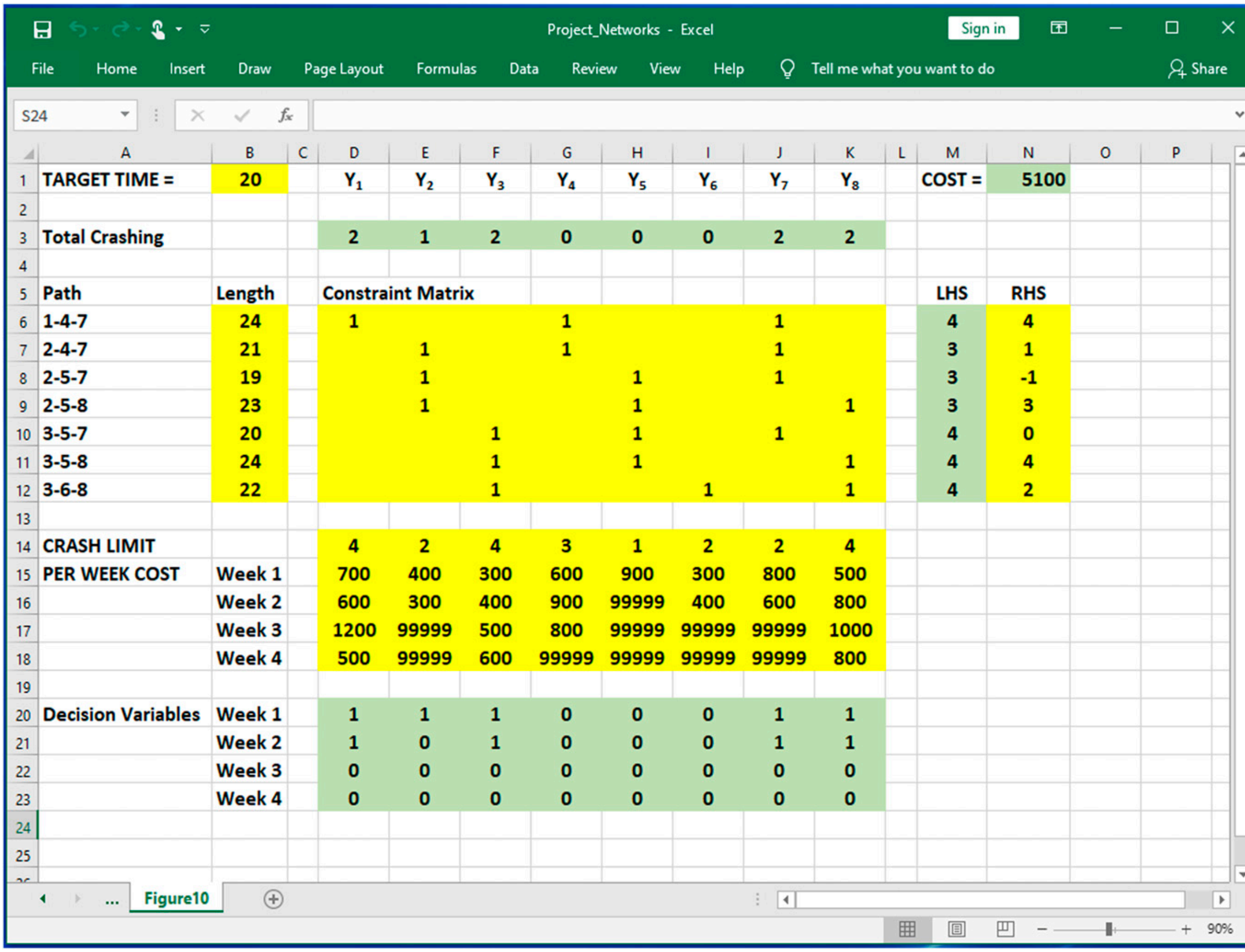
The optimal solution to the F2-D formulation is $y_1^* = y_3^* = y_7^* = y_8^* = 2$, $y_2^* = 1$, and all other variables equal to zero. For example, the binary variables for activity 1 are $v_{11}^* = v_{12}^* = 1$ and $v_{13}^* = v_{14}^* = 0$; therefore, $y_1^* = v_{11}^* + v_{12}^* = 2$. The cost incurred for crashing activity 1 by one week is \$700, and an additional \$600 is incurred for the second week. Likewise, the binary variables for activity 2 are $v_{21}^* = 1$ and $v_{22}^* = v_{23}^* = v_{24}^* = 0$. Therefore, activity 2 is crashed only one week ($v_2^* = v_{21}^* = 1$) at a cost of \$400. The total cost of crashing required to shorten the project to 20 weeks is $Z_2^* = \$5{,}100$.

It is interesting to compare the F2 and F2-D results in Figures 5 and 10, respectively. The solution in Figure 5 shows that activity 4 is crashed by three weeks; however, activity 4 is not crashed at all in the solution in Figure 10. Contrastingly, activity 7 is not crashed at all in the Figure 5 solution but is crashed by two weeks in the Figure 10 solution. Part of the explanation for this result can be deduced from examination of the 1–4–7 path. For the F2 model in Figure 5, the per week cost of crashing activity 4 (\$600) is less expensive than the corresponding costs for activities 1 (\$700) and 7 (\$800), making it the preferable option for shortening the 1–4–7 path. However, for the nonlinear cost structure for the F2-D model in Figure 10, although activity 4 is still less expensive than activities 1 and 7 for the first week of crashing, it is appreciably more expensive for the second week of crashing.

7. Discussion and Limitations

The origin of this paper was a suggestion from the first author while a student in an analytics course for which the second author was the instructor. The crux of the suggestion was that the x_j variables in F1 were unnecessary if one examined all of the paths. The instructor presented formulation F2 to the remainder of the class, and there was general agreement that F2

Figure 10. Excel Template for the F2-D Formulation of Discrete Example



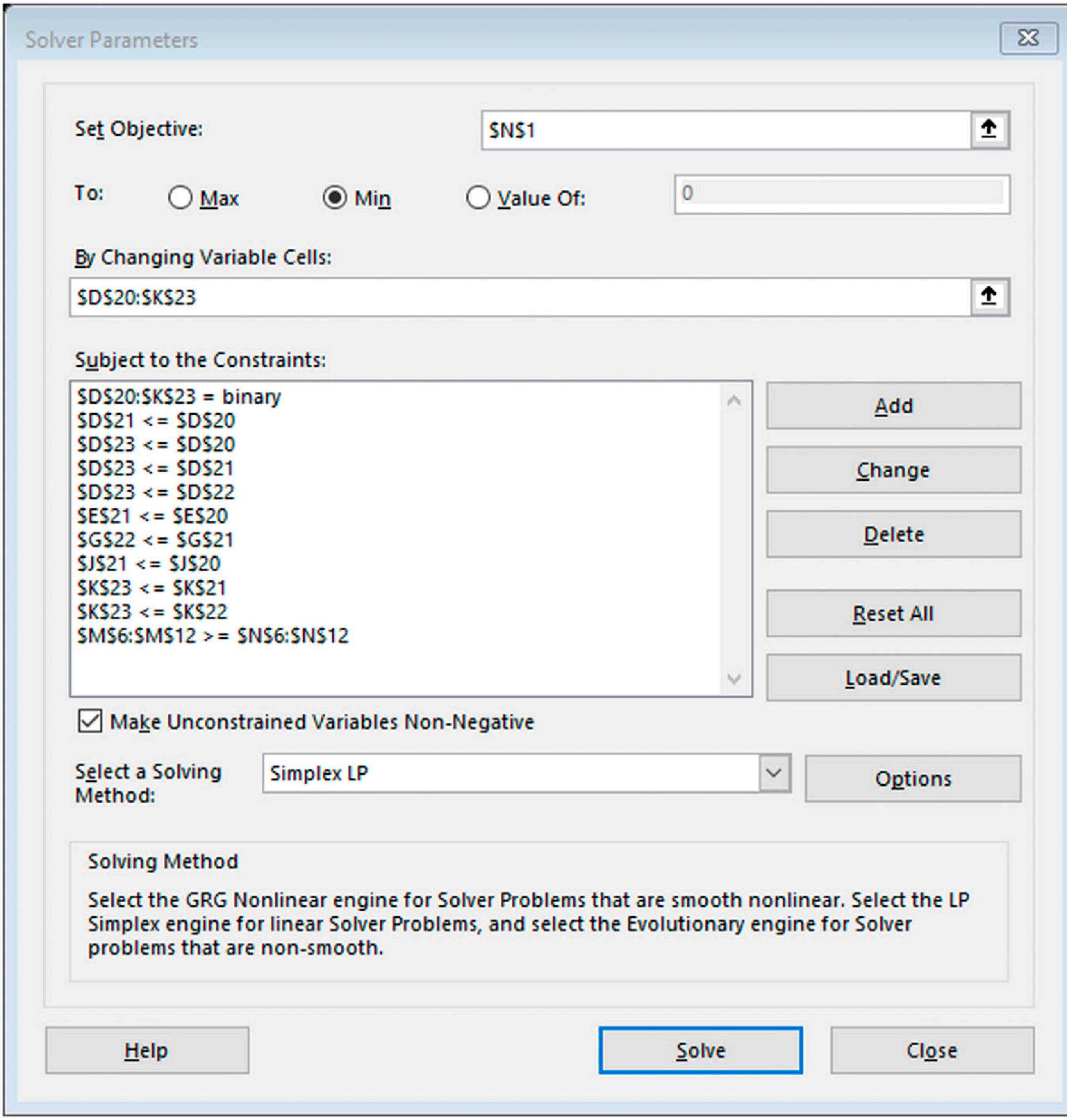
	A	B	D	E	F	G	H	I	J	K	M	N
1	TARGET TIME =	20	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	COST =	5100
2												
3	Total Crashing		2	1	2	0	0	0	2	2		
4												
5	Path	Length	Constraint Matrix								LHS	RHS
6	1-4-7	24	1			1			1		4	4
7	2-4-7	21		1		1			1		3	1
8	2-5-7	19		1			1		1		3	-1
9	2-5-8	23		1			1			1	3	3
10	3-5-7	20			1		1		1		4	0
11	3-5-8	24			1		1			1	4	4
12	3-6-8	22			1			1		1	4	2
13												
14	CRASH LIMIT		4	2	4	3	1	2	2	4		
15	PER WEEK COST	Week 1	700	400	300	600	900	300	800	500		
16		Week 2	600	300	400	900	99999	400	600	800		
17		Week 3	1200	99999	500	800	99999	99999	99999	1000		
18		Week 4	500	99999	600	99999	99999	99999	99999	800		
19												
20	Decision Variables	Week 1	1	1	1	0	0	0	1	1		
21		Week 2	1	0	1	0	0	0	1	1		
22		Week 3	0	0	0	0	0	0	0	0		
23		Week 4	0	0	0	0	0	0	0	0		

was conceptually easier to understand. This anecdotal finding was substantiated by an experimental study in a different course, which is described in Section 3.

It is important to clarify that our emphasis in the presentation of F2 is primarily didactic. That is, we believe the alternative formulation (F2) can be beneficial for illustrating the principles of time–cost tradeoffs in project networks in a more straightforward fashion than the traditional formulation (F1). We are not purporting that F2 is a "better" mathematical formulation than F1. Although F2 has half the number of decision variables and tends to have fewer constraints than F1 for most problems in introductory management science textbooks, some project-planning applications with a high level of detail might have several hundred or even thousands of activities. Moreover, it is possible to construct examples of very *dense* networks in which most activities have a large number of immediate predecessors. In such cases, F2 could require far more constraints than F1. Nevertheless, there are many sparse networks in the literature in which the number of paths is either smaller or just slightly larger than the number of activities (see, for example, Mungle 2014). Finally, we acknowledge that special-purpose network algorithms are computationally preferable to linear programming for large-scale network analysis (Fulkerson 1961, Kelley 1961, Phillips and Dessouky 1977), but these are well beyond the scope of an introductory management science course.

Perhaps the most natural criticism of the F2 formulation is that it requires the generation of all of the paths through the network and the computation of their lengths. However, as noted previously, the number of paths in most textbook problems tends to be very small, and there are also sparse real-world networks in which the number of paths is quite manageable for the comppath macro. Moreover, the consideration of all possible paths is not unprecedented in textbook presentations of project network analysis. For example, Stevenson (2018) presents a heuristic method for project crashing that can be used under the assumption of either the continuous (linear)

Figure 11. Solver Dialog Box for the F2-D Formulation of Discrete Example



or discrete (nonlinear) cost structure. The heuristic requires the knowledge of all path lengths and crashes the project one time period at a time to obtain a solution, but a globally optimal solution to the problem is not guaranteed. Stevenson (2018) also discusses the use of the mean and variance of the completion time for all paths as a basis for computing the probability of completing the project within some time frame.

Appendix

In this appendix, we show the equivalence of the F1 and F2 formulations. The argument is provided via the construction of the F1 formulation from the F2 formulation. Because the y_j variables, the objective function (3), and the constraint sets (4) and (8) are identical for the two formulations, the crux of constructive argument is to show that constraint set (9) of the F2 formulation enforces the same conditions as constraint sets (5)–(7) of the F1 formulation.

We begin the argument by noting that the length of each path (L_i) is just the sum of the normal completion times for the activities on that path. Therefore, considering all paths in constraint set (9), not just those with $L_i > T$, can be rewritten as

$$\sum_{j \in S_i} y_j \geq \sum_{j \in S_i} \tau_{Nj} - T \quad \forall\, 1 \leq i \leq p. \tag{A.1}$$

Rearranging terms, this constraint set can also be conveniently rewritten as

$$\sum_{j \in S_i} (\tau_{Nj} - y_j) \leq T \quad \forall\, 1 \leq i \leq p. \tag{A.2}$$

Next, we introduce the x_j variables and their corresponding nonnegativity constraints (7). Consider an arbitrary path, i, and let f and l correspond, respectively, to the first and last activities in that path. Accordingly, we know that activity f has no predecessors, and activity l has no successors. Because f has no predecessors, we know that its earliest possible finish time is $x_f = \tau_{Nf} - y_f$. Therefore, we can add this equation (or, for consistency with activities that do have predecessors, its inequality variant $x_f \geq \tau_{Nf} - y_f$, which is a member of constraint set (6)) to the constraint set and reexpress the path constraint for path i as

$$x_f + \sum_{j \in S_i \setminus \{f\}} (\tau_{Nj} - y_j) \leq T \tag{A.3}$$

Suppose that q is the second activity in path i. We know that the early finish time for q must equal or exceed the early finish time of its immediate predecessor f plus its own activity time. Thus, we append the constraint $x_q \geq x_f + (\tau_{Nq} - y_q)$ as a member of constraint set (6). In this instance, the less restrictive "≥" condition is essential because there might be a path on which activity q has a different predecessor that has a later early finish time than activity f. The constraint for path i can now be reexpressed as

$$x_q + \sum_{j \in S_i \setminus \{f,q\}} (\tau_{Nj} - y_j) \leq T \tag{A.4}$$

By induction, we can introduce the early finish time variables and their corresponding precedence constraints until we get to the last activity (l) in path i, in which case the final reexpression of the path constraint is simply

$$x_l \leq T, \tag{A.5}$$

and this inequality is clearly a member of constraint set (5). If this process is repeated for all p paths, then all of the precedence constraints in (6) are appended to the formulation because all precedence relationships in the network are captured when enumerating all paths. Moreover, based on the result in (A.5), constraint set (9) will have morphed into constraint set (5), that is, a system of constraints to ensure that all activities with no successors (i.e., activities that are members of G) finish by time period t.

As a simple illustration, consider the example introduced in Section 2 (Table 1 and Figures 1 and 2). We focus on the path 1–4–7 and the assumption that $T = 20$. The constraint for this path associated with constraint set (9) of the F2 formulation can be written as $(10 - y_1) + (9 - y_4) + (5 - y_7) \leq 20$. We introduce the variable x_1 and its corresponding nonnegativity restriction in constraint set (7). Then, we can append the constraint $x_1 \geq 10 - y_1$ associated with constraint set (6) and rewrite the path constraint as $x_1 + (9 - y_4) + (5 - y_7) \leq 20$. Next, we add the variable x_4 and its corresponding nonnegativity restriction in constraint set (7) and append the constraint $x_4 \geq x_1 + (10 - y_1)$ associated with constraint set (6). The path constraint is now rewritten as $x_4 + (5 - y_7) \leq 20$. Finally, we add the variable x_7 and its corresponding nonnegativity restriction in constraint set (7) and append the constraint $x_7 \geq x_4 + (5 - y_7)$ associated with constraint set (6). The path constraint is now just $x_7 \leq 20$, which is one of the constraints in constraint set (5) of the F1 formulation. If we repeat this process for all of the remaining paths, we convert constraint set (9) into constraints (5)–(7), thus converting the F2 formulation to the F1 formulation.

References

Anderson DR, Sweeney DJ, Williams TA, Camm JD, Cochran JJ, Fry MJ, Ohlmann JW (2019) *An Introduction to Management Science*, 15th ed. (Cengage Learning, Boston).

Balakrishnan N, Render B, Stair RM Jr (2007) *Managerial Decision Modeling with Spreadsheets*, 2nd ed. (Pearson, Upper Saddle River, NJ).

De P, Dunne EJ, Ghosh JB, Wells CE (1995) The discrete time-cost tradeoff problem revisited. *Eur. J. Oper. Res.* 81(2):225–238.

Eppen GD, Gould FJ, Schmidt CP (1993) *Introductory Management Science*, 4th ed. (Prentice Hall, Englewood Cliffs, NJ).

Fortier G (2006) The application of "crashing" a project network to solve the time/cost tradeoff in recapitalization of the UH-60A helicopter. Unpublished masters thesis, University of Central Florida, Orlando.

Fulkerson DR (1961) A network flow computation for project cost curve. *Management Sci.* 7(2):167–178.

Hillier FS, Hillier MS (2003) *Introduction to Management Science*, 2nd ed. (McGraw-Hill Irwin, New York).

Kelley JE Jr (1961) Critical-path planning and scheduling: Mathematical basis. *Oper. Res.* 9(3):296–320.

Krogstad JL, Grudnitski G, Bryant DW (1977) PERT and PERT/cost for audit planning and control. *J. Accountancy* 144(5):82–91.

Moscove SA, Simkin MG (1987) *Accounting Information Systems*, 3rd ed. (Wiley, New York).

Mungle S (2014) A portfolio approach to algorithm selection for discrete time-cost trade-off problem. Preprint, submitted December 5, https://arxiv.org/abs/1412.1913.

Murdick RG, Render B, Russell RS (1990) *Service Operations Management* (Allyn and Bacon, Needham Heights, MA).

Phillips S Jr, Dessouky MI (1977) Solving the project time/cost tradeoff problem using the minimal cut concept. *Management Sci.* 24(4):393–400.

Ragsdale CT (2003) A new approach to implementing project networks in spreadsheets. *INFORMS Trans. Ed.* 3(3):76–85.

Render B, Stair RM Jr, Hanna ME (2009) *Quantitative Analysis for Management*, 10th ed. (Pearson, Upper Saddle River, NJ).

Seal KC (2001) A generalized PERT/CPM implementation in a spreadsheet. *INFORMS Trans. Ed.* 2(1):16–26.

Stevenson WJ (2018) *Operations Management*, 13th ed. (McGraw-Hill, New York).

Taylor BW III (2019) *Introduction to Management Science*, 13th ed. (Pearson, Upper Saddle River, NJ).

Winston WL (1994) *Operations Research*, 3rd ed. (Wadsworth Publishing Company, Belmont, CA).