

Crashing the Schedule – An Algorithmic Approach with Caveats and Comments

Gilbert C. Brunnhoeffler, III PhD, P.E. and B. Gokhan Celik PhD LEED AP
Roger Williams University
Bristol, Rhode Island and Providence Plantations

The calculation of the minimum duration of a schedule (crashing the schedule) involves making value judgments about the best order for reducing individual tasks. Every time the baseline schedule is modified, the characteristics of the critical path(s) are altered. Keeping track of the new schedule, altered critical path(s), and potential results of the reduction of duration of the next task to be altered requires a systematic, tabular approach that is automatically updated as the schedule is altered. This paper presents an algorithmic approach to crashing the schedule that is applicable to large, realistic project schedules.

Key Words: algorithm, schedule crashing, maximum cost, minimum time

Introduction

The concept of reducing the duration of a baseline schedule for a project is often introduced to the construction management undergraduate students as a process of reducing the duration of selected tasks located on the critical path incrementally while simultaneously increasing the direct costs of the task being reduced. The new “total time to complete” and “direct cost to complete” values are then calculated. The overhead cost value applied to the project is calculated in light of the reduced project schedule duration and added to the new direct cost value to obtain the new total project cost value. The data points representing the collection of each incremental change in the project duration and the associated project costs are plotted and the result usually shows a declining total cost (direct + indirect cost) until the incremental direct cost overwhelms the incremental decrease in indirect costs. The minimum value of the total cost of the project versus the time to complete is then established and accepted as the optimum duration and cost to complete the project. This traditional presentation of the schedule crashing has several caveats; 1) the fundamental assumption applied to crashing a task assumes that the task duration can be decremented in small units such as days, 2) the assumption that the overhead charges applied to the job can be decreased if the job is accomplished faster than the original baseline duration which implies that the overhead charges will be applied to other company projects, and 3) the lowest cost to complete the project can be determined by reducing the time of the tasks lying on the critical path in reverse order of their incremental costs and this reduction will not change the logic of the schedule. Aside from the questionable assumptions associated with the cost analysis of the reduction in project duration, the process of crashing the schedule requires an algorithmic approach if it is to be applied to realistic schedules of more than thirty tasks. This paper presents an algorithmic process to reduce baseline project duration, and to calculate the associated costs.

The algorithmic process described in this paper can be summarized as shown below:

- Step 1 - Determine all possible paths through the CPM Diagram
- Step 2 - Determine the duration of all possible paths through the CPM Diagram
- Step 3 - Compute the cost to crash each task (or combination of tasks) on the identified critical path(s) - if no task (or combination of tasks) on the critical path(s) can be "crashed" go to Step 9

Step 4 - Select the lowest cost alternative to shortening the overall project schedule
Step 5 - Reduce the duration of the selected task(s)
Step 6 - Compute the new direct project costs and the new duration of the schedule
Step 7 - Identify the new critical path(s)
Step 8 - If a task (or combination of tasks) on the critical path(s) can be crashed, go to Step 3
Step 9 - Record all durations and corresponding project direct costs, compute total project costs, plot direct costs, indirect costs, and total costs v. durations

Schedule Set-up in Microsoft Excel

The information for the demonstration of crashing a critical path method (CPM) schedule in Microsoft Excel is shown in Table 1:

Table 1

Sample layout for CPM network data

	A	B	C	D	E	F	G
1	Task #	Task	IPA Immediately Preceding Activities	Normal Duration	Normal Direct Costs	Crash Duration	Crash Direct Costs
2	1	Task A	-	1 day	\$ 6,000	1 day	\$ 6,000
3	2	Task B	A	6.5 days	\$13,500	5 days	\$15,500
4	3	Task C	A,B	8 days	\$12,350	6.5 days	\$18,000
5	4	Task D	A,B	7 days	\$22,145	5 days	\$26,300
6	5	Task E	B,C	3 days	\$ 2,675	1.5 days	\$ 3,500
7	6	Task F	B,D	8.5 days	\$34,000	5.5 days	\$36,650
8	7	Task G	C,E	4 days	\$ 1,250	2.5 days	\$ 2,150
9	8	Task H	D,F	3 days	\$ 5,005	1 days	\$ 8,705
10	9	Task K	D,G,H	2 days	\$ 2,500	2 days	\$ 2,500

A quick look at a CPM schedule reveals the duration of the project displayed in Table 1 as twenty-eight (28) workdays and the sum of all the values in the normal cost column reveals the project direct cost to be \$99,425. Notice that Task A and Task K cannot be crashed. Further consultation of a CPM schedule reveals that the critical activities are Task A, Task B, Task D, Task F, Task H, and Task K. In order to crash the schedule, a useful technique will provide the user with an identity of the critical path(s), the cost to crash each activity on the critical path(s), and an updated estimate of the duration and direct costs of the project as the schedule is compressed. The scheduler then can select the lowest crash cost activity to compress the project schedule.

The first step in the proposed algorithm is to identify all possible paths through the CPM network. This allows the scheduler to track changes in the schedule and in the project direct costs as the schedule is compressed. In order to do this, one can start with listing the first task (Task A), consulting the IPA (Immediately Preceding Activity) column to locate the first task (Task A), and listing that task once for every time it is found in the IPA column. The task pairs, as shown in Table 2, should be listed each time the first task (Task A) appears in the IPA column.

Table 2

Setting up the path count in a CPM network in Microsoft Excel– step 1

	A	B	C	D	E	F	G	H
16	Duration	Path #	1 st task	2 nd task	3 rd task	4 th task	5 th task	6 th task
17		1	Task A	Task B				
18		2	Task A	Task C				
19		3	Task A	Task D				
20		4						
21		5						

Then proceed to evaluate the second task in the same manner as step 1. Consult the IPA column and list the task triplets for every time the second task (Task B) appears. Step 2 is shown in Table 3.

Table 3

Setting up the path determination in a CPM network – step 2

	A	B	C	D	E	F	G	H
16	Duration	Path #	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
17		1	Task A	Task B				
18		2	Task A	Task B				
19		3	Task A	Task B				
20		4	Task A	Task B				
21		5	Task A	Task C				
22		6	Task A	Task C				
23		7	Task A	Task D				
24		8	Task A	Task D				
25		9	Task A	Task D				
26								
27								
28								

Continue in this fashion until all tasks are accounted for and every path begins with the first task, Task A, and ends with the last task, Task K. As shown in Table 4, this results in a total path count of twelve (12). The next activity in the algorithm is to compute the duration of each path.

Table 4

Setting up the path determination in a CPM network – iteration 12

	A	B	C	D	E	F	G	H
16	Path Duration	Path #	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
17		1	Task A	Task B	Task C	Task E	Task G	Task K
18		2	Task A	Task B	Task C	Task G	Task K	
19		3	Task A	Task B	Task D	Task F	Task H	Task K
20		4	Task A	Task B	Task D	Task H	Task K	
21		5	Task A	Task B	Task D	Task K		
22		6	Task A	Task B	Task E	Task G	Task K	
23		7	Task A	Task B	Task F	Task H	Task K	

24	8	Task A	Task C	Task E	Task G	Task K
25	9	Task A	Task C	Task G	Task K	
26	10	Task A	Task D	Task F	Task H	Task K
27	11	Task A	Task D	Task H	Task K	
28	12	Task A	Task D	Task K		

There are three reasons to calculate the duration length of each path; 1) to determine the critical path(s), 2) to determine the critical path length (duration) and 3) to enable tracking the critical path(s) and path length(s) (duration(s)) as the schedule is compressed. To calculate the path duration(s) in Microsoft Excel one can use a combination of an IF statement and a LOOKUP statement. Select the path containing the largest number of tasks. Check to see if the first cell (1st task) is empty using the IF statement. If the cell is not empty, find its content in the list of tasks and return the value of its duration using the LOOKUP statement. For the duration value of the first task, the Microsoft Excel entry in the Path Duration cell for Path 1, cell A17 is:
=IF(C17="",0,LOOKUP(C17,\$B\$2:\$B\$10,\$D\$2:\$D\$10)). This formula is keyed to the rows and columns shown in Table 5.

Table 5

Microsoft Excel array for path duration determination

	A	B	C	D	E	F	G	H
1	Task #	Task	IPA Immediately Preceding Activities	Normal Duration	Normal Direct Costs	Crash Duration	Crash Direct Costs	
2	1	Task A	-	1 day	\$ 6,000	1 day	\$ 6,000	
3	2	Task B	A	6.5 days	\$13,500	5 days	\$15,500	
4	3	Task C	A,B	8 days	\$12,350	6.5 days	\$18,000	
5	4	Task D	A,B	7 days	\$22,145	5 days	\$26,300	
6	5	Task E	B,C	3 days	\$ 2,675	1.5 days	\$ 3,500	
7	6	Task F	B,D	8.5 days	\$34,000	5.5 days	\$36,650	
8	7	Task G	C,E	4 days	\$ 1,250	2.5 days	\$ 2,150	
9	8	Task H	D,F	3 days	\$ 5,005	1 day	\$ 8,705	
10	9	Task K	D,G,H	2 days	\$ 2,500	2 days	\$ 2,500	
11				Project Cost	\$99,425			
...								
16	Path Duration	Path #	1 st task	2 nd task	3 rd task	4 th task	5 th task	6 th task
17		1	Task A	Task B	Task C	Task E	Task G	Task K
18		2	Task A	Task B	Task C	Task G	Task K	
19		3	Task A	Task B	Task D	Task F	Task H	Task K
20		4	Task A	Task B	Task D	Task H	Task K	
21		5	Task A	Task B	Task D	Task K		
22		6	Task A	Task B	Task E	Task G	Task K	
23		7	Task A	Task B	Task F	Task H	Task K	
24		8	Task A	Task C	Task E	Task G	Task K	
25		9	Task A	Task C	Task G	Task K		
26		10	Task A	Task D	Task F	Task H	Task K	
27		11	Task A	Task D	Task H	Task K		
28		12	Task A	Task D	Task K			

The IF-LOOKUP statement given above can be used as a building block to search each cell in the longest path and return the total duration of that path. The result for cell A17 is:

=IF(C17="",0,LOOKUP(C17,\$B\$2:\$B\$9,\$D\$2:\$D\$10))+IF(D17="",0,LOOKUP(D17,\$B\$2:\$B\$9,\$D\$2:\$D\$10))+
IF(E17="",0,LOOKUP(E17,\$B\$2:\$B\$9,\$D\$2:\$D\$10))+IF(F17="",0,LOOKUP(F17,\$B\$2:\$B\$9,\$D\$2:\$D\$10))+
IF(G17="",0,LOOKUP(G17,\$B\$2:\$B\$9,\$D\$2:\$D\$10))+IF(H17="",0,LOOKUP(H17,\$B\$2:\$B\$9,\$D\$2:\$D\$10))

This formula can be copied to all the Path Duration cells (A17 through A28) to calculate the path duration for each of the twelve (12) paths through the CPM network. The result is shown in Table 6.

Table 6

Calculation of all path durations in the CPM network in Microsoft Excel

	A	B	C	D	E	F	G	H
1	Task #	Task	IPA Immediately Preceding Activities	Normal Duration	Normal Direct Costs	Crash Duration	Crash Direct Costs	Crash Cost
2	1	Task A	-	1 day	\$ 6,000	1 day	\$ 6,000	-
3	2	Task B	A	6.5 days	\$13,500	5 days	\$15,500	\$2,000
4	3	Task C	A,B	8 days	\$12,350	6.5 days	\$18,000	\$5,650
5	4	Task D	A,B	7 days	\$22,145	5 days	\$26,300	\$4,155
6	5	Task E	B,C	3 days	\$ 2,675	1.5 days	\$ 3,500	\$825
7	6	Task F	B,D	8.5 days	\$34,000	5.5 days	\$36,650	\$2,650
8	7	Task G	C,E	4 days	\$ 1,250	2.5 days	\$ 2,150	\$900
9	8	Task H	D,F	3 days	\$ 5,005	1 day	\$ 8,705	\$3,700
10	9	Task K	D,G,H	2 days	\$ 2,500	2 days	\$ 2,500	-
11				Project Cost	\$99,425			
...								
16	Path Duration	Path #	1 st task	2 nd task	3 rd task	4 th task	5 th task	6 th task
17	24.5 days	1	Task A	Task B	Task C	Task E	Task G	Task K
18	21.5 days	2	Task A	Task B	Task C	Task G	Task K	
19	28 days	3	Task A	Task B	Task D	Task F	Task H	Task K
20	19.5 days	4	Task A	Task B	Task D	Task H	Task K	
21	16.5 days	5	Task A	Task B	Task D	Task K		
22	16.5 days	6	Task A	Task B	Task E	Task G	Task K	
23	21 days	7	Task A	Task B	Task F	Task H	Task K	
24	18 days	8	Task A	Task C	Task E	Task G	Task K	
25	15 days	9	Task A	Task C	Task G	Task K		
26	21.5 days	10	Task A	Task D	Task F	Task H	Task K	
27	13 days	11	Task A	Task D	Task H	Task K		
28	10 days	12	Task A	Task D	Task K			
29	28 days	Max						
30	Critical Path	3	Task A	Task B	Task D	Task F	Task H	Task K
31	Cost to Crash			\$2,000	\$4,155	\$2,650	\$3,700	

The duration of the critical path is determined in cell A29 using the MAX function. The formula in cell A29 is =MAX(B17:B28). The contents of cells B30 through H30 can be entered automatically by using an IF function to compare the contents of cells A17 through A28 with the value determined in cell A29 (maximum path duration). If there is a match, the contents of the cells containing the tasks in the path can be entered in cells B30 through H30. The reason for listing the tasks contained in the critical path is to use them to look up the cost to crash the schedule. At this step, one must decide if the schedule is going to be compressed by reducing the critical tasks in increments less than the full value of the difference between the normal and the crash durations or by the full interval that the

task can be crashed. This example is for compressing the schedule by reducing each task a single time by the full value of the normal duration less the crash duration. If one wishes to compress the tasks by shorter intervals, the same techniques presented here can be used but care must be taken to insure that the appropriate increases in direct costs are proportioned to the interval used to reduce the task durations.

The crash cost shown in column H of Table 6 was computed by subtracting the value of the normal cost (column E) from the value of the crash cost (column G) if the value of the normal duration (column D) is larger than the value of the crash duration (column F). Table 7 shows the result of using the crash duration of the least crash direct cost among the critical tasks. The procedure is to crash each task, check to see if the critical path has changed, and then crash the next task on the critical path with the lowest crash cost. For Task B, Task F, and Task H, crashing them, in that order, does not change the critical path (Path #3) and produces the following results shown in Table 7.

Table 7

Schedule and costs resulting from crashing Task B, Task F, and Task H, in order

Project Duration	Project Direct Costs
28 days	\$ 99,425
26.5 days	\$101,425
23.5 days	\$104,075
23 days	\$107,775

After crashing Task B, Task F, and Task H, the critical path shifts to Path #1, as shown in Table 8.

Table 8

Identifying the order of tasks to reduce the overall project duration by lowest cost using the Microsoft Excel spreadsheet – identify Task E to be compressed to 1.5 days at a cost of \$825

	A	B	C	D	E	F	G	H
1	Task #	Task	IPA Immediately Preceding Activities	Normal Duration	Normal Direct Costs	Crash Duration	Crash Direct Costs	Crash Cost
2	1	Task A	-	1 day	\$ 6,000	1 day	\$ 6,000	-
3	2	Task B	A	5 days	\$13,500	5 days	\$15,500	-
4	3	Task C	A,B	8 days	\$12,350	6.5 days	\$18,000	\$5,650
5	4	Task D	A,B	7 days	\$22,145	5 days	\$26,300	\$4,155
6	5	Task E	B,C	3 days	\$ 2,675	1.5 days	\$ 3,500	\$825
7	6	Task F	B,D	5.5 days	\$34,000	5.5 days	\$36,650	-
8	7	Task G	C,E	4 days	\$ 1,250	2.5 days	\$ 2,150	\$900
9	8	Task H	D,F	3 days	\$ 5,005	1 days	\$ 8,705	-
10	9	Task K	D,G,H	2 days	\$ 2,500	2 days	\$ 2,500	-
11				Project Cost	\$107,945			
...								
16	Path Duration	Path #	1 st task	2 nd task	3 rd task	4 th task	5 th task	6 th task
17	23 days	1	Task A	Task B	Task C	Task E	Task G	Task K
18	20 days	2	Task A	Task B	Task C	Task G	Task K	
19	21.5 days	3	Task A	Task B	Task D	Task F	Task H	Task K
20	16 days	4	Task A	Task B	Task D	Task H	Task K	

21	15 days	5	Task A	Task B	Task D	Task K		
22	15 days	6	Task A	Task B	Task E	Task G	Task K	
23	14.5 days	7	Task A	Task B	Task F	Task H	Task K	
24	18 days	8	Task A	Task C	Task E	Task G	Task K	
25	15 days	9	Task A	Task C	Task G	Task K		
26	16.5 days	10	Task A	Task D	Task F	Task H	Task K	
27	11 days	11	Task A	Task D	Task H	Task K		
28	10 days	12	Task A	Task D	Task K			
29	23 days	Max						
30	Critical Path	1	Task A	Task B	Task C	Task E	Task G	Task K
31	Cost to Crash				\$5,650	\$825	\$900	

The result of reducing Task E to 1.5 days is a reduction in the project duration to 21.5 days and an increase in the direct costs by \$825 to \$108,600. Making this cut in Task E, shifts the critical path to Path #1 and Path #3, as shown in Table 9.

Table 9

Identifying the order of tasks to reduce the overall project duration by lowest cost using the Microsoft Excel spreadsheet – identify Task E to be compressed to 1.5 days at a cost of \$825

	A	B	C	D	E	F	G	H
1	Task #	Task	IPA Immediately Preceding Activities	Normal Duration	Normal Direct Costs	Crash Duration	Crash Direct Costs	Crash Cost
2	1	Task A	-	1 day	\$ 6,000	1 day	\$ 6,000	-
3	2	Task B	A	5 days	\$13,500	5 days	\$15,500	-
4	3	Task C	A,B	8 days	\$12,350	6.5 days	\$18,000	\$5,650
5	4	Task D	A,B	7 days	\$22,145	5 days	\$26,300	\$4,155
6	5	Task E	B,C	3 days	\$ 2,675	1.5 days	\$ 3,500	-
7	6	Task F	B,D	5.5 days	\$34,000	5.5 days	\$36,650	-
8	7	Task G	C,E	4 days	\$ 1,250	2.5 days	\$ 2,150	\$900
9	8	Task H	D,F	3 days	\$ 5,005	1 days	\$ 8,705	-
10	9	Task K	D,G,H	2 days	\$ 2,500	2 days	\$ 2,500	-
11				Project Cost	\$108,600			
...								
16	Path Duration	Path #	1 st task	2 nd task	3 rd task	4 th task	5 th task	6 th task
17	21.5 days	1	Task A	Task B	Task C	Task E	Task G	Task K
18	20 days	2	Task A	Task B	Task C	Task G	Task K	
19	21.5 days	3	Task A	Task B	Task D	Task F	Task H	Task K
20	16 days	4	Task A	Task B	Task D	Task H	Task K	
21	15 days	5	Task A	Task B	Task D	Task K		
22	13.5 days	6	Task A	Task B	Task E	Task G	Task K	
23	14.5 days	7	Task A	Task B	Task F	Task H	Task K	
24	16.5 days	8	Task A	Task C	Task E	Task G	Task K	
25	15 days	9	Task A	Task C	Task G	Task K		
26	16.5 days	10	Task A	Task D	Task F	Task H	Task K	
27	11 days	11	Task A	Task D	Task H	Task K		
28	10 days	12	Task A	Task D	Task K			

29	21.5 days	Max						
30	Critical Path	1	Task A	Task B	Task C	Task E	Task G	Task K
31	Cost to Crash				\$5,650		\$900	
32	Critical Path	3	Task A	Task B	Task D	Task F	Task H	Task K
33	Cost to Crash				\$4,155			

Inspecting Table 9 reveals that there are two critical paths (Path #1 and Path #3) and no task that is common to the two paths. This state requires the scheduler to investigate the price of all combinations of tasks that include at least one task from each critical path. Task C and Task D cost \$9,805, if reduced together, and Task G and Task D cost \$5,055, if reduced together. The choice is to reduce Task G and Task D to 2.5 days and 5 days, respectively. Doing this increases the project direct costs by \$5,055 to \$113,655 and reduces the project duration to 20 days. This action returns the critical path to a single path, Path #1, with only Task C remaining to reduce at a cost of \$5,650. Reducing Task C to 6.5 days at a cost of \$5,650, reduces the project time to 19.5 days, increases the project direct costs to \$119,305, and makes Path #3 the only critical path with no available task to reduce the overall project schedule. The final tally of cost v. duration is shown in Table 10.

Table 10

Final results of schedule crashing exercise

Duration	Project Direct Cost	Task Changed	Incremental Cost	Task Time Reduction
28.0 days	\$ 99,425			
26.5 days	\$101,425	Task B	\$2,000	1.5 days
23.5 days	\$104,075	Task F	\$2,650	3 days
23.0 days	\$107,775	Task H	\$3,700	2 days
21.5 days	\$108,600	Task E	\$ 825	1.5 days
20.0 days	\$113,655	Task G & Task D	\$5,055	1.5 days & 2 days
19.5 days	\$119,305	Task C	\$5,650	1.5 days

Conclusion

This algorithmic approach for calculating the reduction in project duration and associated increase in project direct cost lends itself to solving schedules with a large number of tasks through the use of Excel macros. Macros, combined with Excel Solver Add-In, can be coded to find the cheapest combination of tasks to crash when there is more than a single critical path. This would allow a computed and more reliable decision making process which would eliminate the human error that can occur while the number of critical paths increases. Another useful addition would be a way to determine, *a priori*, how many paths are possible through a specific CPM network. Basic programming languages can be used to convert the entire Microsoft Excel process described in this paper into an executable software program. The authors are currently working on these projects and would welcome any input and advice about the algorithmic approach presented here.