

A Linear Programming Approach for the Project Controlling

¹Wakas S. Khalaf and ²Leong Wah June

¹Institute of Mathematical Research, ²Department of Mathematical,
Faculty of Science, Putra Malaysia University, 43400 Serdang, Selangor, Malaysia

Abstract: One of the most challenging jobs that any manager can take on is the management of a large-scale project that requires coordinating numerous activities throughout the organization. A myriad of details must be considered in planning how to coordinate all these activities, in developing a realistic schedule and then in monitoring the progress of the project. The main aim of this study is on using Linear Programming (LP) technique to build two models. The first is to find the duration of project completion (critical path or Longest route of a unit flow entering at the start node and terminating at the finish node). The second model is built because sometimes it is required to complete a project within the predetermined deadline to keep cost at lowest possible level. Failure to do so ultimately leads to increase in total cost. This would let managers to encounter a decision situation, which activities of the project will be crashed to minimize the total cost of crashing. Thus, the second model is minimizing the cost of crashing the project's activities to meet the desired project completion time. We use a software package (Win.QSB) to deal with all the data needed to develop schedule information and then to monitor the progress of the project. Finally, we provide a combined report of the problem via analysis the results that obtained from solving these two models to give us some flexibility in planning, scheduling and controlling.

Key words: Linear programming, CPM, time-cost trade-off, least cost-scheduling, cost optimization, project compression

INTRODUCTION

Construction management involves the coordination of group of activities where in the manager plans, organizes, staffs, directs and controls construction projects to achieve an object, including a specification of their interrelationships and considering the required resources in an acceptable time span. The success of CPM is that it utilizes the planner's knowledge, experience and instincts in a logical way first to plan and then to schedule. CPM can save money through better planning.

Construction managers are continually facing a situation in which they must take a decision whether to complete the project sooner than originally specified in the contract because of the clients request and/or to optimize the cost of expediting.

The objective of Critical Path Method (CPM) is to establish a feasible and desirable relationship between the time and cost of the project by reducing the target time and taking into account the cost of expediting. Number scheduling methods were developed for planning and scheduling of construction projects using graphical

methods such as line of balance and vertical production method. These techniques are neither suitable for the scheduling of linear projects nor adequate for addressing typical challenge related to time-cost trade-off.

Fortunately, two closely related operations research techniques, LP (Linear Programming) and CPM (Critical Path Method) are available to assist the project manager in carrying out these responsibilities. These techniques make heavy use of networks to help plan and display the coordination of all the activities. We use a software package (Win.QSB) to deal with all the data needed to develop schedule information and then to monitor the progress of the project (Lawrence and Pasternack, 1999; Hillier and Liberman, 2001; O'Brien and Plotnick, 2006).

The study mainly focuses on problems faced by the management when dealing with different types of problems, approaches to deal with projects. LP (Linear Programming) is used for developing two models. The first is to determine the critical activities then find a critical path (longest path in the project network). The second model is minimizing the cost of crashing the project's activities to complete project in the desired time.

MATERIALS AND METHODS

Linear programming technique to find the critical path for the project network: We know linear programming is a tool for decision making under certain situation. So, the basic assumption of this approach is that we have to know some relevant data with certainty. We are interested in finding the critical path or the longest route of a unit flows entering at the start node and terminating at the finish node.

The following terms need to illustrate:

- A_i = Project's activities, where $i = (1, 2, \dots, n)$
- $T_{N,i}$ = Normal Time for activity i , which is usually the time to complete the activity with minimal resource
- Y_i = The decision variable for the start time of activity i

Thus, the objective function of linear programming becomes:

$$\text{Maximize } (Z) = \sum_{i=1}^n T_{N,i} \cdot Y_i \quad (1)$$

This objective function is subject to some constraints. These constraints can be classified in to three categories.

- For activities that entered node i

$$\sum_{i=1}^n Y_i = 1 \quad (2)$$

- For activities that leaves node i

$$\sum_{i=1}^n Y_i = -1 \quad (3)$$

- For activities that entered and leaves

Node i : For each node, there is one constraint that represents the conservation of flow: total input flow = Total output flow.

In this formulation, the $Y_i = 0$ or 1 denotes the absence or presence of unite flow from node to another. So the (Eq. 4) is:

$$\sum_{i=1}^n +Y - Y_i = 0 \quad (4)$$

- Nonnegative constraints

All the decision variable $Y_i \geq 0$

Linear programming technique to meet the desired project completion time: The basic data requirements are as follows:

- The crash cost associated with per unit of time for all activities

To reduce the time to complete the activity, more resources are applied in the form of additional personnel and overtime. As more resources are applied, the duration is shortened, but the cost rises. When the maximum effort is applied so that the activity can be completed in the shortest possible time. The equation for the cost slope is

$$\text{Cost slope} = \frac{C_c - C_n}{T_n - T_c} \quad (5)$$

Where,

- C_c and C_n = The crash and normal costs, respectively
- T_c and T_n = The crash and normal times for the same activity

The cost slope shows by how much the cost of the job would change if activities were sped up or slowed down. In general, the steepness of the cost slope increases as an activity is accelerated (Senouci and Eldin, 1996; Islam Mohammad Nazmul *et al.*, 2004; Nicholas, 2004).

- We have to know the project network with activity time, which can be achieved from CPM
- To what extent an activity can be crashed

Before formulating the model, let us define some relevant terms. We know, a project is the combination of some activities, which are interrelated in a logical sequence in the sense that the starting of some activities is dependent upon the completion of some other activities. These activities are jobs, which require time and resources to be completed. The relationship between the activities is specified by using event. As an event represents a point in time that implies the completion of some activities and the beginning of new ones, the beginning and end point of an activity are thus, expressed by two events.

Now let's define the variable of the problem.

- Y_i = The time when an event i will occur, measured since the beginning of the project, where $i = (1, 2, 3, \dots, n)$
- X_q = Amount of times (measured in terms of days, weeks, months or some other units) that each activity q will be crashed, where $q = (1, 2, 3 \dots L)$
- U_q = (Cost slope) Crash cost per unit of time for activity q

The objective is to minimize the cost of crashing total project via minimize the durations of crashing activities that multiplied by their associated costs slope, then adding the resultant cost to the normal cost of project completion. The LP objective function will be:

$$\text{Minimize } Z = \sum_{q=1}^L U_q \times X_q \quad (6)$$

This objective function is subject to some constraints. These constraints can be classified in to three categories.

Crash time constraints: We can reduce the time to complete an activity by simply increasing the resources or by improving the productivity, which also requires the commitment of additional resources. But, it is not possible to reduce the required time to complete an activity after a certain threshold limit. Strive for such intention will result in superfluous resources employment which will be an inefficient approach. That is why, the allowable time to crash an activity has a limit.

Constraints unfolding the network: These set of constraints describe the structure of the network. As we mention earlier that the activities of a project are interrelated, the starting of some activities is dependent upon the completion of some other activities; we must have to establish research sequence of the activities through constraints.

Project completion constraints: This constraint will recognize that the last event (completion of last activities) must take place before the project deadline date.

Nonnegative constraints: All decision variable must ≥ 0 . So, the constraints are:

Crash time constraints: $X_q \leq$ Allowable crashing time for activity q measured in terms of days, weeks, months or some other units.

Constraints unfolding the network: There will be one or more constraints for each event depending on the predecessor activities of that event.

As the event 1 will start at the beginning of the project, we begin by setting the occurrence time for event 1 equals to zero. Thus, $Y_1 = 0$.

The other events will be expressed as follows: start time of this activity (Y_i) = (start time + normal duration - crash duration) for this immediate predecessor.

Project completion constraints: $Y_m \leq$ project deadline after being stretched, where m indicate the last event of that project (Elmaghraby and Pulat, 1997; Natarajan *et al.*, 2005) (Fig. 1).

Hypothetical project: The construction projects Co. decided to accept the offer to construct a new plant. The maximum budgeting that available for the client is \$B, but he desire to complete the plant in F time unit. It is useful at this point to illustrate the procedure with a numerical example with hypothetical data in which there are 23 activities. Table 1 summarizes a project with hypothetical data for the example.

The following terms need to illustrate.

- B = Maximum budgeting available
- A_i = Project's activities, where $i = (1, 2, \dots, n)$
- F = The desired project completion time
- $T.C_a$ = Total cost to complete the project in normal condition
- $T.C_N$ = Total cost to complete the project by crashing all activities (crash condition)

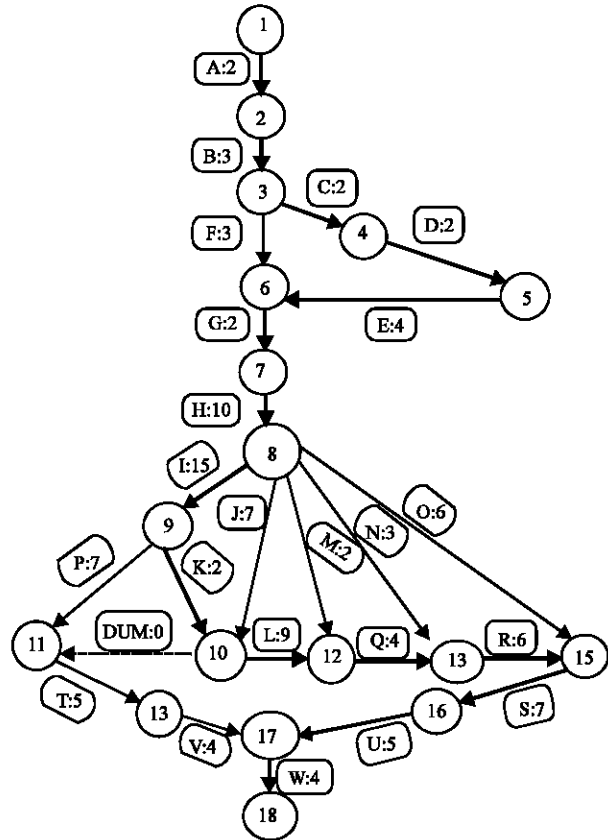


Fig. 1: Project network shows the critical path (bold arrows) in normal conditions critical path

Table 1: Activity data in normal and crash conditions

Activity code	Activity predecessor	Normal time T_n	Crash times T_c	Normal cost C_n	Crash cost C_c	Max reduction in time	Cost slope
A	-	2	1	20000	30000	1 week	10000
B	A	3	1	60000	100000	2 week	20000
C	B	2	1	30000	40000	1 week	10000
D	C	2	1	20000	30000	1 week	10000
E	D	4	2	100000	150000	2 week	25000
F	B	3	2	150000	180000	1 week	30000
G	E, F	2	1	200000	300000	1 week	100000
H	G	10	7	500000	620000	3 week	40000
I	H	15	10	650000	850000	5 week	40000
J	H	7	5	250000	300000	2 week	25000
K	I	2	1	20000	25000	1 week	5000
L	J, K	9	6	300000	420000	3 week	40000
M	H	2	1	20000	25000	1 week	5000
N	H	3	2	30000	40000	1 week	10000
O	H	6	4	120000	150000	2 week	15000
P	I	7	4	450000	570000	3 week	40000
Q	L, M	4	2	350000	500000	2 week	75000
R	O, Q	6	3	550000	760000	3 week	70000
S	N, R	7	5	450000	600000	2 week	75000
T	J, P	5	3	350000	450000	2 week	50000
U	S	5	3	250000	320000	2 week	35000
V	T	4	2	150000	220000	2 week	35000
W	U, V	4	2	100000	150000	2 week	25000
Total cost	-	-	-	5,120,000	6,830,000	-	-

Where, $A_i = (A, B, C, \dots, W)$, $B = \$6,830,000$, $F = 50$, $T.C_N = 5,120,000$, $T.C_a = 6,830,000$.

The project manager decides to use the linear programming technique to build two models, the first is to determine the critical activities then find duration of project completion (critical path or longest path in the project network).

The management looks forward to the challenge of bringing the project in on schedule. However, since it is a doubts that it will be feasible to finish the project within F time unit in normal work conditions, the cost and time of completing the project activities under normal conditions is $\$C_n$, T_n time unit respectively. So, the aim of second model is minimizing the cost of crashing the project's activities to meet the desired project completion time.

Model 1. To determine the project completion in normal conditions: We can find the objective function of linear programming technique:

$$2A + 3B + 2C + 2D + 4E + 3F + 2G + 10H + 15I + 7J + 2M + 3N + 6O + 7P + 2K + 9L + 4Q + 6R + 0DUM + 5T + 4V + 5U + 4W \quad (7)$$

S. to

- For activities that entered node (i):

$$\text{node1 } -A = -1 \quad (8)$$

- For activities that leaves node (i):

$$\text{node18 } W = 1 \quad (9)$$

- For activities that entered and leaves node i:

$$\text{node2 } A - B = 0 \quad (10)$$

$$\text{node3 } B - C - F = 0 \quad (11)$$

$$\text{node4 } C - D = 0 \quad (12)$$

$$\text{node5 } D - E = 0 \quad (13)$$

$$\text{node6 } F + E - G = 0 \quad (14)$$

$$\text{node7 } G - H = 0 \quad (15)$$

$$\text{node8 } H - I - J - M - N - O = 0 \quad (16)$$

$$\text{node9 } I - P - K = 0 \quad (17)$$

$$\text{node10 } J + K - L - DUM = 0 \quad (18)$$

$$\text{node11 } P + DUM - T = 0 \quad (19)$$

$$\text{node12 } M + L - Q = 0 \quad (20)$$

$$\text{node13 } T - V = 0 \quad (21)$$

$$\text{node14 } N + Q - R = 0 \quad (22)$$

$$\text{node15 } O + R - S = 0 \quad (23)$$

$$\text{node16 } S - U = 0 \quad (24)$$

$$\text{node17 } V + U - W = 0 \quad (25)$$

$$\text{node18 } W = 1 \quad (26)$$

Nonnegative constraints: All the decision variable $Y_i \geq 0$ i.e., A, B, ..., $W \geq 0$

RESULTS AND DISCUSSION

Solution summary table (Model 1): Solution of the model is presented in Table 2, which shows the solution of the problem. It includes the decision variable value, contribution to the objective and reduced cost of each decision variable. This also indicates the status of whether the decision variable is on the final basis. This table is available when the optimal solution is achieved.

Solution summary in Table 2 specially column 3 (Solution value) or column 7 (Basis Status) indicates that an activities A, B, C, D, E, G, H, I, K, L, R, S, U, W are critical activities and others are non. Also the result indicates that activities A, B, C, D, E, G, H, I, K, L, R, S, U, W should be completed in 2, 3, 2, 2, 4, 2, 10, 4, 2, 9, 4, 6, 7, 5, 15, respectively, so the duration to complete the project in normal conditions is 77 weeks. This table contains some important columns:

The reduced cost: The reduced cost of the non-basic variables (the variables whose value is zero in the optimum solution) provide us the information about how much objective coefficients of these variables should be increased to have a positive value of those variables in the optimum solution.

In the example, reduced cost of a current non-basic variable F is -5. It means the current coefficient of this variable which is now 3 must increased 5. That means the coefficient would be 8 or up to get a basic value of this variable in the optimum solution (Table 2 Column 6) (Lawrence and Pasternack, 1999).

Sensitivity analysis for OBJ: This analysis shows the ranges of the objective function coefficients such that the current basis holds. For each decision variable, this includes the lower limit and upper limit allowed for its objective function coefficient so that the variable stays in basis, i.e., a basic variable. This is also called the range of optimality. This Analysis is available when the optimal solution is achieved.

In the hypothetical example, the final value of variable A in the objective function is 1. The current coefficient of the variable is 2, allowable Max. c(j) (Table 2 Column 9) is M (infinity) and allowable Min. c(j) (Table 2 Column 8) is -M (infinity negative). It indicates our current solution would remain optimum whatever were the increasing or decreasing in normal duration for activity A. While, the current coefficient of the variable F is 3, allowable Min. c(j) is -M (infinity negative) and allowable Max. c(j) is 8. It indicates our current solution would remain optimum if normal duration for activity F varies from -M to 8.

Constraint summary table: This table shows the constraint status of the problem for the final solution. It includes the left-hand side, right-hand side, surplus or slack and shadow price of each constraint. This also

Table 2: Solution summary by using Win.QSB program

No. of variable	Decision variable	Solution value	Unit cost or profit c(j)	Total contribution	Reduced cost	Basis status	Allowable Min. c(j)	Allowable Max. c(j)
1	A	1	2	2	0	Basic	-M	M
2	B	1	3	3	0	Basic	-M	M
3	C	1	2	2	0	Basic	-3	M
4	D	1	2	2	0	Basic	-3	M
5	E	1	4	4	0	Basic	-1	M
6	F	0	3	0	-5	At bound	-M	8
7	G	1	2	2	0	Basic	-M	M
8	H	1	10	10	0	Basic	-M	M
9	I	1	15	15	0	Basic	5	M
10	J	0	7	0	-10	At bound	-M	17
11	M	0	2	0	-24	At bound	-M	26
12	N	0	3	0	-27	At bound	-M	30
13	O	0	6	0	-30	At bound	-M	36
14	P	0	7	0	-17	At bound	-M	24
15	K	1	2	2	0	Basic	-8	M
16	L	1	9	9	0	Basic	-8	M
17	Q	1	4	4	0	Basic	-13	M
18	R	1	6	6	0	Basic	-11	M
19	DUM	0	0	0	-22	At bound	-M	22
20	T	0	5	0	0	Basic	-M	22
21	V	0	4	0	0	Basic	-M	21
22	S	1	7	7	0	Basic	-10	M
23	U	1	5	5	0	Basic	-12	M
24	W	1	4	4	0	Basic	-M	M
	Objective	Function	Max.	77				

Table 3: Solution of constraint summary by using Win.QSB program

No. of constraint	Constraint	Left hand side	Direction	Right hand side	Slack or surplus	Shadow price	Allowable Min. RHS	Allowable Max. RHS
1	C1	-1	=	-1	0	0	-M	-1
2	C2	0	=	0	0	2	-1	0
3	C3	0	=	0	0	5	-1	0
4	C4	0	=	0	0	7	-1	0
5	C5	0	=	0	0	9	-1	0
6	C6	0	=	0	0	13	-1	0
7	C7	0	=	0	0	15	-1	0
8	C8	0	=	0	0	25	-1	0
9	C9	0	=	0	0	40	-1	0
10	C10	0	=	0	0	42	-1	0
11	C11	0	=	0	0	64	-1	0
12	C12	0	=	0	0	51	-1	0
13	C13	0	=	0	0	69	-1	0
14	C14	0	=	0	0	55	-1	0
15	C15	0	=	0	0	61	-1	0
16	C16	0	=	0	0	68	-1	0
17	C17	0	=	0	0	73	-1	0
18	C18	1	=	1	0	77	0	1

indicates the status of whether the constraint is tight or not. This command is available when the optimal solution is achieved. This table contains some important terms too:

Shadow price: The shadow price of a constraint is the marginal change of the objective function when the right-hand side value of that constraint increases by one unit, for example the objective function is increased 2weeks if the right-hand side value of the second Constraint (C2) increases by one unit (Table 3 Column7).

Sensitivity analysis for RHS: This Analysis shows the ranges of the right-hand sides such that the current basis holds. For each constraint, this includes the lower limit and upper limit allowed for its right-hand side so that the current basic variable is still feasible. This is also called the range of feasibility.

This analysis is available when the optimal solution is achieved.

Right hand sensitivity of the constraints provides us information regarding the status of the constraints- which of these constraints are binding (fully utilized) or non-binding.

Binding constraints having a value in the shadow price column (Table 3 Column7) other than zero, means how much contribution these binding constraints will provide individually in the objective function, if the value of the right hand side of these constraints are increased by 1 unit.

The allowable min. RHS and allowable max. RHS columns in Table 3 indicate the range of allowable decrease and range of allowable increase of the right hand side value of the binding resources within which the

Table 4: Head and tail events activities

Activities	Tail event (Starting event)	Head event (Ending event)
A	1	2
B	2	3
C	3	4
D	4	5
E	5	6
F	3	6
G	6	7
H	7	8
I	8	9
J	8	10
M	8	12
N	8	14
O	8	15
P	9	11
K	9	10
L	10	12
Q	12	14
R	14	15
DUM	10	11
T	11	13
V	13	17
S	15	16
U	16	17
W	17	18

current shadow price would remain unchanged. So, this model would help management to reach the optimality where sensitivity analysis would provide some flexibility in the model. So the right hand side of the C1 is -1, allowable min. c(j) is -M and allowable max. c(j) is -1. It indicates our current solution would remain optimum if C1 varies from -M to -1.

Model 2. To crash the duration of project completion: For the purpose of modeling the problem, it is necessary to define the activities in terms of starting and ending event. The total number of events in this project is 18 (Table 4).

Objective function of (Model II):

$$\begin{aligned} \text{Minimize}(Z) = & 10000X_A + 20000X_B + 10000X_C + \\ & 10000X_D + 25000X_E + 30000X_F + \\ & 100000X_G + 40000X_H + 40000X_I + \\ & 25000X_J + 5000X_K + 40000X_L + \quad (27) \\ & 5000X_M + 10000X_N + 15000X_O + \\ & 40000X_P + 75000X_Q + 70000X_S + \\ & 50000X_T + 35000X_U + 35000X_V + \\ & 25000X_W + 0X_{DUM} \end{aligned}$$

This objective function subject to the following constraints:

Maximum reduction constraints:

$$\begin{aligned} X_A &\leq 1 & (28) \\ X_B &\leq 2 & (29) \\ X_C &\leq 1 & (30) \\ X_D &\leq 1 & (31) \\ X_E &\leq 2 & (32) \\ X_F &\leq 1 & (33) \\ X_G &\leq 1 & (34) \\ X_H &\leq 3 & (35) \\ X_I &\leq 5 & (36) \\ X_J &\leq 2 & (37) \\ X_K &\leq 1 & (38) \\ X_{DUM} &\leq 0 & (39) \\ X_L &\leq 3 & (40) \\ X_M &\leq 1 & (41) \\ X_N &\leq 1 & (42) \\ X_O &\leq 2 & (43) \\ X_P &\leq 3 & (44) \\ X_Q &\leq 2 & (45) \end{aligned}$$

$$X_S \leq 2 \quad (46)$$

$$X_T \leq 2 \quad (47)$$

$$X_U \leq 2 \quad (48)$$

$$X_V \leq 2 \quad (49)$$

$$X_W \leq 2 \quad (50)$$

Start time constraints: We begin by setting the event occurrence time for event 1 to be $Y_1 = 0$. The constraints describe the structure of the network are as follows:

$$Y_1 \geq 0 \quad (51)$$

$$Y_1 + 2 - X_A \geq Y_2 \quad (52)$$

$$Y_2 + 3 - X_B \geq Y_3 \quad (53)$$

$$Y_3 + 2 - X_C \geq Y_4 \quad (54)$$

$$Y_4 + 2 - X_D \geq Y_5 \quad (55)$$

$$Y_5 + 4 - X_E \geq Y_6 \quad (56)$$

$$Y_3 + 3 - X_F \geq Y_6 \quad (57)$$

$$Y_6 + 2 - X_G \geq Y_7 \quad (58)$$

$$Y_7 + 10 - X_H \geq Y_8 \quad (59)$$

$$Y_8 + 15 - X_I \geq Y_9 \quad (60)$$

$$Y_8 + 7 - X_J \geq Y_{10} \quad (61)$$

$$Y_9 + 2 - X_K \geq Y_{10} \quad (62)$$

$$Y_8 + 2 - X_M \geq Y_{12} \quad (63)$$

$$Y_{10} + 9 - X_L \geq Y_{12} \quad (64)$$

$$Y_8 + 3 - X_N \geq Y_{14} \quad (65)$$

$$Y_{12} + 4 - X_Q \geq Y_{14} \quad (66)$$

$$Y_8 + 6 - X_O \geq Y_{15} \quad (67)$$

$$Y_{14} + 6 - X_R \geq Y_{15} \quad (68)$$

$$Y_9 + 7 - X_P \geq Y_{11} \quad (69)$$

$$Y_{10}+0 - X_{DUM} \geq Y_1 \quad (70)$$

$$Y_{11}+5 - X_T \geq Y_{13} \quad (71)$$

$$Y_{15}+7 - X_S \geq Y_{16} \quad (72)$$

$$Y_{16}+5 - X_U \geq Y_{17} \quad (73)$$

Or,

$$Y_{13}+4 - X_V \geq Y_{17} \quad (74)$$

$$Y_{17}+4 - X_W \geq Y_{18} \quad (75)$$

- Project completion desire time

$$Y_{18} \leq 0 \quad (76)$$

Nonnegative constraints: All the decision variable $Y_i \geq 0$ i.e., A, B, C, ..., W ≥ 0 .

As the manager wants to complete the project within 50 weeks, so the last event should be completed before or on 50th week. By analyzing the project on crash time basis the expected date of completion of the project is 50 weeks. So, the maximum extent of crashing the project is 46 weeks. Beyond this time period, the project cannot be crashed.

Solution summary table (model 2): Solution summary for model 2 is presented in Table 5, which shows the solution of the problem.

Solution summary in Table 4 indicates that an activity A should be crashed at 1 weeks, B at 2 weeks, C at 1 week, etc. (Table 5). Subtracting the crash-time amounts from the normal completion, the result indicates that activity A should be completed in 1 week, B in 1 week, C in 1 week, etc. the adding cost(crash cost) by crashing critical activities to reduce the project duration within 50 weeks

Table 5: Solution summary by using Win.QSB program

No. of variable	Decision variable	Solution value	Unit cost or profit c(j)	Total contribution	Reduced cost	Basis status	Allowable Min. c(j)	Allowable Max. c(j)
1	XA	1	10,000	10,000	0	basic	-M	75,000
2	XB	2	20,000	40,000	0	basic	-M	75,000
3	XC	1	10,000	10,000	0	basic	-M	75,000
4	XD	1	10,000	10,000	0	basic	-M	75,000
5	XE	2	25,000	50,000	0	basic	-M	75,000
6	XF	0	30,000	0	30,000	at bound	0	M
7	XG	0	100,000	0	25,000	at bound	75,000	M
8	XH	3	40,000	120,000	0	basic	-M	75,000
9	XI	5	40,000	200,000	0	basic	-M	75,000
10	XJ	0	25,000	0	25,000	at bound	0	M
11	XK	1	5,000	5,000	0	basic	-M	75,000
12	XL	3	40,000	120,000	0	basic	-M	75,000
13	XM	0	5,000	0	5,000	at bound	0	M
14	XN	0	10,000	0	10,000	at bound	0	M
15	XO	0	15,000	0	15,000	at bound	0	M
16	XP	0	40,000	0	40,000	at bound	0	M
17	XQ	1	75,000	75,000	0	basic	70,000	75,000
18	XR	3	70,000	210,000	0	basic	-M	75,000
19	XS	0	75,000	0	0	at bound	75,000	M
20	XT	0	50,000	0	50,000	at bound	0	M
21	XU	2	35,000	70,000	0	basic	-M	75,000
22	XV	0	35,000	0	35,000	at bound	0	M
23	XW	2	25,000	50,000	0	basic	-M	75,000
24	XDUM	0	0	0	0	at bound	0	M
25	Y1	0	0	0	0	basic	-M	M
26	Y2	1	0	0	0	basic	-65,000	M
27	Y3	2	0	0	0	basic	-55,000	M
28	Y4	3	0	0	0	basic	-55,000	M
29	Y5	4	0	0	0	basic	-55,000	M
30	Y6	6	0	0	0	basic	-50,000	M
31	Y7	8	0	0	0	basic	-50,000	25,000
32	Y8	15	0	0	0	basic	-35,000	25,000
33	Y9	25	0	0	0	basic	-35,000	25,000
34	Y10	26	0	0	0	basic	-35,000	25,000
35	Y11	39	0	0	0	basic	-35,000	0
36	Y12	32	0	0	0	basic	-35,000	25,000
37	Y13	44	0	0	0	basic	-35,000	0
38	Y14	35	0	0	0	basic	0	5,000
39	Y15	38	0	0	0	basic	0	40,000
40	Y16	45	0	0	0	basic	-M	40,000
41	Y17	48	0	0	0	basic	-M	50,000
42	Y18	50	0	0	0	basic	-M	75,000
	Objective	Function	(Min.) =	970,000	(Note:	Alternate	Solution	Exists!)

is \$970,000 and the cost to complete the project in normal duration(77 weeks) is \$5,120,000. So, the final project cost is computed by adding cost (crash cost) by crashing critical activities \$970,000 to the normal cost \$5,120,000. So, the final project cost is \$6,090,000.

The reduced cost of the non-basic variables (the variables whose value is zero in the optimum solution) provide us the information about how much objective coefficients of these variables should be increased to have a positive value of those variables in the optimum solution.

In the example, reduced cost of a current non-basic variable X_r is 30,000.

It means the current coefficient of this variable which is now 30,000 must increased 30,000 (that means the coefficient would be 60,000 or up) to get a basic (positive) value of this variable in the optimum solution (Table 5).

Sensitivity analysis for OBJ shows the ranges of the objective function coefficients such that the current basis holds. In our hypothetical example, the final value of variable X_A in the objective function is 1. The current coefficient of the variable is 10,000, allowable min. $c(j)$ is $-M$ and allowable max. $c(j)$ is 75,000 (Table 5). It indicates our current solution would remain optimum if crash cost per unit of time for activity A varies from $-M$ to 75,000.

Constraint summary table: Constraint summary indicates the status of whether, the constraint is tight or not. This command is available when the optimal solution is achieved.

The slack variable is the variable added to the left-hand side of a less than or equal to constraint to convert the constraint into an equality. The slack variable is the starting basic variable for the constraint. It can be interpreted as the unused resource or right-hand side.

Table 6: Solution of constraint summary by using Win.QSB program

No. of constraint	Constraint	Left hand side	Direction	Right hand side	Slack or surplus	Shadow price	Allowable Min. RHS	Allowable Max. RHS
1	C1	1	\leq	1	0	-65,000	0	2
2	C2	2	\leq	2	0	-55,000	1	3
3	C3	1	\leq	1	0	-65,000	0	2
4	C4	1	\leq	1	0	-65,000	0	2
5	C5	2	\leq	2	0	-50,000	1	3
6	C6	0	\leq	1	1	0	0	M
7	C7	0	\leq	1	1	0	0	M
8	C8	3	\leq	3	0	-35,000	2	4
9	C9	5	\leq	5	0	-35,000	4	6
10	C10	0	\leq	2	2	0	0	M
11	C11	1	\leq	1	0	-70,000	0	2
12	C12	3	\leq	3	0	-35,000	2	4
13	C13	0	\leq	1	1	0	0	M
14	C14	0	\leq	1	1	0	0	M
15	C15	0	\leq	2	2	0	0	M
16	C16	0	\leq	3	3	0	0	M
17	C17	1	\leq	2	1	0	1	M
18	C18	3	\leq	3	0	-5,000	2	4
19	C19	0	\leq	2	2	0	0	M
20	C20	0	\leq	2	2	0	0	M
21	C21	2	\leq	2	0	-40,000	1	3
22	C22	0	\leq	2	2	0	0	M
23	C23	2	\leq	2	0	-50,000	1	3
24	C56	0	\leq	0	0	0	0	M
25	C57	0	$=$	0	0	75,000	0	1
26	C24	2	\geq	2	0	75,000	1	3
27	C25	3	\geq	3	0	75,000	2	4
28	C27	2	\geq	2	0	75,000	1	3
29	C28	2	\geq	2	0	75,000	1	3
30	C29	4	\geq	4	0	75,000	3	5
31	C30	4	\geq	3	1	0	$-M$	4
32	C31	2	\geq	2	0	75,000	1	3
33	C32	10	\geq	10	0	75,000	9	11
34	C33	15	\geq	15	0	75,000	14	16
35	C34	2	\geq	2	0	75,000	1	3
36	C35	11	\geq	7	4	0	$-M$	11
37	C36	13	\geq	0	13	0	$-M$	13
38	C37	14	\geq	7	7	0	$-M$	14
39	C38	9	\geq	9	0	75,000	8	10
40	C39	17	\geq	2	15	0	$-M$	17

The surplus variable is the variable subtracted from the left-hand side of a greater than or equal to constraint to convert the constraint into an equality. It can be interpreted as the amount over the requirement or right-hand side. So, we can notice the slack for binding constraints (fully utilized) are zero (Table 6 Column 6).

The shadow price of a constraint is the marginal change of the objective function when the right-hand side value of that constraint increases by one unit. So, the objective function is increased \$65,000 if the right-hand side value of the first Constraint (C1) increases by one unit (Table 6 Column 7).

Right hand sensitivity of the constraints provides us information regarding the status of the constraints-which of these constraints are binding (fully utilized) or non-binding.

Binding constraints having a value in the shadow price column (Table 6) other than zero.

The allowable min. RHS and allowable max. RHS columns in Table 6 indicate the range of allowable decrease and range of allowable increase of the right hand side value of the binding resources within which the current shadow price would remain unchanged. So, this model would help management to reach the optimality where, sensitivity analysis would provide some flexibility in the model, for example the right hand side of (C1) is 1, allowable Max. $c(j)$ is 0 and allowable Max. $c(j)$ is 2. It indicates our current solution would remain optimum if constraint 1 varies from 0-2.

CONCLUSION

These models will provide us systematic and logical approaches for decision making and ultimately increases the effectiveness of the decision. As the solution of these models by software package (Win.QSB) provide us the duration of project completion in normal and crash conditions, also gives us some flexibility by providing a combined report of the problem, which includes the solution value, contribution to the objective, reduced cost and range of optimality for each decision variable and right-hand side, surplus or slack, shadow price and range of feasibility for each constraint.

In this research, we also propose an algorithmic model based on linear programming incorporated with a minimal time-cost trade-off in a construction project. The format of the model lends itself to a wide range of variables and considerations.

The present modeling strategy has shown the resources of this interactive approach including a bulk of data to completely analyze the project is easily possible.

It allows a great number of parameters to simulate project conditions and contractor's preference and provides potentially useful tool for decision making on project scheduling.

The cost of the network activities has been optimized for various overall duration. The optimum trade-off of time against cost has been made. This approach is an acceptable tool of management and proving to be not only superior method for planning, scheduling and controlling project progress, but also is very real and valuable assets to contractors in convincing the owner of their potentials and abilities. With the introduction of better and more rigorous methods of planning research, together with cost analysis, the construction control will become more systematic. Mathematical models are used more and more for executive planning functions. In all of these, decisions must be made to carry out the operation in the best way possible in light of the restraints that are bound to exist.

This approach allows the user to easily manipulate different project networks of various difficulties representing real world applications and to study the effectiveness of the model in the case of large projects. The implementation of the developed model is tested on a large number of linear optimization problems and shown to have more efficient and reliable results, generates a considerable computational savings, along with an increase in robustness.

Defining the variables:

Let,

- Y_1 = Time when event 1 will occur
- Y_2 = Time when event 2 will occur
- Y_3 = Time when event 3 will occur
- Y_4 = Time when event 4 will occur
- Y_5 = Time when event 5 will occur
- Y_6 = Time when event 6 will occur
- Y_7 = Time when event 7 will occur
- Y_8 = Time when event 8 will occur
- Y_9 = Time when event 9 will occur
- Y_{10} = Time when event 10 will occur
- Y_{11} = Time when event 11 will occur
- Y_{12} = Time when event 12 will occur
- Y_{13} = Time when event 13 will occur
- Y_{14} = Time when event 14 will occur
- Y_{15} = Time when event 15 will occur
- Y_{16} = Time when event 16 will occur
- Y_{17} = Time when event 17 will occur
- Y_8 = Time when event 18 will occur
- X_A = Number of days activity A will be crashed
- X_B = Number of days activity B will be crashed

- X_C = Number of days activity C will be crashed
- X_D = Number of days activity D will be crashed
- X_E = Number of days activity E will be crashed
- X_F = Number of days activity F will be crashed
- X_G = Number of days activity G will be crashed
- X_H = Number of days activity H will be crashed
- X_I = Number of days activity I will be crashed
- X_J = Number of days activity J will be crashed
- X_M = Number of days activity M will be crashed
- X_N = Number of days activity N will be crashed
- X_O = Number of days activity O will be crashed
- X_P = Number of days activity P will be crashed
- X_K = Number of days activity K will be crashed
- X_L = Number of days activity L will be crashed
- X_Q = Number of days activity Q will be crashed
- X_{DUM} = Number of days activity DUM will be crashed
- X_T = Number of days activity T will be crashed
- X_V = Number of days activity V will be crashed
- X_S = Number of days activity S will be crashed
- X_U = Number of days activity U will be crashed
- X_W = Number of days activity W will be crashed

REFERENCES

Elmaghraby, S.E. and P.S. Pulat, 1997. Optimal project compression with due-dated events. *NRL*, 26: 331-348. DOI: 10.1002/nav.3800260213.

Hillier, F.S. and G.J. Liberman, 2001. *Introduction to Operations Research*. 7th Edn. MC Graw-Hill. New York, pp: 468-532. ISBN: 0072321695.

Islam Mohammad Nazmul and Baktiar Rana and Rafique Shuddasattawa and Aziz Tasanima, 2004. Crashing project time with least cost: A linear programming a pproach. *J. Bus. Res.*, 6: 320-335. <http://ssrn.com/abstract=1012525>.

Lawrence, J.A. Jr. and B.A. Pasternack, 1999. *Applied Management Science: A Computer-Integrated Approach for Decision Making*. 2nd Edn. John Wiley and Sons, New York, pp: 370-433. ISBN: 978-0-471-42774-2.

Natarajan, A.M., P. Balasubramani and A. Tamilarasi, 2005. *Operations Research, Person Second Education (Singapore) Pte. Ltd.*, pp: 280-322. ISBN: 64039-271 119.

Nicholas, J.M., 2004. *Project Management for Business Principles and Practice*. 2nd Edn. Elsevier Inc. Printed in the United States of America, pp: 193-223. ISBN: 0-7506-7824-0.

O'Brien, J.J. and F.L. Plotnick, 2006. *CPM in Construction Management*. 6th Edn. MC Graw-Hill Company, pp: 73-80. ISBN: 0-07-158911-2. DOI: 10.1036/0071457 690.

Senouci, A.B. and N.N. Eldin, 1996. A time-cost trade-off algorithm for non serial linear projects. *Can. J. Civ. Eng.*, 23: 134-149. DOI: 10.1139/196-014.