AN EFFICIENT ALGORITHM FOR CRASHING

B. R. Kharde, G. J. Vikhe Patil

Abstracts: Time-Cost Trade-off in Projects (TCTP), Least-Cost Schedule (LCS) or crashing technique is used to find optimum project duration to minimize the total cost. In crashing an activities, the direct cost (DC) increases while indirect cost(IC) reduces. So it is double beneficiary technique for managers to decrease the project duration as well as total cost. The goal in crashing is to find the optimum duration or Least Cost Schedule (LCS) where the total cost of the project is least. Unit Time Method (UTM) is the powerful procedure for crashing; yields always optimum solution and is used widely for CPM networks. But much iteration (one for crashing one unit of time) are needed to get to LCS. This is a disadvantage of UTM if project is to be crashed for double figure or more time. Say project crashing for 30 days – 30 iterations! Other short cuts avoiding UTM are error porn and errors are observed in few cases (literature). We propose new algorithm which works on UCM logic but requires less iteration. In some problems iterations are reduced to just number of activities crashed till LCS. Algorithm can be viewed as modified Unit Time Method; would always yield the optimum in very less iterations (10 to 30% approximately).

Index Terms- CPM, PERT, Crashing, time-cost trade-off, Least Cost Schedule, Economic Crash Limit, Unit Time Method

1. INTRODUCTION

CPM is for deterministic times while PERT is for stochastic times. In CPM project duration which is Critical path length (CPL) need to shorten for various requirements of project. Crashing technique is very powerful tool for managers to expedite and reduce the project cost. Smith (1997) shows the different algorithms comparison in excursions. Various algorithms (Liu (1995), Reda (1989), Senouci (1996) …) basically written in the angle of Computer engineering, do not take much in account of the users from construction industry. Senouci (1996) presents dynamic programming approach, but DP is not liked by most of users for its questionable simplicity. Reda (1989) developed LPP model but its application to construction industries is questionable. Gupta (2006) crashes cheapest activity of the network. *Always crash activity only from critical path;* if MCS, TCTP or *economical* duration for the project is the goal Stevens (1996) illustrates networks, dummy adding method of drawing AOA, and Unit Time Method (UTM) for crashing. UTM is the best techniques. In this cheapest activity from the critical path is always crashed for unit time (day/week/…). Its optimal solution is at cost of one-iteration for one-unit-time crashing. If project requires crashing of double figure time (that is what generally required in industries); it is challenging. We present algorithm, with same logic, yields optimal solution but takes very less iterations.

We set bounds/limits for crashing an activity without affecting next path to become critical. The basic goal is to minimize the total cost of project keeping intact the technological constraints. The problem is not solved by LPP; but by algorithmic approach to yield the optimal.

Crashing a Network is as follow:

1. Compute the network critical path
2. Establish an objective total duration
3. Identify the crash time for each activity
4. Prioritize the activities on the critical path(cost slope)
5. Shorten the highest-priority activity by one time period and
6. compare total duration with objective
7. Verify critical path
8. Continue activity reduction (step 4 & 5) until economic crash limit is reached
9. Select next priority activity and continue reduction (steps 4 through 6)

The weakness of crashing unit time is explored such that one activity could be crashed for more than unit time in most of the situations.

Prof. B.R. Kharde,
Production department, Amrutvahini College of Engineering, Sangamner, India, 422608
khardebr@yahoo.com

Dr. G.J. Vikhe Patil,
Principal, Amrutvahini College of Engineering, Sangamner, India, 422608
gjvploni@yahoo.co.in

2. NOTATION

i	Start event ' i' for an activity
j	End event 'j' for an activity
A_{ij}	Activity having start event ' i' and end event 'j'
T_{ij}	time or duration for A_{ij}
NT_{ij}	Normal time for A_{ij}
CT_{ij}	Crash time for A_{ij}
Δ_{ij}	NT_{ij} -CT_{ij}
Δ_K	NT_K -CT_K
NC_{ij}	Normal Cost for A_{ij}
CC_{ij}	Crash Cost for A_{ij}
CS_{ij}	Cost slope for A_{ij} = $\frac{CC_{ij} - NC_{ij}}{NT_{ij} - CT_{ij}}$
CS_K	Cost slope for Activity K
TCS_{ij}	= Σ CS_{ij} for least cost slope activities from different CPs = CS_{ij} for least cost slope activity from critical path
CL_{ij}	Crash length for activity $_{ij}$ = CT_{ij} - CT_{ij}
NC_{ij}	Normal Cost for A_{ij}
CP_K	Project path "K" which happens to be critical
CPL_K	Length for critical path 'K, $CP_K = \Sigma T_{ij}$ for $A_{ij} \in K$
NP_K	the path of the project just shorter(Next) to critical path 'K' NP_K
NNP_K	the path of the project just shorter(Next) to NP_K path 'K' NP_K
NPL_K	Length (duration) for the path of the project just shorter(Next) to critical path 'K" CP_{NK} = ΣT_{ij} for $A_{ij} \in N$
$NNPL_K$	Length (duration) for the path of the NNP_K
F_K	Critical Path(K) Float Limit or Difference between critical path length and length of next to critical path = CPL_K - NPL_K
NF_K	Critical Path(K) Float Limit or Difference between critical path length and length of next to next to critical path, = CPL_K - $NNPL_K$
CTL_{ij}	Crash-Time-Limit or maximum limit activity ' i-j ' could be crashed in one stretch, = min { F_K , (NT_{ij} - CT_{ij})}
$NCTL_{ij}$	Crash-Time-Limit or maximum limit activity ' i-j ' could be crashed in one stretch when crash activity is common to CP and NP, = min { NF_K , (NT_{ij} - CT_{ij})} = min { NF_K , Δ_{ij})

3. NETWORK AND CRITICAL PATH

We use Activity On Arc (AOA) however AON network could be used. Crashing could be done without network preparation by path table.

Critical Path Float Limit

Projects have many paths which are evident from the network. These paths are denoted by number, 1, 2, 3…, K,… (Table 3.1) Each path has length. Critical path/s is/are the path/s with longest duration. This length is denoted by CPL_K. The path which is just shorter than critical path (K) is denoted by NP_K and its length by NPL_K.

Definition: *The difference in these two paths is defined as Critical Path Float Limit,* $F_K = CPL_K - NPL_K$ Activities on critical path have no floats. All floats for critical activities are zero (Total Float, Free float, Safety float, Independent float, and Interfering float). The new definition is not activity float but path float.

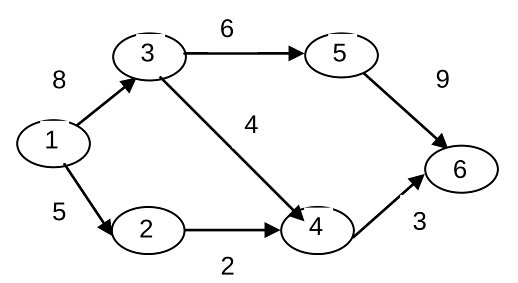


Figure: 3.1

Consider AOA network showing activity durations in Figure 3.1. Three different paths are evident as listed in Table 3.1.

Table 3.1 – Path Table

	Path	Length
1	1-3-5-6	8+6+9 = 23
2	1-3-4-6	8+4+3 = 15
3	1-2-4-6	5+2+3 = 10

- The network have three paths namely path 1, 2 and 3. Path 1 is longest and hence critical path; CP_1 have length CPL_1 = 23 week.
- Next path just short to critical path is path 2 and its length is 15 weeks. This path is relative to critical path 1; NP_1 have length NPL_1 = 15 week.
- The difference between these paths is 8. So with notations, it is *Critical Path Float Limit* for *critical path 1*: $F_1 = CPL_1 - NPL_1 = 23 - 15 = 8$ week

When activity on CP is crashed, the length of CP would be reduced by crashed time. Its effect on NP is observed:

- Case 1: No change in NPL
- Case 2: NPL reduction

E.g. If activity 3-5 is crashed by 1 week, then

CPL_1 = 23-1 = 22 week
NPL_1 = 15 NPL unchanged
F_1 = 22-15 = 7 week

Instead of crashing 3-5, if common activity to CP and NP; 1-3 is crashed by 1 week then:

CPL_1 = 7+6+9 = 22 week
NPL_1 = 7+4+3 = 14 week
NPL_1 is reduced from 15 to 14 week
F_1 = 22-14 = 8 week

Theorem 1*:* If activity on CP is crashed by unit time; NPL might be unchanged or would also reduce by unit time.

Theorem 2: If activity on CP is crashed by unit time; Critical Path Float Limit might be reduced by unit time or unchanged. In Case 2: Critical Path Float Limit is unchanged, and have no effect on criticality (CP is CP and NP is NP). This would continue till common activity which is being crashed; could not be crashed any further. But consider case 1; here for crashing one time unit, CPL reduces by one unit but NPL unchanged. So the crashing could continue for one more time unit, so on. A time would come when CPL and NPL are equal or both are CP. The situation changes now, crashing can not be done with logic of one CP. This point is when CP is crashed by the difference in (CPL – NPL) or Critical Path Float limit (F_K). This is the worst case situation. This infers:

Theorem 3*:* Activity on CP could be crashed by Critical Path Float Limit without affecting Criticality of NP (CP is CP and NP is NP).

Please note cost economics is not considered here.

Crash-Time-Limit for an Activity

Theorem 4: *Any activity on CP could be crashed maximum to crash period,* Δ_{ij} *or (*$NT_{ij} - CT_{ij}$). This is technological constraint on the activity and could not be violated. From theorem 3 and 4 it is evident

Theorem 5*: Any activity on CP could be crashed to Crash-Time-Limit in one stage such that Crash- Time-Limit; CTLij* $= min\ \{F_K; \Delta_{ij}\} = min\ \{F_K; \Delta_K\}$ Theorem 5 is evident because out of two possibilities (bounds) only least bound could be explored in one stage and it will not violate criticality. Our addition of theorem 3, 4 and 5 are implemented in Crashing. Instead of crashing one unit time, activity could be crashed by CTL, without any problem.

Crash-Time-Limit for an Activity when it is common on CP and NP

When activity 'K' is common to CP and NP; both paths would be shortened after the crashing. Hence F_K does not put any constraint on Crash limit. Under such case NNP_K should be taken for consideration. for calculation. $NF_K = CPL_K - NNPL_K$ Crash activity is common to CP, NP and NNP; then further path just shorter than $NNPL_{KN}$ should be considered. Such cases are rare but can not be neglected. NCTL for crashing when activity is common in CP and NP, $NCTL_K = min\ \{NF_K; \Delta_{ij}\} = min\ \{NF_K; \Delta_K\}$

4. COST SHEET

In literature no common standards are followed for crashing. We use cost sheet format from Stevens (1996).

Cost Sheet 4.1					
Activity	Crash time	Cost slope	Time shortened		
i-j	Δ_{ij}	CS_{ij}			
	Time Cut (crash)	////////			
	Project duration				
	Incremental Cost	////////			
	Direct Cost				
	Indirect Cost				
	Total Cost				

Where with notation

Time Cut (crash) = CTL_{ij}
Project duration = CPL_K
Incremental cost (IncC) = $CTL_{ij} * TCS_{ij}$
Direct Cost(DC) = ΣNC_{ij} + InC
Indirect cost(IC) = CPL * Indirect cost/time
Total cost(TC) = DC + IC

5. ALGORITHM USING CRASH-TIME-LIMIT (CTL) FOR CRASHING

We propose CTL algorithm as follow:
Step 1: Prepare table showing activity,
Immediate Predecessors, Normal time, Crash time, Normal cost and Crash cost for each activity Tabulate cost slope (CS_{ij}) or incremental cost per unit time for each activity
$CS_{ij} = (CC_{ij} - NC_{ij}) / (NT_{ij} - CT_{ij})$

Step 2: Prepare Cost-sheet with CS and CTL, Direct and Indirect cost,
Step 3: Prepare AON or AOA diagram.
Show NT on network. Prepare a Path-table showing different paths and find lengths of the each path. Note CP, NP, CPL, NPL and F_K from Path table.(*No need to run Forward/backward passes*)
Step 4: Noting CPL, prepare (1st column of) Cost-Sheet and Total Direct cost(TNC), Total Indirect cost and Total cost of the project
Step 5: If no activity from any one CP could not be crashed, Then Stop, It is crash limit.
If more than one critical path(CP), then go to Step 6, Else go to step 7
Step 6: If any activity/ies is/are common to all CPs which could be crashed
Then note the common activity with least cost slope from CPs (CSC)
And Note one least cost slope activity *which could be crashed* **from each CP**
CSTotal = add cost slope of these activities
If (CSC < CSTotal) then go to step 8: Else

1. Find Crash Time Limit for each such activity
2. Take minimum CTL
3. Reduce each least cost slope activity by minimum CTL
4. Reduce CTL in Network Diagram
5. Update path Table and note CPs, new CPL, new F_K
6. Update Cost sheet for this CPL (Total Incremental cost, DC, Indirect Cost and TC)
7. Go To Step 9

Step 7: Note the activity (K) with least cost slope *which could be crashed** ***from CP***,.
If crash activity(K) is common to CP and NP; Then crash it by $NCTL_K$; Else crash by CTL_{ij}
Step 8:

1. Find CTL or NCTL as applicable
2. Reduce activity to be crashed(K) by CTL or NCTL as applicable in network
3. Update Path table.
4. Find CP, CPL, F_K
5. Update Cost sheet for CPL (Total incremental cost, DC, IC, TC)

Step 9: If Total cost of the project is increased Then Stop; least total cost is the optimum period; solution; Else go to Step 5:

6. ILLUSTRATION

Problem data is given in table 6.1 and AOA network diagram 6.1

Table 6.1

Activity		IP	Duration (days)		Cost ('000 $)	
-	event		Normal	Crash	Normal	Crash
-	' i –j'		NT_{ij}	CT_{ij}	NC_{ij}	CC_{ij}
A	1-2	-	20	14	1600	2170
B	2-4	A	10	6	140	220
C	2-3	A	20	12	800	1720
E	4-5	B, C	40	30	800	1050
F	3-5	C	10	8	1000	1050

Indirect cost of the project is $ 100,000 per day

Step 1: prepare Cost-Slope Table (Table 6.2)

Table 6.2: Cost-Slope Table

Activity		IP	Duration (days)		Cost ('000 $)		Δ time	Cost Slope
-	event		Normal	Crash	Normal	Crash	NT_{ij} -CT_{ij}	CS_{ij} ('000 $/day)
-	' i –j'		NT_{ij}	CT_{ij}	NC_{ij}	CC_{ij}		
A	1-2	-	20	14	1600	2170	6	95
B	2-4	A	10	6	140	220	4	20
C	2-3	A	20	12	800	1520	8	90
E	4-5	B, C	40	30	800	1050	10	25
F	3-5	C	10	8	1000	1050	2	25
TOTAL					4340			

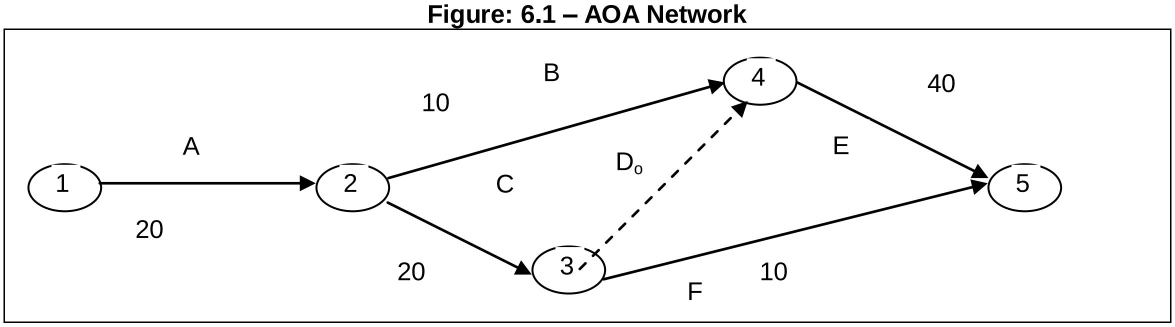
Step 2: prepare Cost sheet (Table 6.3)

Table 6.3 - Cost Sheet

Activity	Δ time	CS_{ij}	Days shortened					
	days	'000 $/day						
A	6	95						
B	4	20						
C	8	90						
E	10	25						
F	2	25						
Days cut		////////////////						
Project duration(CPL)		80						
Incremental cost(IncC)		////////////////						
Direct cost(DC)		4340						
Indirect cost(IC)		8000						
Total cost(TC)		12340						

Step 3: AOA network is prepared (Figure 6.1)

Figure: 6.1 – AOA Network



1. Prepare Path table (table 6.4)

Table 6.4 - Path Table

Path	I	A-B-E	20+10+40 = 70				
	II	A-C-D_o-E	20+20+40 = 80 *				
	II	A-C-F	20+20+10 = 50				
Critical path, CP			II				
Critical path duration, CPL			80				
Next to CP length, NPL			70				
Critical path float limit, F			10				
NCP Float limit, NF			20				
Column			i				

Step 4: update Cost Sheet (data to be put in Table 6.3)

Days cut	/////////////////
Project duration, CPL_{II}	80 days
Incremental cost (IncC)	//////////////////
Direct cost (DC)	ΣNC_{ij} = 4340; IncC = 0; DC = 4340
Indirect cost (IC)	CPL * Indirect cost/time = 80*100 = 8000
Total cost(TC)	= DC+ IC = 12340

Step 5: CP is path II; Only one path; go to Step 7:

Step 7: Path II is critical path; Critical activities are A, C and E (from Path table, path II). From Cost sheet: CS_A = 95; CS_C = 90; and CS_E = 25 *****

- Least cost slope activity is E from CP; It is common to CP(II)and NP(I) but not common to NNP(III)
  NF_{II} = CPL_{II} - $NNPL_{III}$ = 80 - 50 = 30 days
  $NCTL_{II}$ = min { NF_{II} ; Δ_E} = min { 30,10 } = 10 days.
  Activity E to be crashed by 10 days

Step 8:

- Activity 'E' is reduced by 10 days in network
- Update Network.
- Update Path table (column ii; Table 6.5-not shown). CP is path II; CPL_{II} = 70 days; NP is path I ; NPL_{II} = 60 days; F_{II} = 70 - 60 = 10 days
  1. Cost sheet updating(Cost-Sheet 6.4-A): Activity E is crashed by 10 days; Δ_E is reduced by 10 days from 10 to 0;

Days cut = CTL = 10 days; Activity E is crashed by 10 days @ CS_E

IncC = 10 * CS_E = 10 * 25 = 250

DC = 4340 + 250 = 4590; IC = 100 * 70 = 7000; and TC = DC + IC = 4590 + 7000 = 11590.

Step 9: TC (previous) = 12340; TC (after crash) = 11590. As Total cost is reduced, go to step 5:

Step 5: Critical path is path II; only one. go to step 7:

Table 6.3-A: Cost-Sheet (partial)			
Acti-vity	Δ time	CS_{ij}	Days short-ened
	days		
A	6	95	*
B	4	20	
C	8	90	*
E	~~10~~, 0	25	10*
F	2	25	
Days cut		//////////	10
Project duration (CPL)		80	70
Incremental cost (IncC)		//////////	250
Direct cost (DC)		4340	4590
Indirect cost (ID)		8000	7000
Total cost (TC)		12340	11590

Step 7: Path II is critical path; Critical Activities which can be crashed; are A and C (from Cost sheet; E have Δ_E = 0). From Cost sheet: CS_A = 95 and CS_C = 90**

Least cost slope activity is C and Δ_C = 8 days (From Cost sheet)

So Activity C could be crashed. Activity C is not common to CP and NP.

For activity C; Δ_C = 8 days (cost sheet) and F_{II} = 10 days (Path table)

CTL_C = min {Δ_C; F_{II}} = min {8, 10} = 8 day Activity 'C' to be crashed by 8 days

Step 8: Activity 'C' is reduced by 8 days in network

1. Update Network.
2. Update Path table (column iii; Table 5.3): CP is path II; CPL_{II} = 62 days; NP is path I ; NPL_{II} = 60 days; F_{II} = 62 - 60 = 2 days
3. Cost sheet updating: Activity C is crashed by 8 days; Δ_C is reduced by 8 days from 8 to 0; Days cut = CTL = 8 days. Activity C is crashed by 8 days@ CS_C. IncC = 8 * CS_C = 8 * 90 = 720.
   DC = 4590 + 720 = 5310; IC = 100 * 62 = 6200 and

TC = DC + IC = 5310 + 6200 = 11510

Step 9: TC (previous) = 11590 and TC (after crash) = 11510. Total cost is reduced, go to Step 5:

Step 5: Critical path is path II; only one; go to step 7:

Step 7: Path II is critical path;

- Critical Activities are A, C and E (from Path table).
- From Cost sheet: only Activity A could be crashed.
- As Δ_E = Δ_C = 0; CS_A = 95 ***
- Least cost slope activity is A; It is common to CP (II), NP(I) and NNP(III). It does not constraint the $F_{II.}$ However finding NCTL:

NF_{II} = CPL_{II} - $NNPL_{III}$ = 56 – 36 = 20;

$NCTL_{II}$ = *min { NF_{II} ;* Δ_K} = min { 20,6 } = 6 days

Activity A to be crashed by 6 days

Step 8:

1. Activity 'A' is reduced by 6 days in network
2. Update the Network: Update the Path table (column iii; Table 5.3):

CP is path II; CPL_{II} = 56 days; NP is path I ; NPL_{II} = 54 days; F_{II} = 56 - 54 = 2 days

Cost sheet updating:

Activity C is crashed by 6 days; Δ_A is reduced by 6 days from 6 to 0;

Days cut = CTL = 6 days; Activity A is crashed by 6 days @ CS_A

IncC = 6 * CS_A = 6 * 95 = 570;

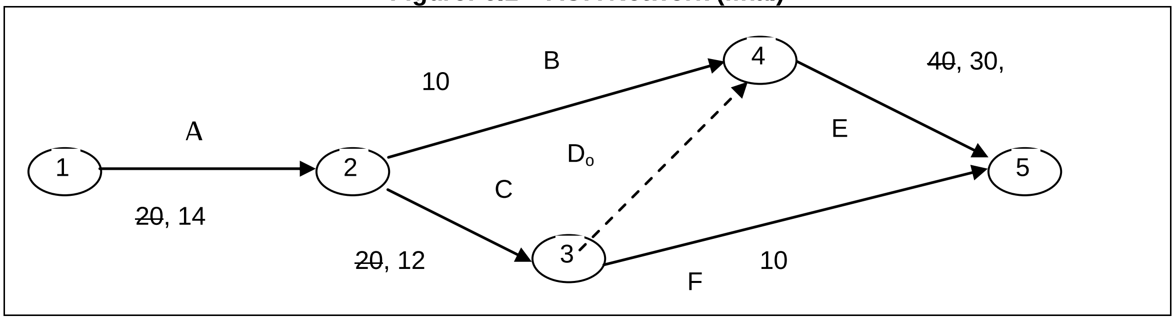
DC = 5310 + 570 = 5880; IC = 100 * 56 = 5600; and TC = DC + IC = 5880 + 5600 = 11480

Step 9: TC (previous) = 11510 and TC (after crash) = 11480

Step 5: Critical path is path II; only one hence go to step 7:

At end Figure 6.1 is as shown below

Figure: 6.1 – AOA Network (final)



At end Path table (Table 6.4) is as shown below

Table 6.4 - Path Table (final)

Path	I	A-B-E	20+10+40 = 70	60	60	54	
	II	A-C-D_o-E	20+20+40 = 80 *	70 *	62 *	56 *	
	II	A-C-F	20+20+10 = 50	50	42	36	
Critical path, CP			II	II	II	II	
Critical path duration, CPL			80	70	62	56	
Next to CP length, NPL			70	60	60	54	
Critical path float limit, F			10	10	2	2	
NCP Float limit, NF			20	20	20	20	
Column			i	ii	iii	iv	v

At end Cost-Sheet (Table 6.3) is as shown below

Table 6.3 - Cost Sheet (final)

Activity	Δ time	CS_{ij}	Days shortened					
	days	'000 $/day						
A	6	95	*	*	* 6	-		
B	4	20						
C	8	90	*	* 8	-	-		
E	10	25	* 10	-	-	-		
F	2	25						
Days cut	////////////////		10	8	6			
Project duration		80	70	62	56			
Incremental cost	////////////////		250	720	570			
Direct cost		4340	4590	5310	5880			
Indirect cost		8000	7000	6200	5600			
Total cost		12340	11590	11510	11480			

* Critical activities
\- Activity can not be crashed any further

Step 7: Path II is critical path;

- No Critical Activities can be crashed as (from Cost sheet)
- E, C and A have: $\Delta_E = \Delta_C = \Delta_A = 0$;
- No activity from critical path could be crashed. Economic Crash limit is obtained
- Stop.

Solution: Least cost schedule = 56 days and Total minimum cost = $ 11,480, 000
Crashing needed:

Activity	Crashed by Days	Crash Cost ($)
A	6	570,000
C	8	720, 000
E	10	250,000

7. CONCLUSION:

The new algorithm has given the solution for test problem in *three iterations*. This solution by Unit Time Method would

require 10+8+6 = 24 iterations. It requires fewer efforts for manual solutions. Definitely this test problem can not be solved manually (24 iterations!) while with our algorithm it is possible.

1. New algorithm requires fewer iterations (12% or 3/24 in test problem)
2. It quickly gives the optimum solution
3. It finds application for almost all problems in CPM
4. Definitely this is great addition in literature
5. CTL Algorithm has been framed for more critical paths.
6. Its application to PERT or stochastic times is still area for further research
7. Its application when more than five or more critical paths exits; is also an area for further research

REFERENCES:

1. Arisawa, S. and Elmaghram, S. E. (1979), Optimal time-cost trade-off in GERT Networks. Management Science, 8(11): 589-599.
2. Hillier, F. S., M. S. Hillier and G. J. Lieberman, (2000), Introduction to Management Science, Irwin-McGraw-Hill, Boston,
3. Gupta, P.K. and Hira D.S. (2006), "Network Analysis in Network Planning"; Operations Research; book, pp 1141-1244
4. Liu, L., Burns, S. and Feng, C. (1995b) Construction Time-Cost Trade-off Optimization System, Proceedings: The Second Congress held in Conjunction with A/E/C Systems '95, Atlanta, GA, ASCE, Vol. 2, 5-8 June, pp. 1142-1149.
5. Reda et al. (1989), "Time-Cost Trade-off Among Related Activities, Journal of Construction Engineering and Management, ASCE, Vol. 115, No. 3, pp. 475-486.
6. Robinson, D.R. (1975), A Dynamic Programming Solution to Cost-Time Trade-off for CPM, Management Science, Vol. 22, No. 2, pp. 158-166.
7. Sakellaropoulos et al (2004), "Project time-cost analysis under generalized precedence relations". Advances in Engineering Software, 35; pp 715-724.
8. Senouci A.B. and Eldin N.N. (1996), "A time-cost trade-off algorithm for non-serial linear projects", Canadian J. of Engineering, Vol.23, pp.134-149.
9. Siemens, N. (1971) "A Simple CPM Time-Cost Trade-off Algorithm", MS, Vol. 17. No. 6, pp. B-354-363.
10. Smith, L.A. (1997) , "Comparing Commercially Available CPM/PERT Computer Programs", Journal of IE, Vol. 10, N°4,
11. Stevens, J.D. (1996), "Techniques for Construction Network Scheduling",
12. Talbot, F.B. (1982), "Resource Constrained Project Scheduling with Time-Resource Tradeoffs:" Management Science, Vol. 28, pp 1197-1210.
13. Vora, N. D. (2005), "PERT and CPM"; Quantitative techniques in management; book, pg 537-596