Unit based Scheduling in Project Management: A Programming Approach

D.Anuradha
Department of Mathematics,
School of Advanced Sciences
VIT University, Vellore-14,
Tamil Nadu, India

S.Bhavani
School of Information Technology
and Engineering
VIT University, Vellore-14,
Tamil Nadu, India

ABSTRACT

A unit based scheduling technique is proposed to assess networks of the given project and find out the optimum schedule. The proposed approach uses an iterative method to execute unit expediting and the expedition is done to the desired amount for all activities along the critical path. The output of the proposed approach includes a distribution of the total cost of project, duration of project completion and it is mainly applicable where cost and duration are of major issues. A computer program has also been developed using the C programming language to support the efficiency of the logic applied in the algorithm. A civil construction scheduling has been solved illustrating the validity of the proposed algorithm. This algorithm can serve as an important tool for the project managers to run the projects speedily and successfully.

Keywords

Project Network, Critical path, Incremental cost, Expediting.

1. INTRODUCTION

Project management is the field of planning, scheduling, and resource management to bring out the successful windup of particular objectives and goals on one-time endeavor such as, a building contract or other project activities that require to apply a scientifically based method implementations. This fundamentally admits the formulating project plan, defining goals and objectives of project, specifying and finding out the project completion duration. It also admits implementing the plan of project, along with careful controls to remain on the "critical path", that is, to make sure that the project is being managed according to plan. Critical Path Method (CPM) is a procedure for process scheduling that defines critical and non-critical tasks with the objective of preventing duration-frame problems. It is an essential tool for the project management to determine the critical activities in the critical path of an activity network.

The expedition of an activity is determined as the maximum duration to which an activity can be constricted and yet is economic. The expediting approach is concentrated on reducing the activities duration on the critical path. Since the critical path finds out the project completion date, the project can be speeded up by adding the necessary resources to diminish the activities duration in the critical path. Such a project shortening is termed as project expediting. Ramini [11] also proposed an algorithm for crashing PERT networks incorporating the use of criticality indices. Ameen [1] developed Computer Assisted PERT Simulation, a simulation program developed as a teaching tool to teach project management techniques. Badiru [2] reported development of another simulation program for project management, called STARC. Pulat and Horn [10] described a project network with a set of tasks to be completed according to some precedence relationship; the objective is to determine efficient project schedules for a range of project realization times and resource cost per time unit for each resource. Bissiri and Dunbar [14] discussed a simulation model that allocates resources to project activities in a way so as to minimize the additional cost of resources. Feng et al. [4] presented a hybrid approach that combines simulation techniques with a genetic algorithm to solve the time cost trade-off problem under uncertainty. Haga and Marold [6] proposed a simulation based method that deals with the time-cost trade-off involved with crashing a project. Chao-gunag et al. [8] proposed a fully fuzzy time-cost trade-off based on genetic algorithms. Haga and Marold [7] developed a heuristic crashing method for project management utilizing simulation. Yang [13] incorporated budget uncertainty into project time-cost trade-off in a chance constraint programming model. Ghazanfari et al. [5] developed a new possibilistic model. Michael E. Kuhl [9] introduced a dynamic simulation-based crashing method to evaluate project networks. Yousefli et al. [15] presented a heuristic method to solve a project scheduling problem by using fuzzy decision making in fuzzy environment. Dishi Xu and Hua [3] designed a crashing algorithm in the applications of project management. Shakeela and Ganesan [12] proposed a method for finding an optimal duration by crashing the fuzzy activities of a project network without converting the fuzzy activity times to classical numbers.

In this paper, we propose a unit based scheduling technique to evaluate project networks and determine the optimum project schedule. The proposed method mainly provides a framework for expediting total maintenance of project duration at the least total cost. The output of this approach includes a distribution of the total cost of project, duration of project completion and the savings of project cost. The paper is organized as follows. In section 2, we discuss some relevant concept of project scheduling techniques. In section 3 we describe our proposed algorithm. To show the application of proposed algorithm, a civil construction scheduling is solved in section 4. The results and its graphical representation are discussed in section 5. At last in section 6 some conclusions are drawn.

2. PROJECT SCHEDULING TECHNIQUES

Execution and completion of any project either tiny or big is becoming highly complex due to so many constraints, especially duration and resource. Hence, the field of 'Project Management' is demanding effective and efficient techniques to optimize the project results. In many situations we may be interested in finding the least possible project completion time

if expediting of activities is permitted. The process of reducing the duration of an activity by spending more resources is called as Expediting. When activities are expedited, it is obvious that more resources to be spent. The total cost of any project comprises direct cost and indirect cost. Direct cost are those which are associated with the individual activities such as materials consumed, equipment used and so on whereas indirect costs are associated with overhead expenses such as managerial services, indirect supplies, cost of security personnel and so on. While expediting an activity there is a lower limit beyond which it is not possible to reduce its time any more. This is called expedite limit of that activity. So each and every activity will have two duration estimates, viz., Normal Duration (ND) and Expedite Duration (ED). Normal duration is the duration taken to execute an activity under normal circumstances. Expedite duration is the minimum duration of an activity beyond which it is not possible to reduce it anymore. The cost associated with the ND is called Normal Cost (NC) and the cost associated with the ED is called Expedite Cost (EC).

Complex projects involve a sequence of activities, some of which must be carried out consecutively and others that can be carried out parallely with other activities. This accumulation of series and parallel tasks can be patterned as network. The network diagram can be used to identify the activities whose duration should be shortened so that the completion time of the project can be shortened in the most economic manner. This procedure is called as Network Expediting. It is true that due date for completion of a project may well affect the cost incurred, because more resources are required to perform work in a shorter period of time. This is called 'time–cost trade–off analysis'.

Fig 1 indicates the duration cost relationship and Fig 2 indicates the total cost of the project which is the addition of the direct cost and indirect cost of the project.
Now we propose a new algorithm for finding an optimum schedule to the unit based scheduling.

3. UNIT BASED SCHEDULING ALGORITHM

The following algorithm focuses on expediting the activities in a given project network day by day instead of expediting it completely. For a given network, find the possible paths to reach the destination. The incremental cost for all the activities are found and sorted in an ascending order. Then the expedition is started in the critical path's activity with the lower incremental cost. The project's total cost is calculated for every critical path after expedition. According to the result the network is redrawn. The process continues until all the activities are expedited to its maximum level specified. Fig 3 depicts the flow chart of the proposed algorithm.

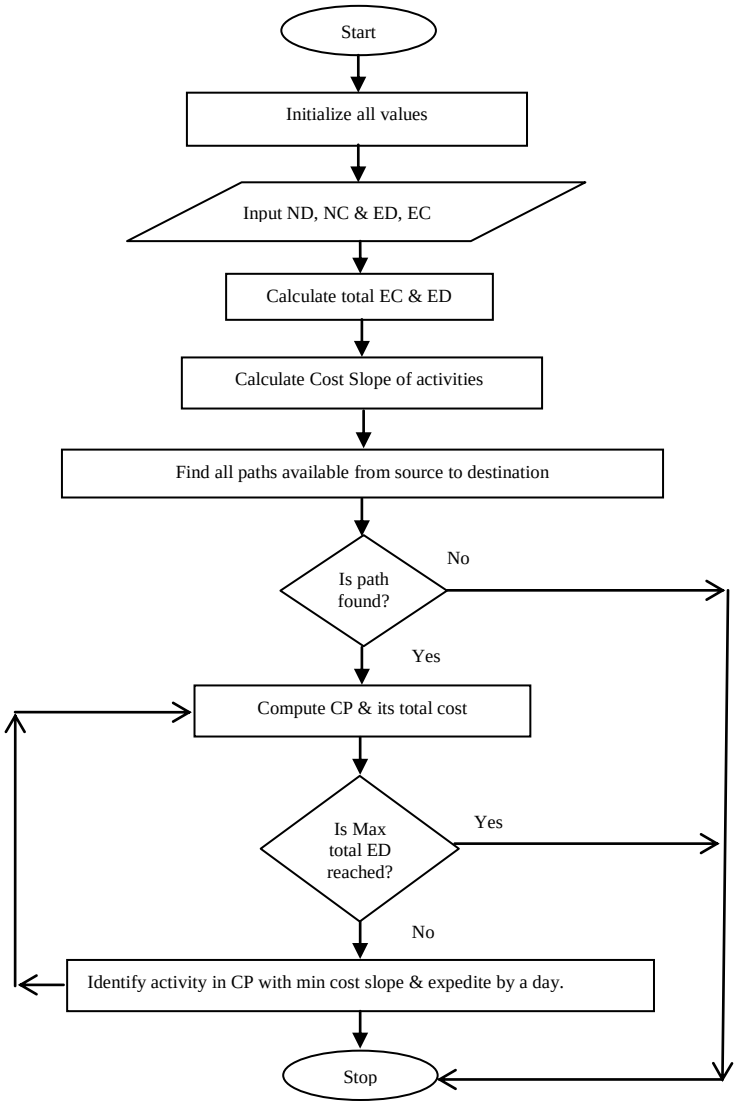

Fig 3 Flowchart

The proposed algorithm proceeds as follows:

Step1:
Initialize all values.

Step2:
Calculate total expedite duration, total expedite cost, total expedite cost slope by using the following criteria.
a[i].totExpdur[j]=a[i].normDur[j]-a[i].expDur[j];
a[i].totExpcost[j]=a[i].expCost[j]-a[i].normCost[j];
a[i].expCostslope[j]=a[i].totExpcost[j]/a[i].totExpdur[j];

Step3:
Find all the paths available from source to destination;

```
void path(int p,int queue[],int rear)   {
                        if(a[p].n!=0)
                        q[rear++]=a[p].val;
                        for(i=0;i<a[p].n;i++)
                        {
                        if(a[p].ad[i]==d)
                        {
                        q[rear++]=d;
                        for(k=0;k<rear;k++)
                        op[inc][k]=q[k];
                        size[inc]=rear;
                        inc++;
                        }
                        else
                        {
                        path(findposition(a[p].ad[i]),q,rear);
                        }}}
```

Step4:
Find the critical paths.

Step5:
Find the minimum expedite Cost Slope.

```
minimum=a[0].expCostslope[0];

if((minimum>a[i].expCostslope[j])&&(a[i].expCostslope[j]!=0))
  minimum=a[i].expCostslope[j];
```

Step6:
Path which has the minimum expedite Cost Slope that activity is chosen for expediting.

```
if(a[i].expCostslope[j]==minimum)
        expediteActivity(i,j);
```

Step 7:
Expedite the given activity by calling the Expedite Activity function. The Expedite Activity function is as follows;

```
void expediteActivity(int a1,int b1)  {
while(true)      {
Go to Step 8
Go to Step 9
Go to Step 10
If( no.of  current critical paths < no of prev critical paths)
Terminate and exit from the loop;
}}
```

Step 8:
Find the Common Activities to Expedite using findCommonPath();

```
flag=0;
for(i=0;i<cv;i++){
for(j=0;j<sizec[i]-1;j++){
 ta[tac].so=cre[i][j];
 ta[tac].de=cre[i][j+1];
 tac++;          }}
for(i=0;i<tac;i++)
{
temps=ta[i].so;
tempd=ta[i].de;
noc=0;
for(j=0;j<tac;j++){
if((ta[j].so==temps)&&(ta[j].de==tempd)){
noc++; }}
  ps[sc].sou=temps;
  ps[psc].des=tempd;
  ps[psc].nc=noc;
  psc++;   }
  cac=0;
for(k=0;k<psc;k++)   {
if((ps[k].nc==cv)&&(getTotalExpediteDur(ps[k].sou,ps[k].des)!=0))  {
   printf("\nCommonPath:%d->%d",ps[k].sou,ps[k].des);
   ca[cac].sour=getsPosition(ps[k].sou,ps[k].des);
   ca[cac].dest=getdPosition(ps[k].sou,ps[k].des);
   flag=1;      }}
if(flag==0)   {
for(k=0;k<psc;k++) {
 if((ps[k].nc==cv-1)&&(getTotalExpediteDur(ps[k].sou,ps[k].des)!=0))  {
   printf("\nCommonPath:%d->%d",ps[k].sou,ps[k].des);
   ca[cac].sour=getsPosition(ps[k].sou,ps[k].des);
   ca[cac].dest=getdPosition(ps[k].sou,ps[k].des);
   flag=1;
   index=indexOfProblematicCP(ps[k].sou,ps[k].des);
 for(l=0;l<sizec[index]-1;l++)     {
   ts=cre[index][l];
   td=cre[index][l+1];
   if(IsnotPresentInOtherCPath(ts,td,index))    {
   printf("\nCommonPath:%d->%d",ts,td);
   cac++;
   ca[cac].sour=getsPosition(ts,td);
   ca[cac].dest=getdPosition(ts,td);
   goto x;     }     }
   goto x;   } } }
if(flag==0)  {
        for(k=0;k<psc;k++)   {
         if((ps[k].nc==cv-2)&&(getTotalExpediteDur(ps[k].sou,ps[k].des)!=0))  {
          printf("\nCommonPath:%d->%d",ps[k].sou,ps[k].des);
          ca[cac].sour=getsPosition(ps[k].sou,ps[k].des);
          ca[cac].dest=getdPosition(ps[k].sou,ps[k].des);
for(i=0;i<cv;i++) {
              tf=0;
for(j=0;j<sizec[i]-1;j++) {
          if((cre[i][j]==ps[k].sou)&&(cre[i][j+1]==ps[k].des))  {
          tf++;            } }
          if(tf==0)    {
                ind[ins]=i;
                ins++;           } }
         ix=0;
          for(i=0;i<ins;i++)       {
          for(j=0;j<sizec[ind[i]]-1;j++)  {
          se[ix]=cre[ind[i]][j];
          dn[ix]=cre[ind[i]][j+1];
          ix++;  }  }
          int h,helpflag;
          helpflag=0;
```

```
        for(i=0;i<ix;i++)     {
            h=0;
        for(j=0;j<ix;j++)      {
if(((se[i]==se[j])&&(dn[i]==dn[j]))&&(getTotalExpediteDur(
se[i],dn[i])>0))
                    h++;       }
        if(h==2) {
             cac++;
            helpflag++;
            flag=1;
            printf("\nCommonPath:%d->%d",se[i],dn[i]);
            ca[cac].sour=getsPosition(se[i],dn[i]);
            ca[cac].dest=getdPosition(se[i],dn[i]);
            goto x; } }
        if(helpflag==0) {
                int ts,td,ijk;
                for(i=0;i<ins;i++) {
                ijk=0;
                for(j=0;j<sizec[ind[i]]-1;j++) {
   ts=cre[ind[i]][j];
   td=cre[ind[i]][j+1];
if((IsnotPresentInOtherCPath(ts,td,ind[i]))&&(getTotalExpedi
teDur(ts,td)>0)) {
   if(ijk==0)
   cac++;
    printf("\nCommonPath:%d->%d",ts,td);
   ca[cac].sour=getsPosition(ts,td);
   ca[cac].dest=getdPosition(ts,td);
   ijk++;    } }
   ws:
   continue; } }
          goto x; } } }
          x:
    ;
```

Step 9:
Expedite the activity by calling ExpediteTheCorrectActivity();

```
void expediteTheCorrectActivity(int sl,int dl)  {
a[s1].totExpDur[d1]--;
a[s1].normDur[d1]--; }
```

Step 10:
Calculate the critical Path by calling Critical();

```
Void critical() {
max=sum(0);
for(i=0;i<5;i++) {
                if(sum(i)>max) {
                max=sum(i);
                posi=i; } }
                for(j=0;j<5;j++) {
                                if(max==sum(j))
Take the path as Critical Path; } }  }
```

Step 11: End.

4. ILLUSTRATIONS

The proposed unit based scheduling algorithm for finding the optimum project schedule is illustrated by the following example.

Fig 4 depicts the activities involved in the construction of a house. The work commences with the study of plant layout and the clearance of site. The normal duration, its respective expedite duration is to be depicted. For the normal and expedite cost of all the activities, the feasible expedite value for the cost and duration will be estimated. The construction work takes different tracks for its completion.

The activities involved depend on the previous activities for its commencement. The earthwork starts that will lead to the laying of foundation, after the completion of the study on plant layout. Meanwhile the construction materials such as bricks, sand, cement and concrete are procured. The drainage and sewage systems are laid parallel with the foundations. The other materials namely pipelines for electric wires and water pipes are procured and also will be laid. The building constructed after all basic foundations that is followed by connecting the building to water and electricity. Finally, the finishing work of the building will be performed with the indirect cost of ₹1000 per day.

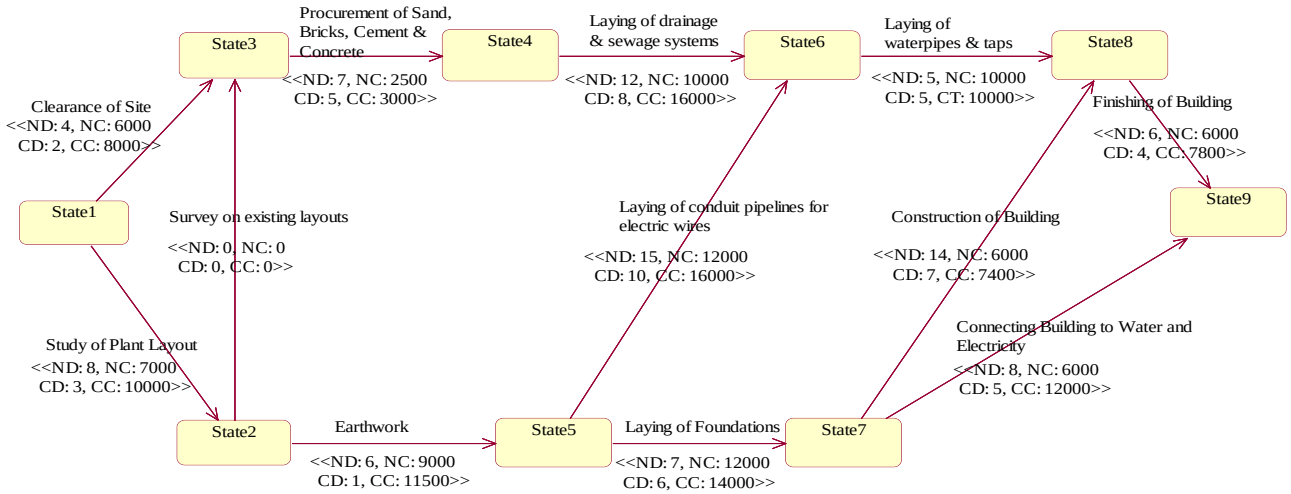

Fig 4 Construction Network

5. RESULTS AND DISCUSSIONS

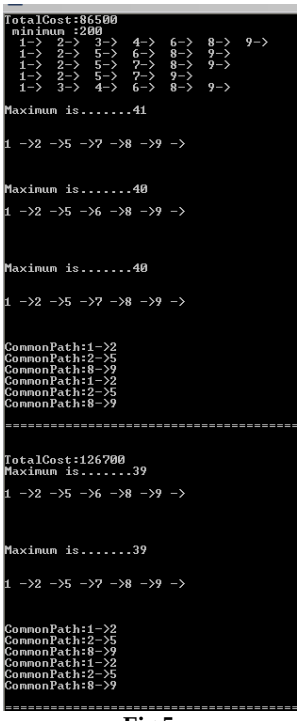

Fig 5

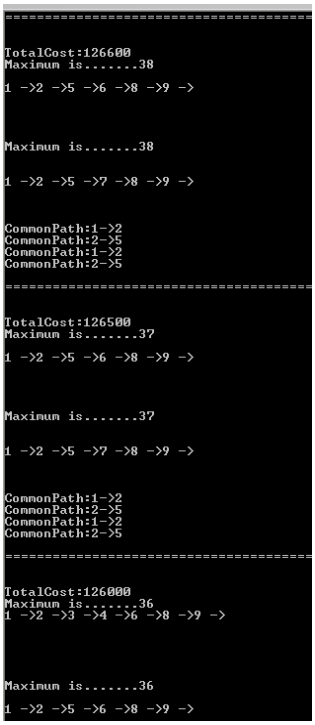

Fig 6

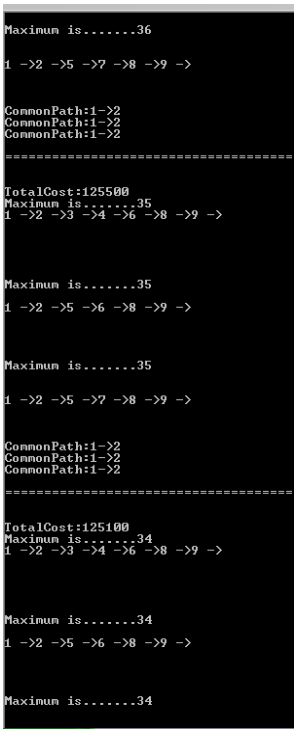

Fig 7

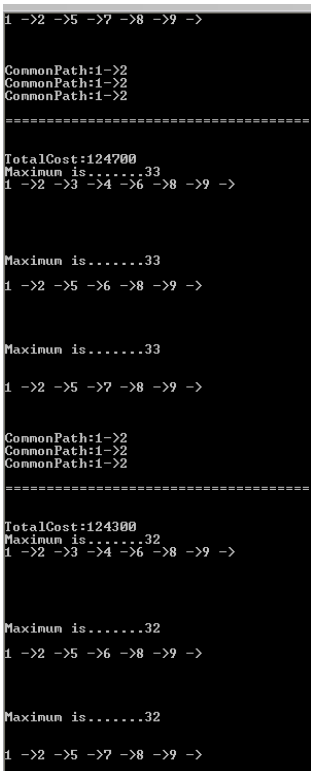

Fig 8

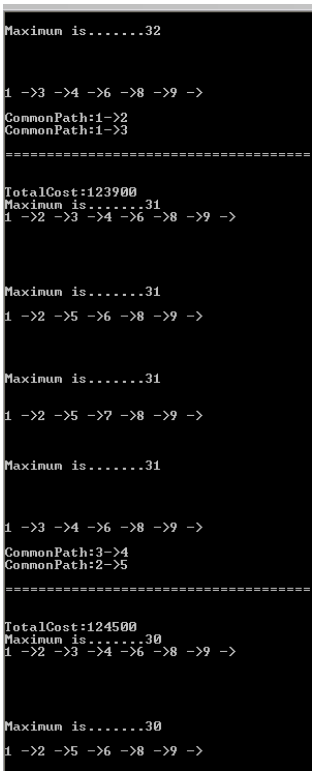

Fig 9

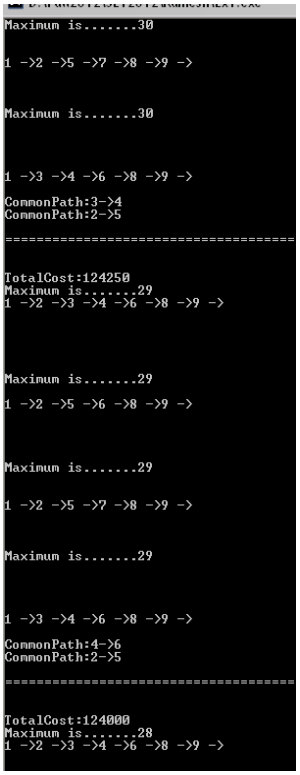

Fig 10

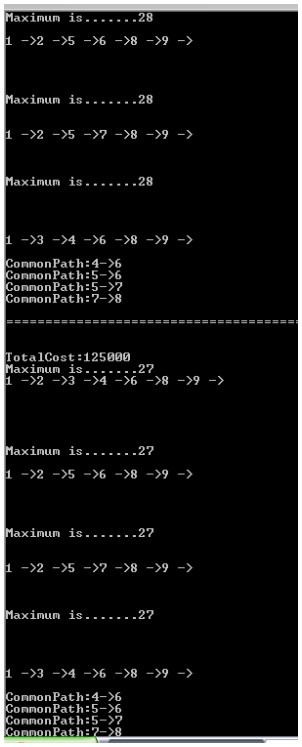

Fig 11

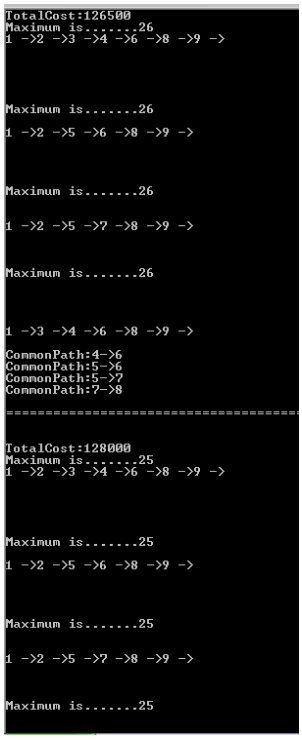

Fig 12

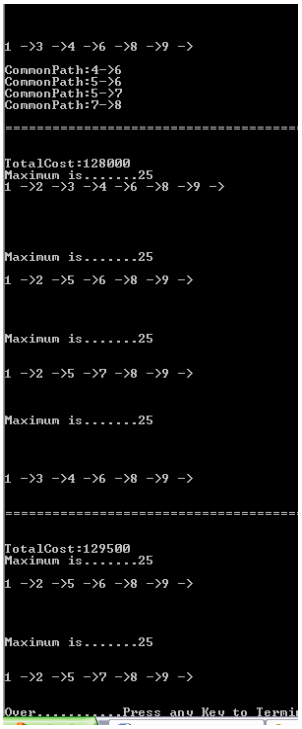

Fig 13

For the given project network from fig 4, we observe that the critical path is 1→2→5→7→8→9. The project duration of the critical path is 41 days with the corresponding total cost of ₹ 127500. By applying the proposed unit based scheduling algorithm, we obtain four critical paths in the given network after 16 expeditions. They are C1:1→2→3→4→6→8→9, C2:1→2→5→6→8→9, C3:1→2→5→7→8→9 and C4: 1→3→4→6→8→9 respectively with the project duration 25 days with the corresponding total cost of ₹ 129500.

5.1. Graphical Representation

The fig 14 portrays the results obtained by the proposed algorithm, namely unit based scheduling algorithm of a given project network.

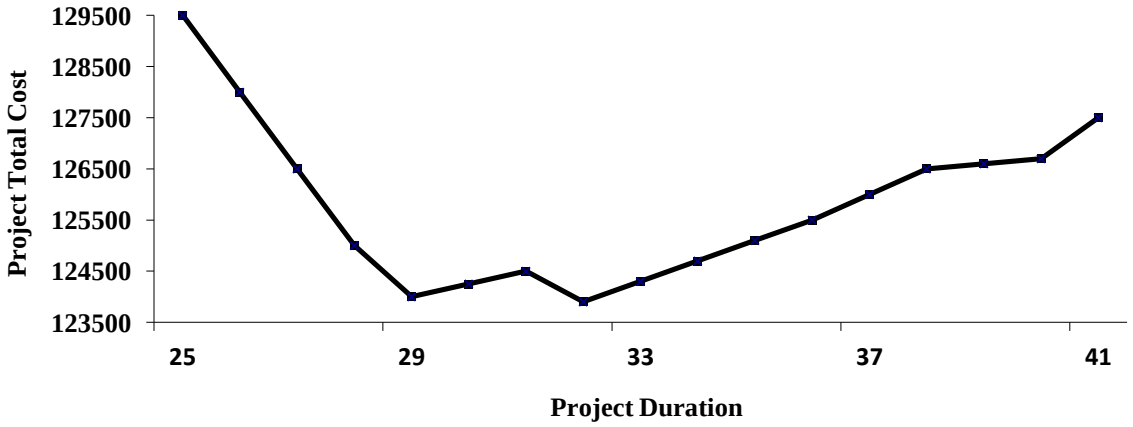

Fig 14

From the fig 14, we notice that the expected duration of the project was successfully reduced from 41 days to 25 days upon the increase in the amount of money from ₹127500 to ₹129500 to the project.

6. CONCLUSIONS

This paper investigated the unit based scheduling algorithm in the project management. The validity of the proposed algorithm is examined with the illustration. The project manager can scrutinize the effect on total cost of changing the expected project duration to different alternative values. We can consider this approach faster and easier to complete the project in shortest possible duration. By using the proposed approach, the project manager will become competent to find out how much to be expedited in each activity and thus can minimize the total cost of meeting any defined deadline for the project.

7. ACKNOWLEDGMENTS

The authors are thankful to the Managing Editor and the reviewers whose suggestions improved the version of the paper.

8. REFERENCES

[1] Ameen, D.A. 1987. A computer assisted PERT simulation, J. Syst. Management, 38, pp.6-9.

[2] Badiru, A. B. 1991. A simulation approach to network analysis, Simulation, 57, pp. 245- 255.

[3] Dishi Xu and Xianggang Hua .2011. The Applications of Crashing Algorithm in Project Management, IEEE, pp. 349-354.

[4] Feng, C.W.,L.Liu and S.A.Burns .2000. Stochastic construction time-cost trade-off analysis, Journal Comput. Civil Engg., 14, pp.117-126.

[5] Ghazanfari M, Shahanaghi K, Yousefli K A .2008. An Application of possibility goal programming to the time-cost trade-off problem, JUS, World academic press, 2, pp.22-28.

[6] Haga, W. A., and K. A. Marold .2004. A simulation approach to the PERT CPM time-cost trade-off problem, Project Management Journal, 35, pp. 31-37.

[7] Haga, W. A., and K. A. Marold .2005. Monitoring and control of PERT networks, The Business Review, 3, pp. 240-245.

[8] JIN Chao-guang, JI Zhuo-shang, LINYan Zhao Yuan-min, Zhen-dong .2005. Research on the fully fuzzy time-cost trade-off based on genetic algorithms, Journal of Marine Science and Application, 4, pp.18-23.

[9] Michael E.Kuhl .2008. A dynamic crashing method for project management using simulation based optimization, Proceedings of the 2008 Winter Simulation Conference, pp. 2370-2376.

[10] Pulat, P.S and S.J.Horn .1996. Time-Resource trade-off problem (project scheduling), IEEE Trans. Eng. Management, 43, pp.411-417.

[11] Ramini S. 1986. A simulation approach to time-cost trade-off in project network, Modeling and Simulation on Microcomputers, Proceedings of the Conference, pp.115-120.

[12] Shakeela Sathish and K.Ganesan .2012. Fully fuzzy time-cost trade-off in a project network- A new approach, Mathematical Theory and Modeling, 2, pp.53-65.

[13] Yang T .2007. Impact of budget uncertainty on project time-cost trade-off, Eng Manage, 52, pp.167-174.

[14] Yassiah Bissiri and Scott Dunbar .1999. Resource Allocation Model for a Fast-Tracked Project, Intelligent Processing and Manufacturing of Materials, 1, pp.635-640.

[15] Yousefli K A, Ghazanfari M, Shahanaghi K, Heydari M .2009. A new heuristic model for fully fuzzy project scheduling, JUS, World academic press, 2, pp.73-78.