Decision Support Tool for Project Time-Cost Trade-off

Fikri Dweiri
Industrial Engineering and Engineering Management Department
University of Sharjah
Sharjah, UAE, 27272
fdweiri@sharjah.ac.ae

Abstract

Shortening the project duration (crashing of projects) sometimes is necessary function of project management. This is achieved by applying additional resources to the project activities. Benefits of crashing may be in the form of specified rewards in the contract for completing the project ahead of time or transferring resources to other projects. There are two basic methods for performing project crashing: The first method is the linear programming method. It offers an excellent optimization tool when the relationship between cost and time reduction is linear. However, using this method requires project managers to have good knowledge of optimization. The second method is the Critical Path Method (CPM) that is usually used for manual crashing of small networks. However, this method becomes cumbersome when many parallel critical paths exist. The objective of this research is to develop decision support tool that automates the CPM steps of project crashing. The developed tool is easy to use, windows oriented, and it has a wide range of capabilities such as: file saving, retrieving, modifying, and reporting. Moreover, the tool allows crashing projects that has a large number of activities. The tool was validated using several small and large project examples.

Keywords
Decision Support tool, project crashing, project management, linear programming, critical path method, project scheduling.

1. INTRODUCTION

Many of the most engineering challenges in recent decades are to implement large projects such as the development of manned space programs and the construction of vast petroleum production facilities in the North Sea off the shore of Great Britain (Shtub). The implementation of such projects with high performance capabilities and within acceptable schedules and budgets has required the development of new methods of planning, scheduling, organizing, and controlling. A project involves the application of knowledge and skills, tools, and techniques to project activities in order to meet stockholders needs and expectations from a project (Project Management Institute 1996). Project progress and outcomes depend on technology, time, cost, profit, resource utilization and market acceptance (Shtub). Project effectiveness is measured in terms of the degree of achievement of the project objectives (Belout; Baccarini; and Shenar). Project management is a complex task that includes planning, organizing, staffing, controlling, monitoring, and directing functions to the successful completion of the project (Elsayed).

Good scheduling helps achieving the project management functions optimally since it is considered as the cornerstone of these functions. Scheduling activity integrates information on activity duration, the precedence relationships among activities, resources and budgeting constraints, and due-dates. Scheduling is usually performed by graphically depicting the activities and their relationships. The Critical Path Method (CPM) was developed in 1959 by the DuPont Company for achieving better scheduling of a chemical plant construction (Kelly). Program Evaluation and Review Technique (PERT) was also developed in 1959 by Booz, Allen, and Hamilton (management consultants) and Lockheed Aircraft Corporation for the use in the development of the Polaris ballistic missile (Kerzner). Both CPM and PERT are graphical methods that are based on network representations of a project by arrows and nodes. CPM is considered as a deterministic diagraphic method while PERT is considered as a probabilistic diagraphic method (Elsayed). CPM and PERT networks are integral parts of project management and they provide benefits such as identification of the critical path(s) activities, which if delayed, the project completion will be delayed, and if shortened, the project completion will be shortened (Shtub).

Shortening the project duration may sometimes be the objective of project management. This is achieved by applying additional resources to the project activities. This process is called project crashing. Benefits of crashing may be in the form of specified rewards in the contract for completing the project ahead of time (Babcock) or transferring resources to other projects. It was reported that Lockheed's contract for C-5A aircraft contained a $12,000 per day penal clause and General Dynamics was rewarded $800,000 for flying the F-16 aircraft 10 days ahead of schedule (Elsayed).

Crashing projects may be realized by two basic methods (Gordon): The linear programming (LP) model, and the critical path analysis method. The former method requires the manager to have the knowledge of operations research to be able to formulate the problem in the required format and to interpret the solution. It also requires LP software with large memory to perform crashing of projects that have moderately large number of activities. The latter method is based on manual crashing of the critical paths of the network. It is well suited for small networks where not many critical paths are present. This method becomes tedious and more complex if the number of activities of the project is moderately large.

The objective of this research is to develop user-friendly decision support tool that automates the steps of the critical path method for project crashing. The intended tool is deals with any reasonable large number of activities. Dealing with probabilistic model (PERT) is beyond the scope of this paper. To the authors' knowledge and based on an internet search, the available project management software packages allow the user to perform project crashing by manually adding resources to the activity or activities on the critical path(s). The crashing process here may be considered as manual crashing of projects even though it is performed on computers because the manager chooses the activity to crash. Hence, the project manager might select by mistake to crash some critical activity that does not have the minimum cost increase per day.

This paper is structured as in the following: The next section provides some background on the available tools on project management. The methodology of the research including the algorithm of the tool is then presented. After that, a problem from a book is borrowed and solved using the LP method, the manual critical path method, and finally using the developed tool. A contrast between the three methods is then presented. The section before the last outlines the features of the proposed tool. The last section presents summery and conclusions of the research.

2. SOFTWARE IN PROJECT MANAGEMENT

Project management software packages have been developed to aid the project manager with the establishment, monitoring, revision of the project schedule, and project resource planning. Over 35 project management software packages are available with prices ranging from $250 for Quick-Plan II to about $20,000 for Qwiknet professional (Fawcette; Shtub). Microsoft Project and Primavera Project Planner 4.0 are the most popular packages. These software packages have many capabilities (Corder) such as: resource leveling to avoid conflict of resources, graphical displays (GANTT and PERT), use of part-time employees, realized cost versus predicted cost reports, and linking multiple projects together. These packages require the knowledge of related project management tools (GANTT and CPM/PERT) before they could be used (Sivasankaran).

Key measures for evaluating project management software packages (Corder) are: performance, ease of learning, ease of use, documentation, and user anxiety of losing the results of a session through ignorance, carelessness, or other problems such as disk full. The available project management software packages allow the user to perform project crashing by manually adding resources to the activity or activities on the critical path(s). This process is cumbersome, especially for larger projects where multiple critical paths may be present. The crashing process here may be considered as manual crashing of projects even though it is performed on computers. Crashing may be performed on software packages that are not project management oriented such as linear or integer programming where the operations research (OR) knowledge is a pre-requisite by the user.

3. METHODOLOGY

In many projects, management is willing to allocate additional resources (money, manpower, equipment, etc.) in order to reduce the project duration. Project contracts may contain specific clauses for penalties for delays beyond the specified project due-date, and it may specify rewards for project completion ahead of the due-date (Babcock). The objectives of optimal allocation of additional resources are the maximum reduction in duration for a given budget and the lowest possible cost for a given reduction in project duration (Gordon). As depicted in Fig. (1), the normal time (r) originally estimated for the activity is the time associated with the lowest cost (u) to complete the activity.

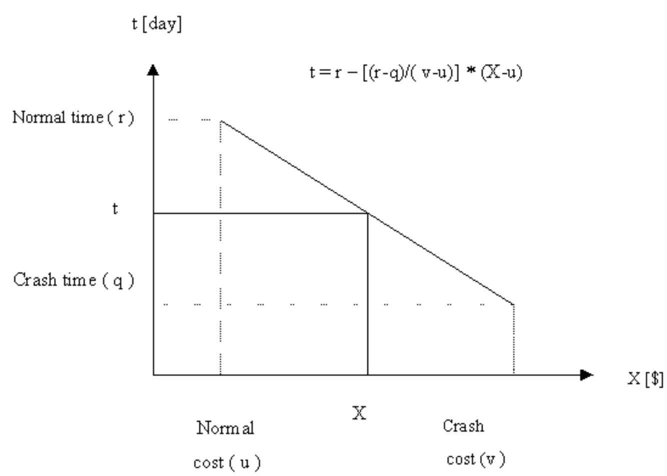

Fig. 1: Typical time-cost trade-off curve

For crashable activities, there will be some shorter (crash) time (q) that can be achieved at a higher (crash) cost (v) by using overtime, larger crews, more expensive equipment, or subcontractors. In this paper, the assumption is that the use of overtime, larger crews, new or more equipment, or subcontractors are translated into an increase in the activity completion cost (crashing cost). Assuming a linear relationship between crashing cost and crashing duration then the crash cost per unit time of an activity can be calculated as in the following equation (Gordon, 1990):

$$C = (v\text{-}u)/(r\text{-}q) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

When crashing, managers select to crash first the critical activity or activities with the lowest crashing cost per unit time. Budgeting decisions are easier when the time–cost relationship is known (Shtub).

It is intended in this research to develop a user-friendly decision support tool that automates the steps of the critical path method for project crashing. The algorithm used for the development of the new tool is the algorithm presented by Elsayed. Fig. (2) illustrates the flowchart of the algorithm. It is simply based on the following steps:

1) Enter the following data for each activity of the network: normal time, normal cost, and preceding activities. If the activity is crashable, enter its crashing time and its crashing cost.
2) Compute the crashing cost per day (C) for each crashable activity as in equation 1.
3) Find the critical path(s) of the network based on the forward and backward pass techniques.
4) Find the normal cost of the schedule as the sum of the normal costs of all activities.

5) Find crashing alternatives of the network. If the network has one critical path, each activity is a candidate (alternative) for crashing. If the network has more than one critical path, crashing an activity from only one critical path does not reduce the project completion time. It only creates slack time on the uncrashed path(s). Thus, an activity from each critical path must be crashed to realize reduction in the project completion time. Note that a common activity among all paths may be crashed which results in reducing all critical paths times and thus reducing the project completion time without creating slack time. If a critical activity is uncrashable, it is not considered as part of the alternatives.
6) Find the total crashing cost per day for each crashing alternative.
7) Select the crashing alternative that has the lowest total crashing cost and reduce the length of each activity in this alternative by one day. This step guarantees that all current critical paths are crashed simultaneously by one day and hence the length of the project is crashed by only one day.
8) Continue crashing the project each time by one day until an uncrashable critical path is found.

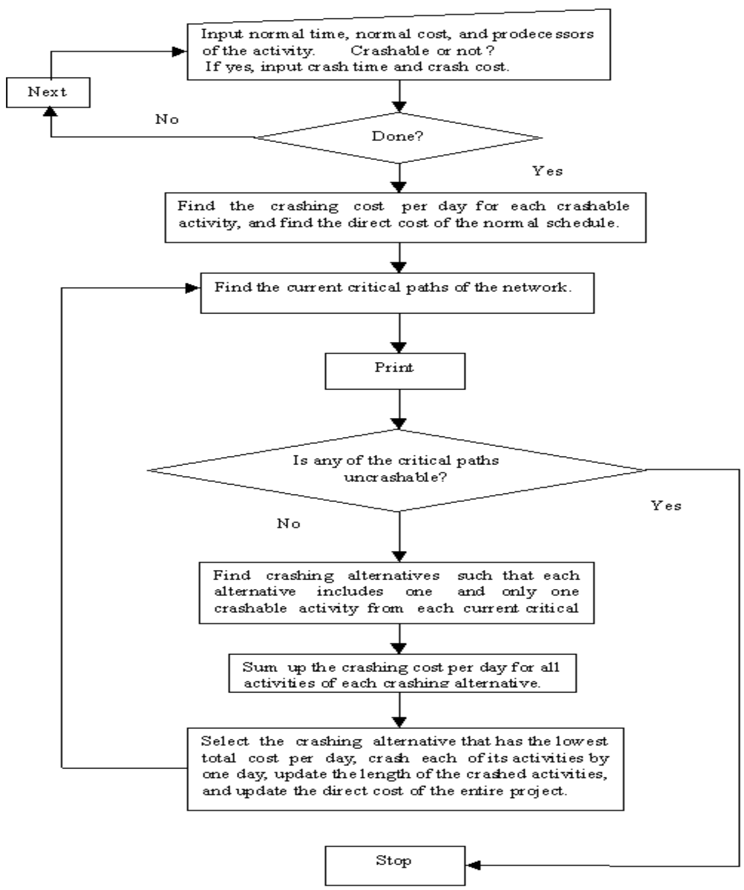

Fig. 2: Flow chart for the crashing algorithm used in the proposed decision support tool

4. COMPARING CRASHING APPROACHES

To illustrate the different crashing approaches, a problem is selected from (Elsayed). In the following sub-sections, the problem is solved using the linear programming method, the manual critical path method, and finally using the proposed decision support tool. This section provides a comparison between the three methods. Moreover, it provides some validation for the proposed decision support tool.

4.1 Statement of the Problem

The data of the problem is presented in Table 1. The graph of the network for the problem is shown in Fig. 3. In this problem it is required to find the shortest project duration (crash the project) for the least possible budget. In the following, the problem will be solved using linear programming method, the manual critical path method, and finally using the developed tool.

Table 1: Data for the illustrative problem of the project (Elsayed)

Activity	Code	Pre-act.	Normal Time (r) [days]	Normal cost (u) [$]	Crash time (q) [days]	Crash cost (v) [$]	Crashing Cost per day (C)
A	1	---	10	180	8	200	10
B	2	---	12	400	11	450	50
C	3	---	7	500	7	500	---
D	4	3	4	160	3	200	40
E	5	3	8	80	6	100	10
F	6	1	12	500	11	600	100
G	7	1,2,4,5	9	350	9	350	---
H	8	3	16	610	13	700	30
			Sum	$2,780	Sum	$3,100	

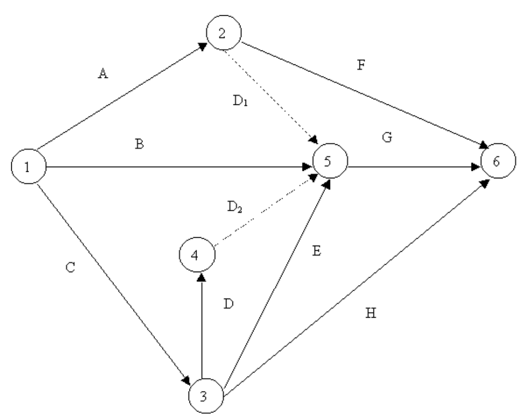

Fig. 3: Network representation of the project for the illustrative problem

4.2 The Linear Programming Solution

Linear programming can be used as an optimization tool when the relationship between cost and time reduction of the activities is linear as shown in Fig. (1). Let us assume that we have an available budget (M) that we want to spend on crashing. For an activity i, let us use the following notation for the formulation of the resource allocation problem as a linear programming model:

u_i = minimum cost of activity i.
v_i = maximum cost of activity i.

q_i = minimum duration of an activity i.
r_i = maximum duration of an activity i.
X_i = resource allocation to activity i.
t_i = expected duration of activity i after it gets the allocation X_i.
C_i = cost of one day reduction of activity i.

Now we want to find optimal values of X_A, X_B, X_C, etc. such that the project is crashed as much as possible within the budget M (Gordon, 1990). The following equation can be derived from Fig. (1):

$$t_i = r_i - (1/C_i) * (X_i - u_i) \quad (2)$$

For an activity i, equation 2 represents a direct relationship between X_i and t_i. Let us consider the graph of the network in Fig. (3), and introduce the following notation:
T_j = earliest expected occurrence time of event j, then for an activity like A in Fig. (3), we require that (Gordon):

$$T_2 - T_1 \geq t_A = r_A - (1/C_A) * (X_A - u_A) \quad (3)$$

Hence it follows that:

$$C_A * T_2 - C_A * T_1 + X_A \geq r_A * C_A + u_A \quad (4)$$

This analysis can be generalized for the other activities and the following linear programming model can be written for the resource allocation problem at hand as below:
Model:
$Min=T_6$;
$10*T_2 - 10*T_1 + X_A >= 280$;
$50*T_5 - 50*T_1 + X_B >= 1000$;
$T_3 - T_1 >= 7$;
$40*T_4 - 40*T_3 + X_D >= 320$;
$10*T_5 - 10*T_3 + X_E >= 160$;
$100*T_6 - 100*T_2 + X_F >= 1700$;
$T_6 - T_5 >= 9$;
$30*T_6 - 30*T_3 + X_H >= 1090$;
$T_5 - T_2 >= 0$;
$T_5 - T_4 >= 0$;
$X_A >= 180$; $X_A <= 200$;
$X_B >= 400$; $X_B <= 450$;
$X_C = 500$;
$X_D >= 160$; $X_D <= 200$;
$X_E >= 80$; $X_E <= 100$;
$X_F >= 500$; $X_F <= 600$;
$X_G = 350$;
$X_H >= 610$; $X_H <= 700$;
$M >= 2780$; $M <= 3100$;
$M = 3000$;
$X_A + X_B + X_C + X_D + X_E + X_F + X_G + X_H <= M$;
$T_i >= 0$;

This formulation of the resource allocation problem is based on a selected budget of M=$3,000. The values of M should be in the range of ($2,780, $3,100) as shown in Table I. Constraints three and seven in the above formulation ensure that activities C and G in the graph of the network are uncrashable. Moreover, constraints nine and ten in the formulation ensure that D_1, and D_2 are dummy activities. This problem was solved using the optimization software LINGO. The optimal values of the decision variables (X_A, X_B, X_C, X_D, X_E, X_F, X_G, X_H) were found to be (200, 450, 500, 200, 100, 500, 350, 700), respectively, as shown in Fig. 4.

```
 Global optimal solution found at iteration:      20
 Objective value:                                 22.00
Variable              Value        Reduced Cost
   T6                 22.0000            0.000000
   T2                 10.0000            0.000000
   T1                  0.0000            1.000000
   XA                200.0000            0.000000
   T5                 13.0000            0.000000
   XB                450.0000            0.000000
   T3                  7.0000            0.000000
   T4                 10.0000            0.000000
   XD                200.0000            0.000000
   XE                100.0000            0.000000
   XF                500.0000            0.000000
   XH                700.0000            0.000000
   XC                500.0000            0.000000
   XG                350.0000            0.000000
   M                3000.0000            0.000000
```

Fig. 4: The solution of the linear programming model using LINGO

Using equation 2, the following durations of the activities A, B, C, D, E, F, G and H can be computed as 8, 11, 7, 3, 6, 12, 9, and 13, respectively. This implies that activity E is crashed from 8 to 6 days, activity H is crashed from 16 to 13 days, and the other activities are left uncrashed. The computer output indicates that the length of the project has been crashed from 24 days to 22 days, ($T_6 = 22$).

The project manager should run the model for different values of the budget to be able to graph the cost-time trade off curve for the project. In our case problem, the model was run for different values of M starting at $2,780 with an increment of the minimum cost of one day reduction of all activities (C=$10/day) and ending at $3100 (the number of runs in this example is 32). The time-cost trade-off curve is then graphed as shown in Fig. 5. The graph in Fig. 5 helps the manager to find the critical budget Point (CBP). CBP can be defined in general as the point on the time-cost trade-off curve where the project duration is minimum and the invested budget is as low as possible. Fig. 5 indicates that investing greater than $2,830 does not reduce project completion time below 22 days. It should be pointed out that in the previous solution the used budget was M=$3,000 and the project was crashed from 24 days to 22 days. However, to achieve this crashing, as shown in Fig. 5, a budget of M=$2,830 is sufficient. This means that a $170 was wasted. In this example problem, it is only useful for the project manager to consider budgets between minimal allowable budget ($2,780) and the budget at the critical budget point ($2,830).

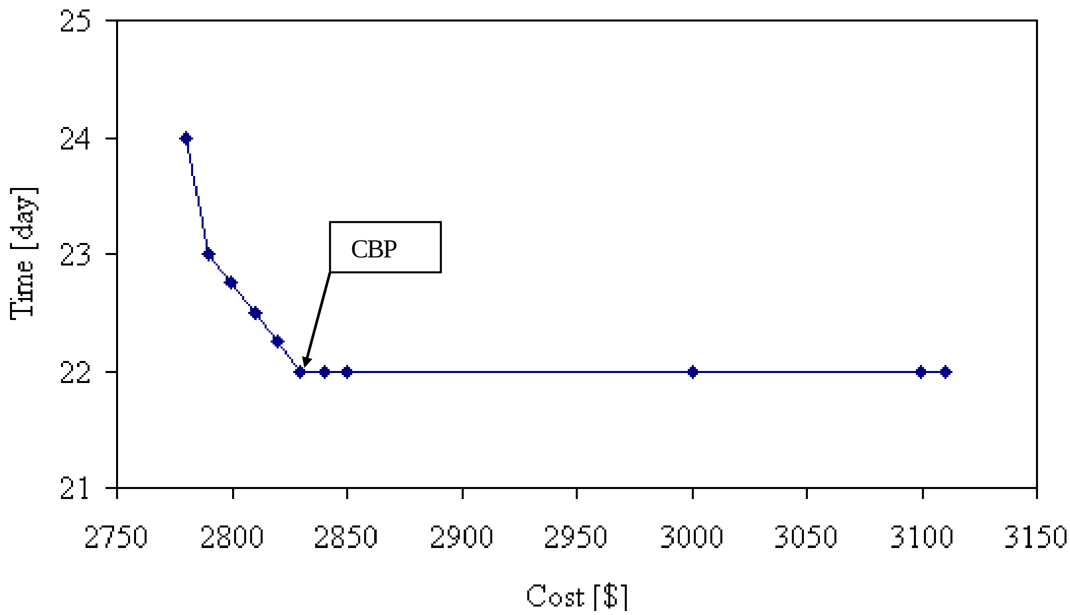

Fig. 5: Time-cost trade-off curve for the linear programming solution of the problem showing the Critical Budget Point (CBP).

It is now apparent that crashing using linear programming, the project manager to have a good knowledge of operations research. Proper formulation of the problem and interpretation of the solution are not simple tasks. The manager must also run the model many times and plot the results to find the CBP in order to select the best possible crashing schedule. In addition, applying this method requires linear programming software with relatively large memory especially if the size of the network is very large.

4.3 The Critical Path Method Manual Solution

This solution is based on the algorithm presented earlier. The solution for the problem starts by calculating the daily reduction of each activity (C). From the network in Fig. (3), the project manager must determine the critical path using the well-known forward pass and backward pass. From the forward pass, the project completion time is found. This process requires calculations of the earliest start (ES), earliest finish (EF), latest start (LS), and latest finish (LF) for each activity. In our problem, one critical path (C, E, G) is found at first with a duration of 24 days. The project cost equals $2,780, which represents the sum of costs of all activities before crashing (sum of normal costs u_i.). The crashing choice here is activity E with the crashing cost per day of $10/day. Activities C and G are not part of the alternatives since they are uncrashable. Next we update the network by reducing the duration for activity E from 8 days to 7 days. The project completion time equals 23 days and its cost is $2,790. Then the forward and backward passes are performed. The network now has two critical paths (C, E, and G; and C, and H). Here, we have to crash activity E from the first path and activity H from the second path. The cost of reducing the project duration by one day equals to $C_E + C_H$ = $40. This makes the project completion time equals 22 days and its cost is $2,830. Updating the project network by reducing the durations of activity E from 7 days to 6 days and activity H from 16 days to 15 days. Then the forward and backward passes are performed. The network has then three critical paths (*path 1*: C, E, and G; *path 2*: C, and H; and *path 3*: A, and F). The crashing alternatives here are: **Alternative 1**: C from path 1 and path 2, and A from path3; **Alternative 2**: C from path 1 and path 2, and F from path 3; **Alternative 3**: E from path 1, H from path 2, and A from path 3; **Alternative 4**: G from path 1, H from path 2, and A from path 3; **Alternative 5**: E from path 1, H from path 2, and F from path 3; and **Alternative 6**: G from path 1, H from path 2, and F from path 3. We have to crash all the activities on any of the alternatives to have a feasible crashing process that results in project completion time reduction, otherwise a slack will be created on the uncrashable critical path(s). Creating a slack on any of the critical paths means that we invested extra resources without reducing the project completion time. In this problem, none of the alternatives is feasible to crash since activities C, E, and G are now uncrashable, and hence the project duration can't be reduced below 22 days.

This process becomes complicated when the number of activities in a project is very large. It is also noticed that this process is tedious and cumbersome, especially when the project has many alternative (parallel) paths.

4.4 Decision Support Tool Solution of the Problem

The proposed decision support tool (developed using Java programming) is used here to solve the problem presented earlier. From the file menu, the new project option is chosen. The new activity button is chosen and the information related to the first project activity with code 1 is entered in the corresponding spaces as illustrated in Fig. 6.

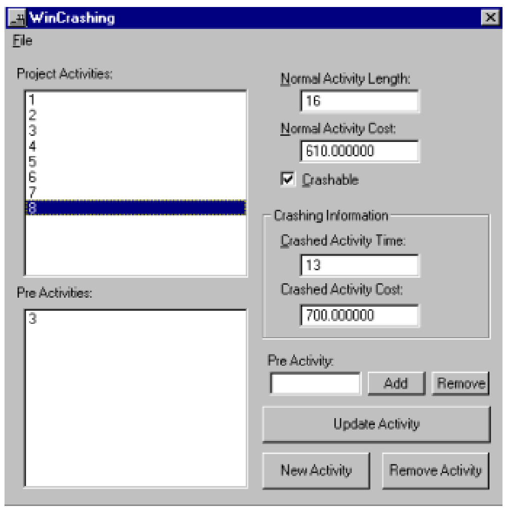

Fig. 6: The screen for entering the data of the problem

Then the Update button is clicked to save the information of the current activity. This process is repeated for all activities of the project. Then, the save option from the file menu is chosen to save the file under the desired name. Selecting the crash option from the file menu to perform the crashing follows this. The decision support tool crashes the project one time period at a time till it becomes uncrashable (perform all crashes). The user is prompted to choose an output file name in order to save the solution. The user then may view the output on the screen or make a printout. If the project manager wants to view an existing project data or modify some of the project information, then the project must be loaded from the file menu using the load option.

The process of entering the information of the project activities takes few minutes for a moderate size project (about 30 activities) and it takes very few seconds to perform the crashing. The output of the crashing tool is shown in Fig. 7.

Original project cost: 2780
Total Project Time: 24

Activity	T	ES	EF	LS	LF	FS	TS	Current Cost	Cost Increase/day (c)
1	10	0	10	2	12	0	2	180	10
2	12	0	12	3	15	3	3	400	50
3	7	0	7	0	7	0	0	500	UNCRASHABLE
4	4	7	11	11	15	4	4	160	40
5	8	7	15	7	15	0	0	80	10
6	12	10	22	12	24	2	2	500	100
7	9	15	24	15	24	0	0	350	UNCRASHABLE
8	16	7	23	8	24	1	1	610	30

Critical Paths:
3, 5, 7

Possibility	Cost Increase
5	10

Total project cost after reducing 1 day: 2790
Total Project Time: 23

Activity	T	ES	EF	LS	LF	FS	TS	Current Cost	Cost Increase/day (c)
1	10	0	10	1	11	0	1	180	10
2	12	0	12	2	14	2	2	400	50
3	7	0	7	0	7	0	0	500	UNCRASHABLE
4	4	7	11	10	14	3	3	160	40
5	7	7	14	7	14	0	0	90	10
6	12	10	22	11	23	1	1	500	100
7	9	14	23	14	23	0	0	350	UNCRASHABLE
8	16	7	23	7	23	0	0	610	30

Critical Paths:
3, 8
3, 5, 7

Possibility	Cost Increase
5, 8	40

Final project cost: 2830
Total Project Time: 22

Activity	T	ES	EF	LS	LF	FS	TS	Current Cost	Cost Increase/day (c)
1	10	0	10	0	10	0	0	180	10
2	12	0	12	1	13	1	1	400	50
3	7	0	7	0	7	0	0	500	UNCRASHABLE
4	4	7	11	9	13	2	2	160	40
5	6	7	13	7	13	0	0	100	UNCRASHABLE
6	12	10	22	10	22	0	0	500	100
7	9	13	22	13	22	0	0	350	UNCRASHABLE
8	15	7	22	7	22	0	0	640	30

Critical Paths:
1, 6
3, 8
3, 5, 7

Fig. 7: The project time-cost trade-off decision support tool output for the problem.
The dark shaded (red) activities are critical and the light shaded (green) are crashing alternatives.

The time-cost trade-off curve is depicted in Fig. (8). The points of the curve are extracted from the output of the tool shown in Fig. (7). It can be noticed here that the critical budget point (CBP) is given directly by the last point of the curve where the budget is $2,830 and the project duration is 22 days.

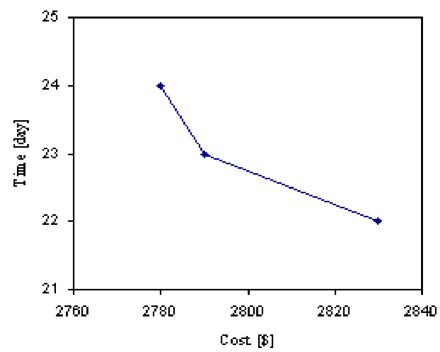

Fig. 8: Time-cost trade-off curve for solution decision support tool of the problem

5. Features of the Proposed Decision Support Tool

The new developed project crashing decision support tool exhibits the following features:

- ♦ Easy to learn and use.
- ♦ Windows oriented (Win 95 and higher).
- ♦ Easy to use pull down menu for file saving, loading (retrieving), and editing (modifying).
- ♦ The solution is saved on a separate file that may be screen viewed or printed.
- ♦ The solution is color coded to facilitate better interpretation.
- ♦ It is capable of crashing projects with very large number of activities. The limiting factor to the number of activities of the project is the computer's memory size.
- ♦ The project is crashed one time period at a time till it becomes uncrashable.
- ♦ It takes only few minutes to have printed output the project schedule (ES, EF, LS, and LF of all the activities) and all the crashing possibilities. The manager's time is saved and he/she may concentrate on other functions of project management.

6. Summary and Conclusion

Project managers have many complicated tasks. Project crashing is one of these important and complex tasks. There are two basic methods for performing project crashing (Gordon): The first method is linear programming method. It offers an excellent optimization tool when the relationship between cost and time reduction is linear. To perform crashing using this method, the project manager must have good knowledge of operations research. The network must be drawn to represent the activity relationship diagram. Formulating the problem in the proper form and interpreting the solution is not a simple task. The project manager must run the model for all possible budgets with increments not greater than the least cost increase per day of all activities (C_i) in order to plot the time-cost trade-off curve and determine the critical budget point (CBP). The second method is the Critical Path Method. It is concerned with the critical path(s) of the network. This method is tedious and requires finding the critical path(s) at every step. It becomes very complex when many parallel critical paths are present where the choice of activities to crash becomes hard to enumerate. Hence, the authors were motivated to develop user-friendly tool that automates the crashing process based on the algorithm presented by Elsayed. The contribution of the proposed tool is to make the complicated task of project crashing as easy as possible. This will free the project manager to concentrate on other tasks that may not be easy to automate. The new project crashing tool has the following advantages: easy to learn and use; windows oriented; wide range of capabilities such as: file saving, retrieving (loading), modifying (editing), and reporting; easy to read and interpret the output which makes it possible for project managers to make decisions without the required knowledge of linear programming or to perform the tedious job of manually crashing the project; ability to crash projects that consist of large number of activities; and finally, the tool can be used as an educational tool for those taking project management courses.

The decision support tool was validated using several small and large problems where it performed perfectly well as expected. One possible extension of this research is to interface the proposed tool with other commercially available project management software packages such as Microsoft Project or Primavera Project Planner.

Acknowledgements
The author would like to thank Engineer Mohammad Al Shami for computerizing the algorithm.

References
Babcock, D. L. and Morse, L. C., Managing Engineering and Technology, Third Edition, Prentice Hall Inc., Upper Saddle River, New Jersey, 2002.
Baccarini, D. “The logical framework method for defining project success”, Project Management Journal, Vol. 30 No. 4, pp. 25-32, 1999.
Belout, A. “Effects of human resource management on project effectiveness and success: toward a new conceptual framework”, International Journal of Project Management, Vol. 16 No. 1, pp. 21-26, 1998.
Corder, S. and Ruby, R, Jr. “Microcomputer-Based Software Alternatives for Project Management”, Journal of Education for Business, Vol. 69 No. 1, pp. 56-60, 1993.
Elsayed, A. E. and Boucher, T. , Analysis and Control of Production Systems, Second Edition, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1994.
Fawcette, J. E. “Choosing Project Management Software”, Personal Computing, Vol. 8 No. 10, pp. 154-167, 1984.
Gordon, G., Pressman, I., and Cohen, S., Quantitative Decision Making for Business, Third Edition, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1990.
Kelly, J. E. Jr. and Walker, M. R. “Critical Path Planning and Scheduling”, Proceedings of Eastern Joint Computer Conference, Boston, 1959.
Kerzner, H., Project Management: A System Approach to Planning, Scheduling, and Controlling, Forth Edition, Van Nostrand Reinhold Company, Inc., New York, 1992.
Project Management Institute (PMI), Project Management Body of Knowledge (PMBOK), 1996.
Shenhar, A.J., Levy, O., and Dvir, D. “Mapping the dimensions of project success”, Project Management Journal, Vol. 28 No. 2, pp. 5-13, 1997.
Shtub, A., Brad, J. F., and Globerson, S. , Project Management: Engineering, Technology, and Implementation, Prentice Hall, Englewood Cliffs Inc., New Jersey,1994.
Sivasankaran, T. R. and Ruby, R., Jr. “Microcomputers and Project Management”, Journal of Education for Business, Vol, 64 No, 7, pp. 315-318,1988.

Biography
Fikri Dweiri is Associate Professor and Vice Dean of College of Engineering at the University of Sharjah (UoS), UAE since 2013. He holds a PhD in Industrial Engineering from the University of Texas at Arlington. He served for 5 years as the chairman of the Industrial Engineering and Engineering Management Department at the UoS, Acting Dean of the School Technological Sciences at the German-Jordanian University and the Founding Chairman of the Industrial Engineering Department at the Jordan University of Science and Technology. His research interest includes multi-criteria decision making, fuzzy logic, quality management, supply chain management, project management, organization performance excellence.