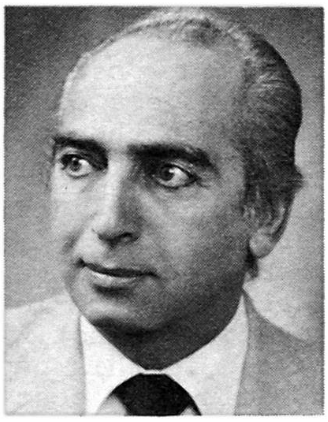
An Integrated Computer Programme for Cost-Time Trade-Off Curves

Un programme informatique intégré pour le calcul des fonctions coût-temps

Ein integriertes Computer-Programm für die Berechnung der Kosten-Zeit-Funktionen

Ch. EPHREMIDIS
Prof. Dr.
National Technical University
Athens, Greece



Charalambos Ephremidis, born 1927, received his mechanical and electrical engineering degrees at the National Technical University of Athens, Greece, and his Dr.-Ing. at the Technical University of Hannover, FR Germany. For ten years was site engineer. Ch. Ephremidis is now Professor at the National Technical University of Athens and director of the Dep. for Construction Project Management.

SUMMARY

The purpose of this paper is to present a new integrated computer programme which calculates the time-cost trade-off curves starting with the basic calculations of the project network and covering all optimization procedures, without additional intermediate input. The optimization procedure is based on a simplified simplex algorithm using lower and upper bounds for the activity durations in order to extend program capacity and to reduce execution time. The programme, which can run on every PC using DOS facilities, has been already tested on real cases with up to 150 activities and 100 paths.

RÉSUMÉ

Cet article présente un nouveau programme informatique intégré, qui calcule les fonctions coût-temps en partant de calculs de base PERT, et couvrant toutes les procédures d'optimalisation, sans mise en mémoire intermédiaire de données supplémentaires. La procédure d'optimalisation se base sur un algorithme Simplex simplifié, utilisant des limites inférieures et supérieures pour la durée des activités dans le but d'étendre la capacité du programme et de réduire le temps d'exécution. Le programme, qui peut être mis en service sur chaque ordinateur personnel avec système DOS, a déjà été testé sur des cas réels avec jusqu'à 150 activités et 100 chemins.

ZUSAMMENFASSUNG

Zweck der vorliegenden Arbeit ist die Präsentation eines integrierten Computer-Programmes, welches die optimalen Werte der Kosten-Zeit-Funktionen berechnet, und zwar ausgehend von der anfänglichen Netzwerkplanung bis zur Ausarbeitung aller Optimierungsprozesse ohne Zwischenspeicherung von zusätzlichen Daten. Das Optimierungsverfahren stützt sich auf einen vereinfachten Simplex-Algorhythmus, welcher untere und obere Grenzen für die Vorgangsdauer benutzt. Der Algorhythmus vergrössert die Rechen-Kapazität und verringert gleichzeitig die Rechenzeit. Das Programm, welches auf PC mit DOS-System läuft, wurde bereits mit Erfolg auf Projekten mit 150 Vorgängen und 100 Wegen verwendet.

1. Indroduction

One of the main goals of project planning is the optimization of the project duration giving the best time to finish the project to the minimum possible cost. Due to the increase of project volume as well as many other unexpected factors, which affect negatively the project duration time and project cost, the optimization technique of the cost-time curves, derived from the network analysis, is today an essential assistance and a powerful tool towards the reduction of project cost through quantitative procedures, that is through objective functions, which lead to exact optimal results. Micro-Computers, which can be easily installed on every site, have increased the significance of the optimization procedures on the improvement of productivity in the construction industry.

The problem of minimizing project direct cost for a given project duration and the total project cost, taking into account the indirect cost, for a variable project duration, as it arises today in practice, can be formulated in the following three ways:

a) For a given project the initial planning is carried out through a network analysis based on normal activity time estimates. The cost control process during project implementation, which is realised with the cumulative Budget S-Curve derived from the network analysis, gives quantitative indications whether the project is running according to schedule or not. In most cases, due to factors which will not be discussed in this paper, the project is running off schedule as to time delay and cost overrun. The contractor in this case has to bring the project back as near as possible to the anticipated values and to the minimum increase of the project direct cost. An optimization procedure must be developed in this case.

b) Another interesting problem appears when the customer wishes to reduce the initial project complection time, due to a recalculation and/or reconsideration of the feasibility factors affecting the project. This must be done also to the minimum possible cost increase.

c) A third formulation of the cost optimization problem, especially in public works or projects which are financed by the goverment, is the need to present sufficient cost data based on objective calculations in order to justify additional payments. The time-cost trade-off procedure to determine additional cost when project duration time reduction is desired is accepted to day by the authorities as a leaglly sound method.

This demand of the construction industry puts the argument for a more reliable procedure to manage cost-time problems, that must also be easily comprehensible and applicable by the user, even by the non specialist one.

The integrated computer program, which is presented in this paper, is one more attempt to this direction: to give the user a computer-aided, quick, exact and easy to apply method to calculate the optimum values of the direct cost-time curve as well as the optimum time to finish a project corresponding to the minimum total project cost.

2. Development of the algorithm

It is assumed that the activity time-cost trade-off points lie on a continuous linear decreasing curve as shown in Fig.1. The case of a piece-wise linear decreasing curve on convex shape is an extension of the former assumption and can also be handled by the program. This case will be examined in a separate paper.

The objective function of the direct cost is formulated as follows:

$$CD = \sum_{j=1}^{n} CNj + \sum_{j=1}^{n} (TNj-Tvj) \times CAj \rightarrow Min \qquad (1)$$

subject to:

$$\sum_{j=1}^{n} A(i,j) \times Tvj \leq TP \qquad i = 1 \text{ to } m \qquad (2) \qquad I$$

and $Tmj \leq Tvj \leq TNj$ (3)

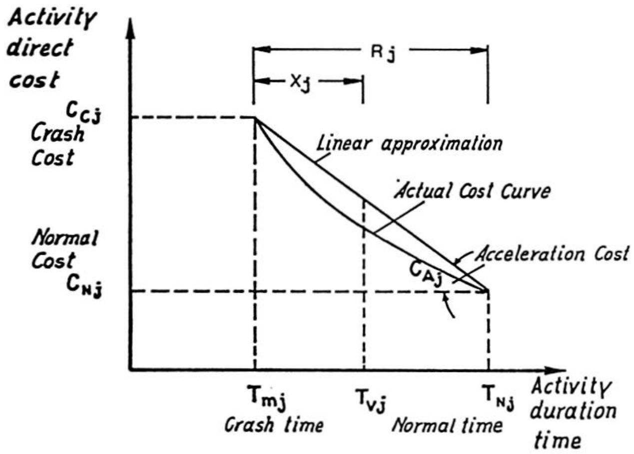


Fig.1. Activity cost-time curve is assumed to be a continuous linear decreasing curve. The algorithm and the present computer program is based on this assumption.

where:
CD = Direct project cost for the disired project time TP which has to be minimized.
CNj= Normal cost of activity j for the normal duration time TNj.
Tvj= Decision variable of time for activity j.
CAj= Acceleration cost of activity j.
A(i,j) = Coefficient of activity j which can be equal to 1 or 0, "i" is the number of path, "j" the number of activity.
TP = Project duration time.
Tmj and TNj = Lower and upper bounds of variable Tvj.
TNj= Normal activity duration time = activity upper bound.

The linear model (I) is equivalent to:

$$\sum_{j=1}^{n} CAj \times Tvj \rightarrow Max \qquad (1') \qquad (1')$$

Subject to:

$$\sum_{j=1}^{n} A(i,j) \times Tvj \leq TP \qquad i = 1 \text{ to } m \qquad (2') \qquad II$$

and

$$Tmj \leq Tvj \leq TNj \qquad (3')$$

or after the substitutions:

$$0 \leq Tvj - Tmj \leq TNj - Tmj$$
$$Xj = Tvj - Tmj$$
$$Rj = TNj - Tmj$$

The new linear model becomes:

$$\sum_{j=1}^{n} CAj \times Xj \rightarrow Max$$

Subject to:

$$\sum_{j=1}^{n} A(i,j) \times Xj \leq TP - \sum_{j=1}^{n} A(i,j) \times Tmj \qquad i = 1 \text{ to } m \qquad III$$

and $0 \leq Xj \leq Rj$

Thus, the model maximises the savings against the crash cost subject to the conditions that

a) the sum of the activity durations on any network path is less than or equal to the project duration.
b) the duration of each activity is between crash and normal time.

3. Final optimality. Minimizing total project cost.

The total project cost-time curve is given by the function:

$$CT = CD_{opt} + G \times TP \longrightarrow Min$$

where:

G = Indirect project cost = expenses or other money values per time unit.
TP = The considered project time.

The CT_{Min} - Value, calculated by the program, is the minimum total cost for the considered time TP to finish the project. Finally, through a common graphic assistance such as IBM or Lotus, the program prints the cost-time curves in order to give a comprehensive control view of the optimization problem as a whole.

4. The computer program

The computer program is based on the normal technique of the activity on arrow networks and on a condensed linear programming model using lower and upper-bounded variables for the activity duration times. The program is written in BASICA-language and can be run in a Micro-Computer (PC-IBM or any other compatible computer using DOS facilities).

The project planning program can handle projects with up to 550 activities, may be more depending on the number of nodes of the network, which affect the computer memory available. If optimization of project cost and of the corresponding activity time chart is desired, then the capacity of the program is reduced normally to 150 activities, this value depending again on the total number of paths in the network. This means that if a project has many parallel interconnected paths then the number of activities is reduced so that the product activities x paths does not exceed the limit of approximately 14000.

5. Analysis of the computer program

5.1. Main program

The main program is a computer aided network analysis which gives the following project data:

a) Table of activity duration times with code number, description, earliest start and finish, latest finish and start, total and free float (Fig.2 and 3).
b) Bar chart schedule of activities with indication of critical path and free float (Fig.4).
c) Table of nodes with time data, which enables the user to develop drawings of the network (Fig.5).
d) Project cost control based on a cumulative standart S-Curve (budget curve) with a print-out of time and cost control (Fig.6).

The input data required for this program are:

The activity duration time, which is defined as normal activity time.

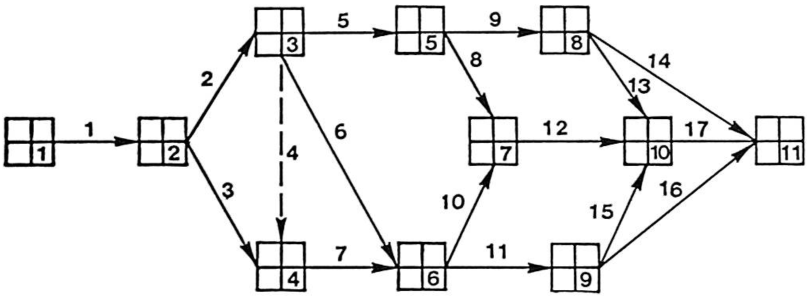
The activity sequence in the network based simply on the node numbers.

The code number and description.

The normal activity cost.

The charts and graphics issued by the program will be demonstrated on the following example.

If project cost-control is not desired, then the program is linked automatically to the optimization programs which follow.




```
Κλαδος 1     1- 2- 4- 7- 10- 12- 17
Κλαδος 2     1- 2- 4- 7- 11- 15- 17
Κλαδος 3     1- 2- 4- 7- 11- 16
Κλαδος 4     1- 2- 5- 8- 12- 17
Κλαδος 5     1- 2- 5- 9- 13- 17
Κλαδος 6     1- 2- 5- 9- 14
Κλαδος 7     1- 2- 6- 10- 12- 17
Κλαδος 8     1- 2- 6- 11- 15- 17
Κλαδος 9     1- 2- 6- 11- 16
Κλαδος 10   1- 3- 7- 10- 12- 17
Κλαδος 11   1- 3- 7- 11- 15- 17
Κλαδος 12   1- 3- 7- 11- 16
```

Fig.2. A network presentation with 17 activities, 11 nodes and 12 paths to be used to demonstrate the optimization algorithm.

```
*********************************************
*                                           *
*       PROJECT TIME AND COST PLANNING      *
* Program ACRO-H.Efr. (Copyright Okt.1985)  *
*********************************************
Number of Activities: 17    Number of nodes: 11
       PROJECT DURATION TIME: 72 DAY(S)
```

A/n	ACTIVITY	CRITIC	TIME DAY(S)	EARLST TIME START	EARLST TIME END	LATEST TIME START	LATEST TIME END	FLOATS Rt	FLOATS Rf
1	AAA1	CRITIC	12	0	12	0	12	0	0
2	AAA2	CRITIC	10	12	22	12	22	0	0
3	AAA3		8	12	20	18	26	6	2
4	Dummy		0	22	22	26	26	4	0
5	AAA5		8	22	30	30	38	8	0
6	AAA6	CRITIC	14	22	36	22	36	0	0
7	AAA7		10	22	32	26	36	4	4
8	AAA8		12	30	42	44	56	14	8
9	AAA9		16	30	46	38	54	8	0
10	AA10		14	36	50	42	56	6	0
11	AA11	CRITIC	12	36	48	36	49	0	0
12	AA12		8	50	58	56	64	6	6
13	AA13		10	46	56	54	64	8	8
14	AA14		12	46	58	60	72	14	14
15	AA15	CRITIC	16	48	64	48	64	0	0
16	AA16		14	48	62	58	72	10	10
17	AA17	CRITIC	8	64	72	64	72	0	0

Fig.3. Table of activity time data.

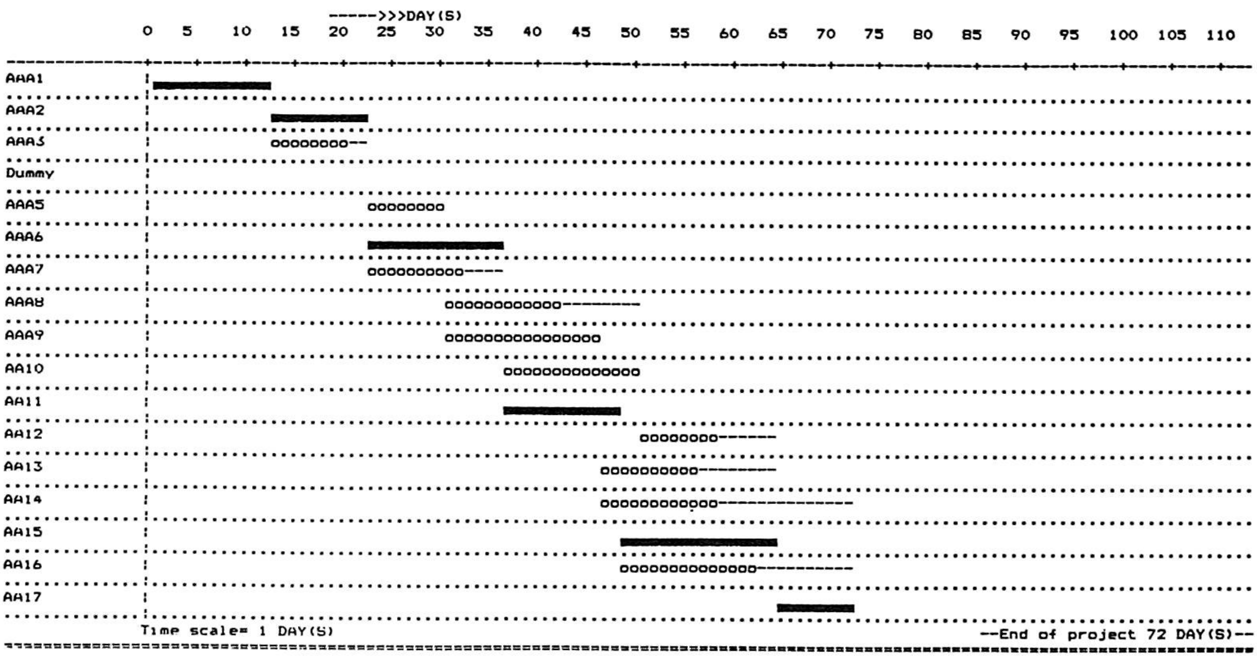


Fig.4. Bar chart schedule with critical path and free floats.

ACTIVITY SEQUENCE AND NODE TIME DATA

	ACTIVITY	NODE START	SZ	FZ	Ro		NODE END	SZ	FZ	Ro
1	)AAA1	< 1 >	0	0	0	▪▪▪▪>>	< 2 >	12	12	0
2	)AAA2	< 2 >	12	12	0	▪▪▪▪>>	< 3 >	22	22	0
3	)AAA3	< 2 >	12	12	0	xxxx>>	< 4 >	26	22	4
4	)Dummy	< 3 >	22	22	0	xxxx>>	< 4 >	26	22	4
5	)AAA5	< 3 >	22	22	0	xxxx>>	< 5 >	38	30	8
6	)AAA6	< 3 >	22	22	0	▪▪▪▪>>	< 6 >	36	36	0
7	)AAA7	< 4 >	26	22	4	xxxx>>	< 6 >	36	36	0
8	)AAA8	< 5 >	38	30	8	xxxx>>	< 7 >	56	50	6
9	)AAA9	< 5 >	38	30	8	xxxx>>	< 8 >	54	46	8
10	)AA10	< 6 >	36	36	0	xxxx>>	< 7 >	56	50	6
11	)AA11	< 6 >	36	36	0	▪▪▪▪>>	< 9 >	48	48	0
12	)AA12	< 7 >	56	50	6	xxxx>>	< 10 >	64	64	0
13	)AA13	< 8 >	54	46	8	xxxx>>	< 10 >	64	64	0
14	)AA14	< 8 >	54	46	8	xxxx>>	< 11 >	72	72	0
15	)AA15	< 9 >	48	48	0	▪▪▪▪>>	< 10 >	64	64	0
16	)AA16	< 9 >	48	48	0	xxxx>>	< 11 >	72	72	0
17	)AA17	< 10 >	64	64	0	▪▪▪▪>>	< 11 >	72	72	0

Fig.5. Table of nodes with time data.

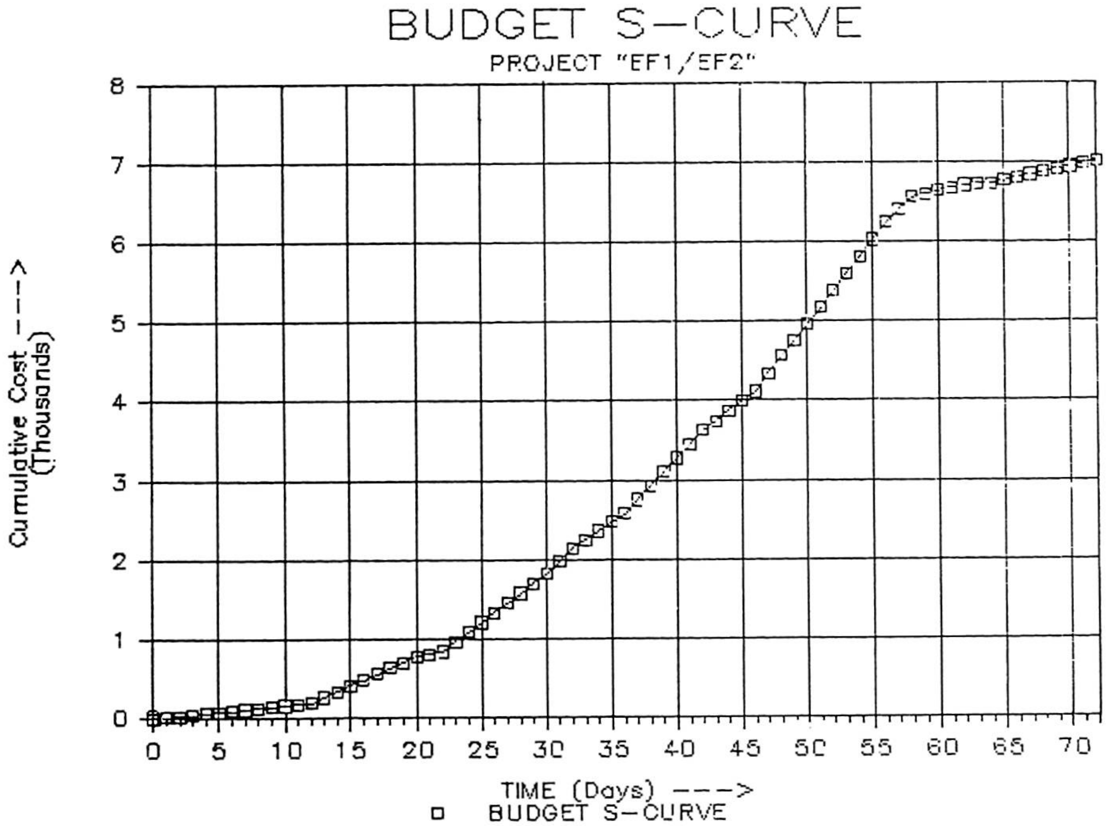
5.2. Identification of all paths with the corresponding activities in the network.

Since the optimization of the cost-time curve is based on a linear programming model it is necessary to identify all paths of the network with the corresponding activities. This program is the basis of the constraint matrix of the LP model (Fig.2,7).

INITIAL PROJECT DATA

Activity		Lower limit	Normal time	Acceler. cost	Normal cost
1	AAA1	6	12	25	200
2	AAA2	5	10	30	300
3	AAA3	8	8	0	350
4	Dummy	0	0	0	0
5	AAA5	4	8	20	300
6	AAA6	10	14	15	480
7	AAA7	10	10	0	500
8	AAA8	8	12	18	620
9	AAA9	12	16	22	380
10	AA10	12	14	30	450
11	AA11	8	12	25	800
12	AA12	8	8	0	400
13	AA13	5	10	20	550
14	AA14	7	12	30	850
15	AA15	10	16	15	200
16	AA16	12	14	22	350
17	AA17	4	8	30	280

INDIRECT COST 55 PER TIME UNIT   TOTAL COST   7010



CONTROL DATE: 52 DAY(S)

CUMULATIVE PROJECT COST TO CONTROL DATE: CA(52)= 5800 Money units

WORK PERFORMED TO CONTROL DATE: CR(52)= 4120 Money units

PROJECT COST ACCORDING SCHEDULE TO CONTROL DATE: CS(52)= 5385 Money units

COST INCREASE (o/o): 41 O/O
TIME DELAY: 5 DAY(S)
ESTIMATED NEW FINAL PROJECT COST: 9868 Money units INITIAL: 7010

COST INDEX: .7103448
TIME INDEX: .7650881

Fig.6. Initial project data, Cumulative S-Curve and Corresponding cost control card for the 52d day.

	B(I)	C(I)	A(I,J) 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	3	0	1	1	0	1	0	0	1	0	0	1	0	1	0	0	0	0	1
2	5	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	1	0	1
3	7	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0
4	13	0	1	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1
5	12	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
6	14	0	1	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0
7	3	0	1	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	1
8	5	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1
9	7	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
10	0	0	1	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	1
11	2	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1
12	4	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0
		ZC(J)=	-25	-30	0	0	-20	-15	0	-18	-22	-30	-25	0	-20	-30	-15	-22	-30
		R(J)=	6	5	0	0	4	4	0	4	4	2	4	0	5	5	6	2	4

OPT.VALUE= 0
R= 17
0 1000000 4
10 17

	B(I)	C(I)	A(I,J) 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	3	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	5	0	0	1	-1	1	0	0	0	0	0	-1	1	-1	0	0	1	0	0
3	7	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	1	0
4	13	0	0	1	-1	0	1	0	-1	1	0	-1	0	0	0	0	0	0	0
5	12	0	0	1	-1	0	1	0	-1	0	1	-1	0	-1	1	0	0	0	0
6	14	0	1	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0
7	3	0	0	1	-1	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
8	5	0	0	1	-1	0	0	1	-1	0	0	-1	1	-1	0	0	1	0	0
9	7	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
10	0	30	1	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	1
11	2	0	0	0	0	0	0	0	0	0	0	-1	1	-1	0	0	1	0	0
12	4	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0
		ZC(J)=	5	-30	30	0	-20	-15	30	-18	-22	0	-25	30	-20	-30	-15	-22	0
		R(J)=	6	5	0	0	4	4	0	4	4	2	4	0	5	5	6	2	4

OPT.VALUE= 0
R= 14
14 1000000 5

	B(I)=	C(I)=	A(I,J) 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	3	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	3	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	6	0	0	0	0	0	0	0	0	-1	1	0	0	1	1	0	0	0	0
5	3	15	0	1	-1	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
6	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	-1
7	3	20	1	1	0	0	1	0	0	0	-1	0	0	0	-1	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	30	1	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	1
11	2	25	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	-1	0
12	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
		ZC(J)=	50	5	40	0	0	0	40	18	2	0	0	30	0	30	0	12	35
		R(J)=	6	5	0	0	4	4	0	4	4	2	4	0	5	5	6	2	4

OPT.VALUE= 609
R= 15
Z= 609
X 6 = 3
X 5 = 3
X 10 = 0
X 11 = 2
X 15 = 0
****

OPTIMAL TIME SCHEDULE
====================

PROJECT TIME: 48 MINIMUM DIRECT COST: 7665 TOTAL COST: 10305

Optimal time for Activities:

T 1	T 2	T 3	T 4	T 5	T 6	T 7	T 8	T 9	T 10	T 11	T 12	T 13	T 14	T 15	T 16	T17
6	5	8	0	7	13	10	12	16	12	10	8	10	12	10	14	4

Fig.7. Print-out of the constraint matrixes (initial, first and final). Optimal time squedule for project due time TP=48 days (crash time).

5.3. Auxiliary program

The linear model is supplemented in this program with the additional project data which are essential for the optimization procedure. These are the lower limit of the activity time, defined as lower bound (the upper bound or normal time has already been inputed in the "Main Program"), the acceleration cost for each actiivty and the normal cost.

5.4. Optimization algorithm

This is the last stage of the optimization procedure. The solution of the LP-Model, which has been formed by the previous programs, follows the normal simplex transformations supported by the method of the lower and upper bounded variables in order to reduce the size of the constraint matrix to the number of paths and consequently the computer calculation time.

For a given project time (Fig.8, project due time = 50) the program gives the minimum direct cost, the total project cost for the same time and the corresponding optimal time schedule for all activities. The computation time needed for one time period for a project with 17 activities and 12 paths (Fig.2) after the data input is approximately 30 seconds (IBM-PC).

PROJECT TIME: 50 MINIMUM DIRECT COST: 7520 TOTAL COST: 10270

Optimal time for activities:

1	2	3	4	5	6	7	8	9	10
6	6	8	0	8	12	10	12	16	14
11	12	13	14	15	16	17			
12	8	10	12	10	14	4			

PROJECT TIME: 51 MINIMUM DIRECT COST: 7490 TOTAL COST: 10295

Optimal time for activities:

1	2	3	4	5	6	7	8	9	10
6	6	8	0	8	12	10	12	16	14
11	12	13	14	15	16	17			
12	8	10	12	10	14	5			

PROJECT TIME: 52 MINIMUM DIRECT COST: 7460 TOTAL COST: 10320

Optimal time for activities:

1	2	3	4	5	6	7	8	9	10
6	6	8	0	8	12	10	12	16	14
11	12	13	14	15	16	17			
12	8	10	12	10	14	6			

Fig.8. Optimal time schedule for intermediate project period time TP=50, 51 and 52 days.

If it is required to compute the above mentioned data for all the time periods between crash and normal project duration time with a given time step (every subsequent time unit) then the program makes all calculations and print-outs and gives the optimum time to finish the project, the minimum direct cost and the minimum total cost for the optimal time (Fig.9). Finally, if desired, it is very easy and without additional effort to have a graphic representation of the time-cost curves, both direct cost curve and total cost curve, using a common graphic assistant, as shown in Fig.10 and 11.

The simplification in the algorithm refers:

a) to the elimination of the columns for the basic variables with no danger of ambiguity arising.

PROJECT TIME: 71 MINIMUM DIRECT COST: 7025 TOTAL COST: 10930

Optimal time for activities:

1	2	3	4	5	6	7	8	9	10
12	10	8	0	8	13	10	12	16	14
11	12	13	14	15	16	17			
12	8	10	12	16	14	8			

PROJECT TIME: 72 MINIMUM DIRECT COST: 7010 TOTAL COST: 10970

Optimal time for activities:

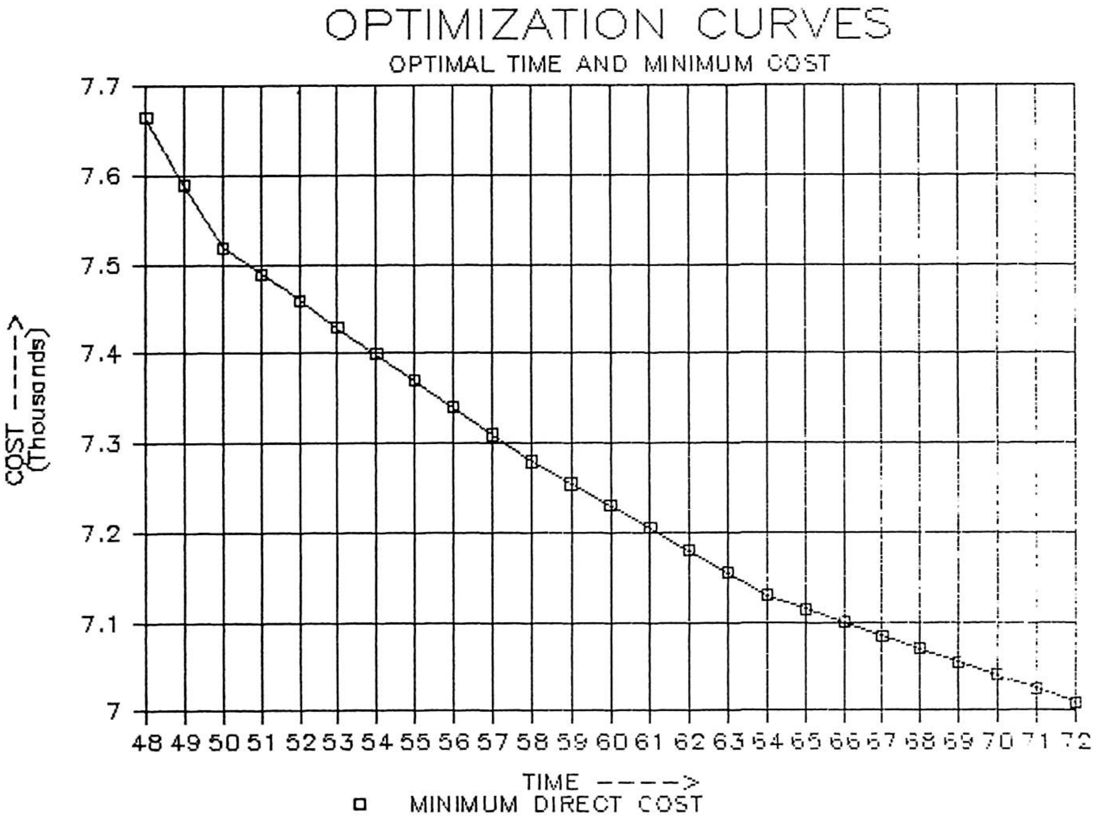
1	2	3	4	5	6	7	8	9	10
12	10	8	0	8	14	10	12	16	14
11	12	13	14	15	16	17			
12	8	10	12	16	14	8			

FINAL OPTIMALITY

---->BEST TIME TO FINISH PROJECT	50
---->MINIMUM DIRECT COST	7520
---->MINIMUM TOTAL COST	10270

End of analysis

Fig.9. Full optimal time schedules with project total cost as a function of completion time. Project normal time = 72 days. Crash time = 48 days.



(Lotus printgraph)

Fig.10. A graphic representation of the optimal direct cost-time curve from the crash to the normal project period time.

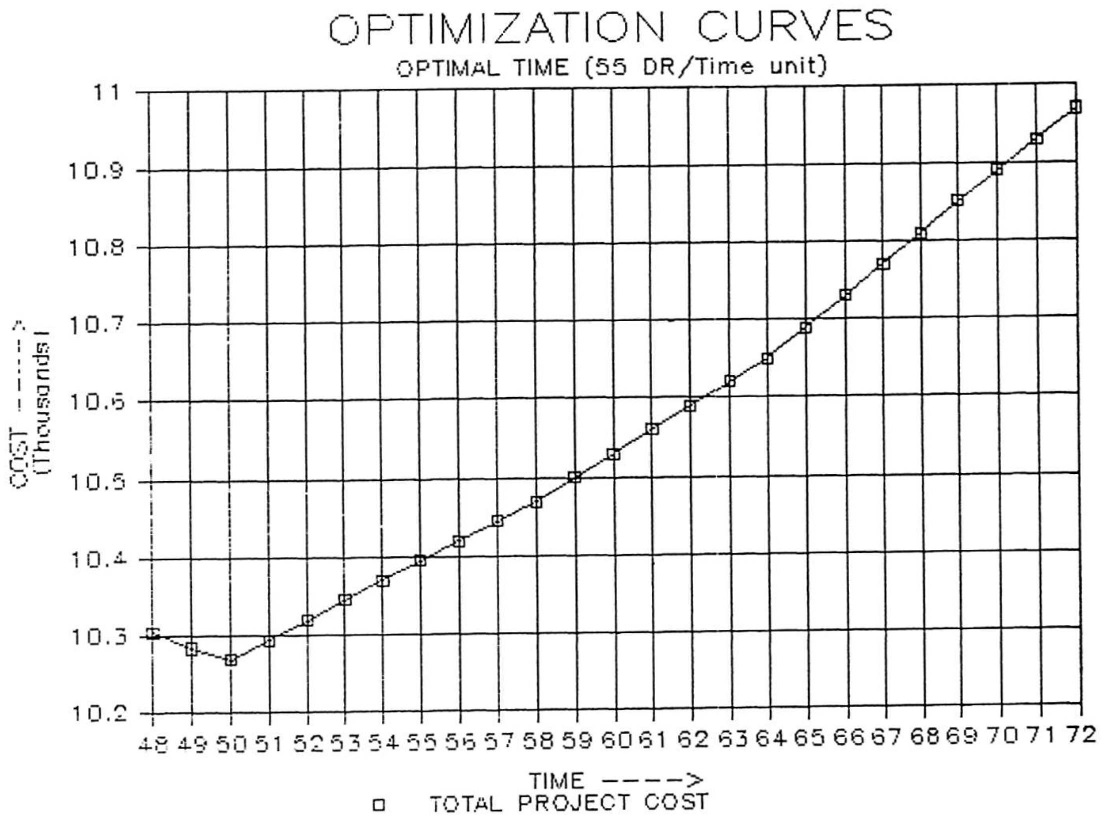


Fig.11. A graphic representation of the optimal project total cost-time curve.

b) to the use of lower and upper bounds for the time variables. This reduces the number of constraints to the number of paths.

c) to a single pass data input in such a way that the user can manipulate with the program without owing special knowledge.

REFERENCES

1. TAHA H., Operations Research, An Introduction, book, 3d Edition, Macmillan Publishing Co., Inc., New York, 1982.
2. JURECKA W., Netzwerkplanung im Baubetrieb, book, Bauverlag GmbH, Wiesbaden, 1967.
3. MODER J., PHILIPS C., DAVIS E., Project Management with CPM and Pert, book, Van Nostrand R.Comp., New York, 1983.
4. WAGNER H., Principles of Operations Research, book, 2d Edition, Practice-Hall, Inc., New Jersey, 1975.
5. PANAGIOTAKOPOULOS D., A CPM Time-Cost computational algorithm for realistic activity cost functions, Working paper, McGill University, Montreal, 1975.
6. FULKERSON R., Expected Critical Path Lengths in PERT-Networks, Operations Research, Vol. 10, No.6, November 1962.
7. FULKERSON R., A Network Flow Computation for Project Cost Curves, Management Science, Vol.7. No.2, January 1961.
8. SEELING R., Zeit-Kosten Optimierung von Netzplänen, Die Bautechnik, Heft 4, 1979.