Mansoura Engineering Journal

Volume 25 | Issue 4 Article 1

2-6-2021

Redundant Path Elimination: An Efficient Time-Cost Trade-off Optimization Model.

Mohammad Ammar
Civil Engineering Department., Faculty of Engineering., Tanta University., Tanta., Egypt.

Recommended Citation
Ammar, Mohammad (2021) "Redundant Path Elimination: An Efficient Time-Cost Trade-off Optimization Model.," *Mansoura Engineering Journal*: Vol. 25 : Iss. 4 , Article 1.
Available at: https://doi.org/10.21608/bfemu.2021.146823

REDUNDANT PATH ELIMINATION: AN EFFICIENT TIME-COST TRADE-OFF OPTIMIZATION MODEL

نموذج رياضي كفؤ لحل مشكلة الوقت والتكلفة باستبعاد المسارات الغير فعالة

By Mohammad A. Aminar
Assistant Prof., Faculty of Engineering, Tanta University

الخلاصة :

يعتبر ضغط الوقت/التكلفة من المشكلات التي تواجه المشتغلين بمشروعات التشييد حيث يعتمد عليها في تحديد الوقت الأمثل لتنفيذ المشروع بأقل تكلفة. تستخدم الطرق التقريبية والبرمجة الرياضية لحل هذه المشكلة. من العيوب الأساسية والتي تحد من استخدام البرمجة الرياضية هو العدد الكبير من المتغيرات والقيود المطلوبة مما يجعلها غير قابلة للتطبيق إلا على المشروعات الصغيرة فقط. في هذا البحث تم استحداث نموذج برمجة رياضية يعتمد على تقليل عدد القيود المطلوبة وبالتبعية عدد المتغيرات المطلوبة- مما يجعل النموذج الجديد صالحا للتطبيق على المشروعات الكبيرة. عند التطبيق العملي يتمثل عدد كبير من المسارات الحرجة (الفعالة) بالمقارنة بعدد المسارات الكلية المكونة لشبكة الممثلة للمشروع أما المسارات التي لن تكون حرجة في أي دورة من عملية ضغط البرنامج الزمني غير فعالة. يوجد عدد من الأنشطة التي تقع فقط على المسارات الغير فعالة والتي يعتبر ادخالها غير ذي فائدة مع أنه يزيد من حجم المشكلة عند عمل النموذج نستبعد نحن استبعاد المسارات الغير فعالة وكذلك الأنشطة الغير فعالة مما ينتج عنه تقليل عدد القيود المطلوبة بدرجة كبيرة وكذلك عدد المتغيرات. قدم البحث تطبيقا على مثال لتوضيح كيفية تكوين معادلات دالة الهدف والقيود المختلفة ومقارنتها بالنموذج الرياضي في حالة عدم استبعاد المسارات والأنشطة الغير فعالة.

ABSTRACT

Time-cost trade-off is a decision making problem in construction management. Numerous optimization models have been developed to solve this problem. Almost all of them are not applicable for even small-sized projects, because of the unmanageable required number of variables and constraints. The proper sequence (logic) of activities can be maintained by allowing for one constraint for each possible path. In practice, only small number of paths are dominant, while the others are redundant. In this paper, an efficient time-cost optimization model, which minimizes project cost, is developed depending on eliminating redundant paths. Precise activity time-cost relationship is used and overlapping between consecutive activities is permitted. The model is formulated in the form of zero-one programming. The model constraints include zero-one constraints and network logic constraints. In formulating network logic constraints, only dominant paths are considered. Redundant paths are eliminated and consequently unnecessary decision variables are excluded. The model guarantees the optimal solution. The model is entirely formulated by interface with computer subroutine. The model requires as input: precedence relationship between activities, overlap values, and discrete utility data for project activities. The model efficiently reduces the problem size and can be used for large-sized project networks

Accepted November 27, 2000

INTRODUCTION

The time and cost parameters of a construction project have been identified as major factors of the decision making process. In Critical Path Method (CPM) analysis, the objective is to establish a minimum project cost by making use of time durations of each activity at minimum possible cost. The primary impact of timing is money. Time is thus an equally essential factor. Therefore, construction management must focus on minimizing overall project cost with reasonable time schedule based on realistic assumptions. Since cost is actually a function of time, it is necessary to determine the project time-cost trade-off curve, which gives the minimum possible cost for completing the project within a specified completion time.

The project time-cost problem has an infinite number of solutions, because so many combinations of alternative ways for performing project activities are possible. If time is of no concern, each activity could be performed at its lowest possible (normal) cost. If cost is of no importance, each activity could be speeded up to be completed in the least (crash) time. Between these two limits lies the best (optimal) solution, but to find it, it requires consideration of complex collection of concurrent, interrelated, and overlapping activities (Antill and Woodhead, 1982). Time-cost trade-off problem has been traditionally solved by two distinct approaches; mathematical optimization and heuristic methods. Mathematical modeling of the problem requires large number of variables and constraints for even medium-sized construction projects. This research introduces an attempt to reduce significantly the problem size.

In this paper a new optimization model for solving project time-cost trade-off problem is presented. The discrete activity time-cost relationship is considered. Overlapping between project activities in the form of FTS is allowed. The model is formulated in the form of zero-one linear programming problem which minimizes project cost in a general form. The model constraints include zero-one constraints and network logic constraints. The general model is, then, reformulated to eliminate those unnecessary decision variables as well as redundant paths constraints. Network logic constraints are formulated only for dominant paths.

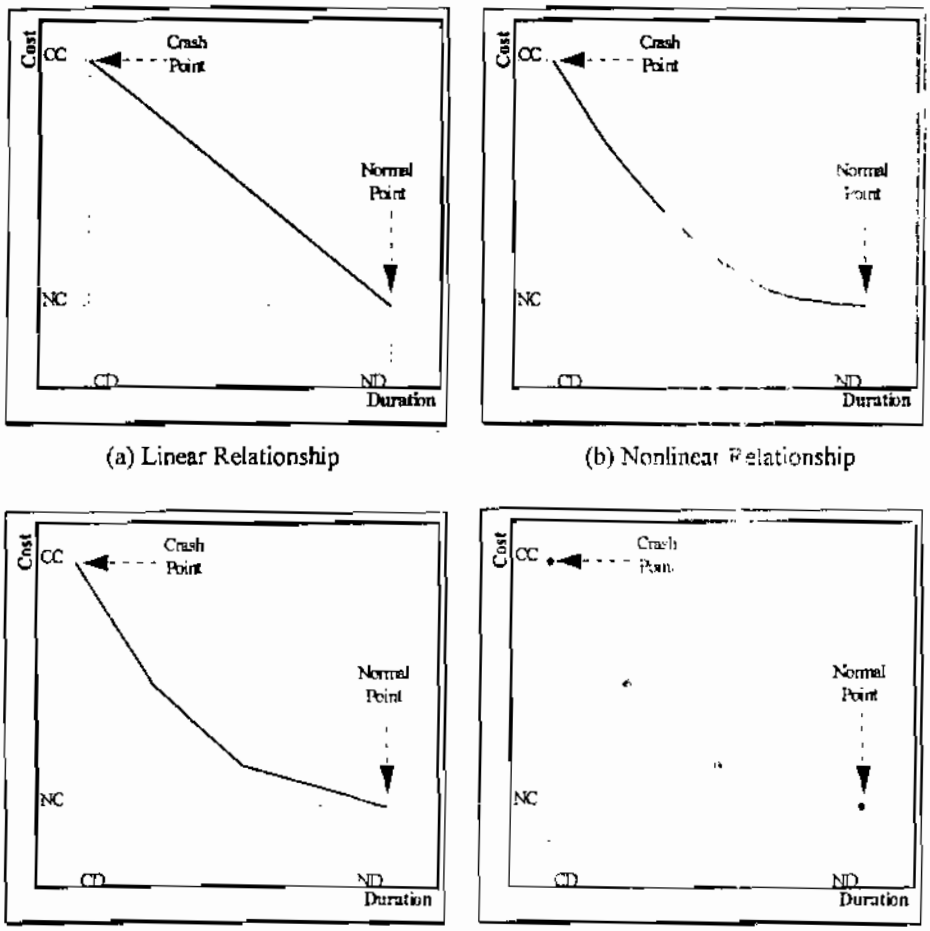
TIME-COST TRADE-OFF PROBLEM

Mathematical optimization and heuristic methods are the two major approaches used to solve the time-cost trade-off problem. Recently, other techniques have been introduced to solve the same problem. These include Genetic Algorithms; GA, (Li and Love 1997) and Neural Networks (Adeli and Karim 1997). Mathematical methods convert the problem into standard mathematical programming models and then use linear, integer, or dynamic programming to obtain the optimal solution of the problem. However, formulating the objective function as well as the required constraints of the problem is time-consuming and prone to errors (Liu et al 1995). Heuristic methods provide a way to obtain good solutions but do not guarantee optimal solutions. However, they require less computational effort than mathematical methods.

In developing project time-cost curve, the scheduler should rely on realistic assumptions. Many forms of activity time-cost relationship have been assumed. They include linear, nonlinear, piece-wise, and discrete point relationship. They are shown in Fig.1. The least direct cost for carrying-out an activity is called the normal cost (NC) and the corresponding duration is called normal duration (ND). The shortest possible duration for performing an activity is referred to as crash duration (CD) and the corresponding cost is called crash cost (CC). The points between these two limits show the costs for the various feasible time in which an activity can be speeded up. The detailed time and cost information for an activity obtained form construction estimate are referred to as utility data.

The function that may relate time and cost of construction activities is of great importance, since the type of optimization technique depends, mainly, on that relationship

Eldosouky et al (1991) proved that the discrete point relationship is the realistic representation for activity time-cost relationship. They concluded that: "Although this representation is difficult to be used in time-cost optimization problem, it is the correct one." Most of the existing time-cost optimization methods assume linear activity time-cost relationship in order to control computational effort. On the other hand, every reported methodology that attempts to deal with other activity time-cost relationship either completely fails to reach the computer model stage or, if it does, it is accompanied by the "*only for small networks*" warning (Panagiotakopoulos 1977).



(a) Linear Relationship (b) Nonlinear Relationship

(c) Piece-Wise Relationship (d) Discrete Relationship

Figure 1. Activity Time-Cost Relationship

Another factor which must be considered is the fact that most construction activities overlap. Harris (1978) defines four types of overlap that may exist between construction activities. The most common used one is the Finish To Start (FTS) relationship. Harris concluded that overlapping activities allows for more realistic modeling of construction projects Perera (1982) introduced a linear programming model to handle overlapping activities. However, the model is based on the unrealistic assumption that activity time-cost relationship is linear. Although

Eldosouky et al (1991) formulated an integer programming model considering discrete utility data and overlapping activities, the model is intended for small- to medium-sized projects due to excessive computer storage requirements.

FORMULATION OF THE GENERAL MODEL

Consider a project having (n) activities, where utility data for all project activities are represented by discrete point relationship. For each activity (i), m_i discrete points must be specified, where $m_i \geq 1$. Every discrete point corresponds to a specific way of carrying-out the activity. Let d_i and c_i be variables representing duration and cost for the activity (i), respectively. Let d_{i1} and c_{i1} represent the normal discrete point, while d_{im_i} and c_{im_i} correspond to the crash point. For activities having only one discrete point, the normal and crash points coincide.

Decision Variables

A single zero-one variable; x, is needed for each discrete point in the utility data. These zero-one variables are introduced to insure that only one discrete point is selected for each activity. The duration and cost for an activity (i), in terms of zero-one variables, can be expressed as follows:

$$\text{Activity Duration } (d_i) = d_{i1}x_{i1} + d_{i2}x_{i2} + \ldots\ldots + d_{im_i}x_{im_i} = \sum_{j=1}^{m_i} d_{ij}x_{ij} \qquad (1)$$

and,

$$\text{Activity Cost } (c_i) = \sum_{j=1}^{m_i} c_{ij}x_{ij} \qquad (2)$$

Where x_{ij} is a zero-one variable belongs to the discrete point number (j) for the activity (i). The project direct cost is the summation of cost of all activities, and can be formulated as follows:

$$\text{Project Direct Cost (PDC)} = \sum_{i=1}^{n}\sum_{j=1}^{m_i} c_{ij}x_{ij} \qquad (3)$$

For an activity (i), the model can be forced to select a single duration, and consequently the corresponding cost, at a time by the following condition:

$$\sum_{j=1}^{m_i} x_{ij} = 1 \qquad i = 1, 2, \ldots\ldots\ldots\ldots, n \qquad (4)$$

This type of constraints will be referred to as zero-one constraint. Since every discrete point requires zero-one variable, the number of zero-one variables needed is the sum of discrete points for all activities, whereas number of zero-one constraints equals number of activities; n.

Network Logic Constraints

The completion time of a project, λ, could be constrained by one of two methods (Crowston 1970). The first approach is to allow for a precedence constraint for each immediate predecessor relationship in the project network, and in total they constraint the finish time of the last activity, which is actually the project completion time. This approach was used in all existing

optimization techniques (Liu et al 1995, Eldosouky et al 1991, Cusak 1985, Pe... 1982, Wiest 1973, Meyer and Shaffer 1965, and Fulkerson 1961). The second is to allow for one constraint for each possible path from the first activity to the last one in the project network. Satisfying all possible paths constraints simultaneously, the right sequence of project activities is maintained. In the present model, the second procedure will be considered.

Path Constraints Formulation

Let {PATH} be a set of all possible paths comprising the project network. Each path contains some or all project activities. Each path P_k in the set {PATH} could be expressed as follows:

$$P_k = \{a_1y_1, a_2y_2, \ldots\ldots, a_iy_i, \ldots\ldots, a_ny_n\}_k$$

Where y_i is a zero-one parameter, which takes the value of unity if activity a_i exists on the path P_k, otherwise it takes the value of zero. The set of all possible paths {PATH} could be expressed as follows:

$$\{PATH\} = \{P_1, P_2, \ldots\ldots, P_k, \ldots\ldots, P_K\}$$

Where K is the total number of all possible paths. The set of paths {PATH} could be determined either by visual inspection of project network (for small project networks) or by following a systematic simple procedure to generate all possible paths. After identifying the activities comprising each path, values of the (y) parameter can be determined.

Having determined the set of all paths {PATH}, the network logic constraints could be specified. The duration length of any path must be less than or equal to the required project duration; λ. That is to say, for any path k,

$$\{d_1y_1 + d_2y_2 + \ldots\ldots + d_iy_i + \ldots\ldots + a_ny_n\}_k \le \lambda$$

Where d_i is the duration of activity (i) as given by Eq. (1). If overlap exists between any two consecutive activities along the path (k), summation of overlap values are to be deduced form the left hand side. Therefore, logic constraint for path (k) could be rewritten as follows:

$$\{d_1y_1 + d_2y_2 + \ldots\ldots + d_iy_i + \ldots\ldots + a_ny_n\}_k \le \lambda + SO_k$$

or,

$$\sum_{i=1}^{n} \{d_iy_i\}_k \le \lambda + SO_k \qquad (5)$$

Where SO_k is the algebraic Sum of Overlap values that may exist between any two activities along path k. Because activity duration can assume any feasible value between normal and crash durations, overlap values are consequently variable. Variable values of overlap are usually expressed as percentage of activity duration (Eldosouky et al 1991). The new model can accept overlap as a fixed value or as a percentage of activity duration. Path constraints of type (5) must be formulated for all possible paths. Therefore, the total number of required constraints is (n+K).

The general model can be summarized as follows:

Minimize: $$PDC = \sum_{i=1}^{n} \sum_{j=1}^{m_i} c_{ij} x_{ij}$$

Subject to:

(a) $$\sum_{j=1}^{m_i} x_{ij} = 1 \qquad i = 1, 2, \ldots\ldots\ldots, n$$

(b) $$\sum_{i=1}^{n} \{d_i y_i\}_k \le \lambda + SO_k \qquad k = 1, 2, \ldots\ldots\ldots, K$$

EFFICIENT REFORMULATION OF THE GENERAL MODEL

In practice, there are only limited number of ways to accelerate an activity, and thus only a finite number of discrete points on the activity time-cost relationship are defined as given by Eldosouky et al (1991). They concluded that the number of discrete points is limited and usually ranges from one to four points from the practical point of view. Therefore, the number of variables required for the general model is manageable even for large projects having many hundreds of activities. On the other hand, the major obstacle faced when applying the general model to even medium-sized projects is the large number of constraints required. Medium-sized projects can contain many thousands of paths which makes the general model of minor value. In the following sections, the unnecessary decision variables and constraints will be eliminated, thus making the modified model more efficient.

Decision Variables Reduction

Recalling Eq. (4) and rearranging, this yields:

$$x_{i1} = 1 - (x_{i2} + x_{i3} + \ldots\ldots\ldots + x_{im_i}) = 1 - \sum_{j=2}^{m_i} x_{ij} \qquad (6)$$

Since, for each activity, a single zero-one variable can assume a value of unity, the upper bound on the summation part of Eq. (6) is one. Mathematically, this condition can be expressed as:

$$\sum_{j=2}^{m_i} x_{ij} \le 1 \qquad (7)$$

Obviously, zero-one constraints of the form (7) are required only for those activities having more than one discrete point. Activities having single discrete point utility data are to be excluded in formulating zero-one constraints. Substituting from Eq. (6) into Eq. (1) and Eq. (2), this yields:

$$d_i = d_{i1} - \sum_{j=2}^{m_i} (d_{i1} - d_{ij}) x_{ij} \qquad (8)$$

$$c_i = c_{i1} + \sum_{j=2}^{m_i} (c_{ij} - c_{i1}) x_{ij} \qquad (9)$$

The term $(c_{ij}-c_{i1})$ represents the additional cost (over normal cost) incurred if an activity i) is speeded up at the duration d_{ij} beyond its normal duration d_{i1}. Substituting form Eq. (8) into Eq. (5), and rearranging yields:

$$\sum_{i=1}^{n}\sum_{j=2}^{m_i}\{(d_{i1}-d_{ij})x_{ij}y_i\}_k \geq \sum_{i=1}^{n}\{d_{i1}y_i\}_k - \lambda - SO_k \qquad (10)$$

Making use of these adjustments, the project direct cost would be:

$$PDC = \sum_{i=1}^{n} c_{i1} + \sum_{i=1}^{n}\sum_{j=2}^{m_i}(c_{ij}-c_{i1})x_{ij} \qquad (11)$$

The first term of Eq. (11) is a constant part which represents the normal project cost, while the second term denotes the extra cost resulting from accelerating some activities. With these adjustments, the number of decision variables are reduced by one for each activity. On the other hand, zero-one constraints are required only for those activities having more than one discrete point.

Redundant Paths Elimination

In practice, only small number of paths are dominant which must be considered in the process of expediting the project from normal to crash duration. The other paths are redundant and can be eliminated from the optimization model. Ahuja (1984) refers to this phenomenon as the criticality theorem. Redundant paths are those that will never be critical in any crashing cycle. If any path has a duration length (with all activities having normal durations) less than or equal to crash project duration (CPD) then this path is redundant. Redundant paths add nothing to the mathematical model except increasing the problem size. Redundant paths elimination makes the optimization model more effective, since unnecessary constraints are removed. Let the number of dominant (effective) paths are found to be (K_e) and the number of effective activities, which comprise the dominant paths, are (n_e). The following systematic procedure can be used to identify redundant paths:

- Perform forward pass of ordinary CPM analysis considering crash duration for all activities to determine crash project duration (CPD).
- Generate the set of all possible paths {PATH}. The procedure given in Appendix II can be used.
- Considering the normal duration for all activities, determine the duration length of all paths in the set {PATH}.
- Eliminate those paths whose duration length ≤ CPD. These are the redundant paths, the remaining ones are the dominant paths.
- Enumerate the activities comprising the dominant paths. They are the effective activities (n_e) which will be considered in formulating the mathematical model.

The final form of the modified model could be summarized as follows:

Minimize: Extra Cost $= \sum_{i=1}^{n_e}\sum_{j=2}^{m_i}(c_{ij}-c_{i1})x_{ij}$

Subject to: (c) $\sum_{j=2}^{m_i} x_{ij} \leq 1$ $\qquad i = 1, 2, \ldots, n_e$

$$(d) \qquad \sum_{i=1}^{n_e} \sum_{j=2}^{m_i} \{(d_{i1} - d_{ij}) x_{ij} y_i\}_k \geq \sum_{i=1}^{n_e} \{d_{i1} y_i\}_k - \lambda - SO_k \qquad k = 1, 2, \ldots, K_e$$

As all the used decision variables are zero-one, a zero-one programming subroutine should be used to solve the problem. Zero-one programming problem is a special case of integer programming problems, since all variables are restricted to have a value of zero or one only. Zero-one programming techniques have many advantages over integer programming (computational efficiency, storage, etc.). The FORTRAN subroutine given by Kuester and Mize (1972) is used in the present study after being recoded in QuickBASIC programming language Version 4.5.

The model requires as input precedence relationship of project activities, overlap between consecutive activities (if any), and activities discrete point utility data. Making use of these data, the project is analyzed to get both the all-normal and all-crash durations. An interface subroutine is prepared to use this information to establish automatically the objective function and the required constraints. Zero-one programming subroutine is then called to solve the mathematical model. For each feasible project duration, the optimization subroutine selects the optimum duration and cost for each activity. These data are then used by a simple CPM subroutine to determine scheduled timings corresponding to each project duration. The project time-direct cost curve is then determined, and consequently the optimum project duration can be specified

ILLUSTRATIVE EXAMPLE PROJECT

To illustrate the previously discussed concepts, consider the simple example project given by Ahuja (1984). The project is depicted by the network shown in Fig.1. Ahuja assumes linear activity time-cost relationship. In the present analysis, utility data are discretized at interval of unit duration and overlap values are assumed to suit the model requirements. Overlap values (FTS) are shown on the arcs linking consecutive activities. Utility data for each activity are given in Table 1, where the first value in each column represents the normal condition and the last value represents the crash condition. Duration values; (d), are given in weeks and cost values; (c), are given in US dollars. The cost values for all activities represent the extra cost associated with crashing an activity beyond its normal condition.

The project requires a total duration of 69 weeks to complete, if all activities are performed at their normal durations. However, the all-crash solution produces a project completion time of 57 weeks. Therefore, the two extreme project time limits are 69 and 57 weeks. The project network comprises 14 paths, and they are listed in Table 2. If the problem is formulated according to the general model, it requires 49 zero-one variables (number of discrete points for all activities), and 28 constraints (14 zero-one constraints and 14 network logic constraints).

All possible paths from the first activity to the last one are listed in Table 2. If an activity exists on a certain path, a value of 1 is inserted in the column belongs to that activity. Otherwise, the column is left blank. Assuming normal duration for all activities (considering the overlap between consecutive activities), the normal duration length (NDL) of each path is calculated and the results are given in the last column of Table 2. Comparing normal duration length of each path with CPD, it is apparent that only 4 paths are dominant and the other 10 paths are redundant. The dominant paths are those numbered 3, 8, 10, and 14. The activities comprising the dominant paths are 1, 2, 5, 6, 7, 8, 9, 10, 11, 13, and 14. Decision zero-one variables are needed for activities 6, 7, 8, 9, 10, 11, and 13 only since the others have utility data with single discrete point (i.e., activities 1, 2, 5, and 14).

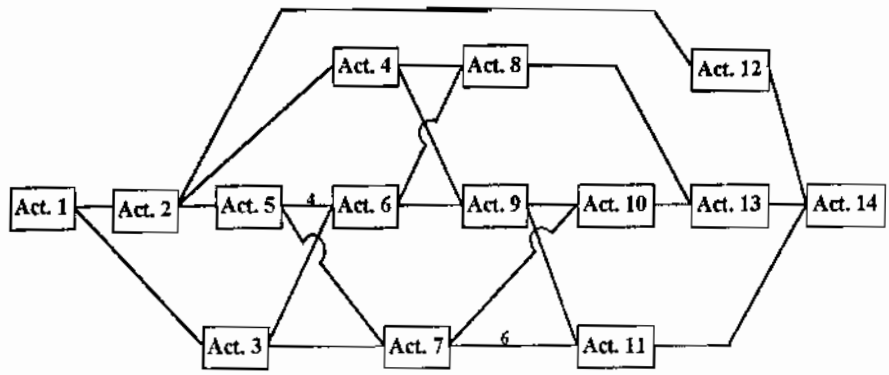


Fig. 1. Precedence Network of the Example Project

Table 1. Activities Utility Data

Act. 1		Act. 2		Act. 3		Act. 4		Act. 5		Act. 6		Act. 7		Act 8		Act. 9		Act 10		Act 11		Act. 12		Act. 13		Act. 14	
d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c	d	c
5	0	10	0	5	0	5	0	15	0	10	0	19	0	19	0	10	0	10	0	12	0	10	0	10	0	0	0
				4	30	4	65			9	80	18	70	18	75	9	50	9	85	11	40	9	45	9	50		
				3	60	3	130			8	160	17	140	17	150	8	100	8	170	10	80	8	90				
				2	90	2	195			7	240	16	210	16	225			7	225								
						1	260					15	280	15	300			6	340								
												14	350					5	425								
												13	420					4	510								
												12	490														
												11	560														

Table 2. Project Network Paths

Path No.	Activity														NDL
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	1	1		1				1					1	1	49
2	1		1			1		1					1	1	49
3	1	1			1	1		1					1	1	65
4	1	1										1		1	25
5	1		1				1			1			1	1	49
6	1	1		1					1	1			1	1	50
7	1		1			1			1	1			1	1	50
8	1	1			1		1			1			1	1	69
9	1		1				1				1			1	35
10	1	1			1	1			1	1			1	1	66
11	1	1		1					1		1			1	42
12	1		1			1			1		1			1	42
13	1	1			1		1				1			1	55
14	1	1			1	1			1		1			1	58

Activities Duration

Let the first subscript associated with zero-one variables refers to activity number and the second refers to discrete point number for that activity. For instance, x_{74} denotes the zero-one variable belonging to the fourth discrete point of activity number 7. Activities duration in terms of zero-one variables would be:

$$d_6 = 10 - x_{62} - 2x_{63} - 3x_{64}$$
$$d_7 = 19 - x_{72} - 2x_{73} - 3x_{74} - 4x_{75} - 5x_{76} - 6x_{77} - 7x_{78} - 8x_{79}$$
$$d_8 = 19 - x_{82} - 2x_{83} - 3x_{84} - 4x_{85}$$
$$d_9 = 10 - x_{92} - 2x_{93}$$
$$d_{10} = 10 - x_{102} - 2x_{103} - 3x_{104} - 4x_{105} - 5x_{106} - 6x_{107}$$
$$d_{11} = 12 - x_{112} - 2x_{113}$$
$$d_{13} = 10 - x_{132}$$

For the first dominant path (i.e., path number 3), the logic constraint would be:

$$x_{62} + 2x_{63} + 3x_{64} + x_{82} + 2x_{83} + 3x_{84} + 4x_{85} + x_{132} \geq (69-\lambda-4)$$

or,

$$x_{62} + 2x_{63} + 3x_{64} + x_{82} + 2x_{83} + 3x_{84} + 4x_{85} + x_{132} \geq (65-\lambda)$$

The mathematical statement of the problem would be:

Objective Function:

Min. : Extra Cost = $80x_{62} + 160x_{63} + 240x_{64} + 70x_{72} + 140x_{73} + 210x_{74} + 280x_{75} + 350x_{76} + 420x_{77} + 490x_{78} + 560x_{79} + 75x_{82} + 150x_{83} + 225x_{84} + 300x_{85} + 50x_{92} + 100x_{93} + 85x_{102} + 170x_{103} + 255x_{104} + 340x_{105} + 425x_{106} + 510x_{107} + 40x_{112} + 80x_{113} + 50x_{132}$ (12)

Subject To:

Zero-One Constraints:

$x_{62} + 2x_{63} + 3x_{64}$	≤ 1	(13)
$x_{72} + 2x_{73} + 3x_{74} + 4x_{75} + 5x_{76} + 6x_{77} + 7x_{78} + 8x_{79}$	≤ 1	(14)
$x_{82} + 2x_{83} + 3x_{84} + 4x_{85}$	≤ 1	(15)
$x_{92} + 2x_{93}$	≤ 1	(16)
$x_{102} + 2x_{103} + 3x_{104} + 4x_{105} + 5x_{106} + 6x_{107}$	≤ 1	(17)
$x_{112} + 2x_{113}$	≤ 1	(18)
x_{132}	≤ 1	(19)

Network Logic Constraints:

$x_{62} + 2x_{63} + 3x_{64} + x_{82} + 2x_{83} + 3x_{84} + 4x_{85} + x_{132}$	$\geq (65-\lambda)$	(20)
$x_{62} + 2x_{63} + 3x_{64} + x_{92} + 2x_{93} + x_{102} + 2x_{103} + 3x_{104} + 4x_{105} + 5x_{106} + 6x_{107} + x_{132}$	$\geq (66-\lambda)$	(21)
$x_{62} + 2x_{63} + 3x_{64} + x_{92} + 2x_{93} + x_{112} + 2x_{113}$	$\geq (58-\lambda)$	(22)
$x_{72} + 2x_{73} + 3x_{74} + 4x_{75} + 5x_{76} + 6x_{77} + 7x_{78} + 8x_{79} + x_{102} + 2x_{103} + 3x_{104} + 4x_{105} + 5x_{106} + 6x_{107} + x_{132}$	$\geq (69-\lambda)$	(23)

The upper and lower bounds on the parameter λ are the normal and crash project durations, which are 69 and 57 weeks, respectively. The normal direct cost of the project is $6,600. The Computer program was run for all feasible solutions of project duration. Due to discrete nature of activities time-cost relationship, feasible solutions are found for all project durations. The computer solution of the problem, based on the new mathematical model, is given in Table 3.

If overlap between activities are omitted as given by Ahuja (1984), the computer solution of the problem against the manual solution as given by Ahuja is shown in Table 4. For comparison purpose, solutions for project durations given by Ahuja are only given. It is apparent that manual solution does not guarantee optimal solution. This is because in manual methods, once an activity is crashed in a certain compression cycle, the activity is not allowed to be decompressed in subsequent cycles. Decompression of certain activities and compression of other activities may produce optimal solution. Optimization modeling of the problem permits decompression of crashed activities to minimize crashing cost on condition that network logic is satisfied.

Table 3. Computer Solution of the Example Problem

Project Duration (wks)	Extra Cost ($)	Direct Cost ($)
69	---	6,600
68	50	6,650
67	120	6,720
66	190	6,790
65	260	6,860
64	345	7,945
63	495	7,095
62	645	7,245
61	795	7,395
60	955	7,555
59	1115	7,715
58	1275	7,875
57	1435	8,035

Table 4. Manual (Ahuja) Vs Optimum Solution

Project Duration (Weeks)	Direct Cost ($) (Ahuja)	Direct Cost ($) (New Model)
70	6,600	6,600
69	6,650	6,650
68	6,700	6,700
65	7,150	7,115
61	7,790	7,755

MODEL COMPARISON

A relatively medium-sized project was analyzed using the new developed model. The project involves the construction of an earth dam and consists of 34 activities (Thompson 1982). The project network comprises 35 paths. Analysis of the project to produce project utility data was given by Ammar (1992). The number of discrete points for all activities is 75. If the project is solved using the model developed by Eldosouky et al (1991), the required number of variables is 109 whereas the number of constraints is 82.

Using the new developed model in the general form, 75 variables and 35 constraints are required. Applying the modified new model, the dominant paths are 21 only, while the effeetive activities are only 12. In this case, the required number of variables and constraints are 23 and 21 respectively. These data are summarized and are given in Table 5. It is apparent that the new model reduces significantly the problem size.

Table 5. Model Comparison

Model	Number of Variables	Number of constraints
Eldosouky et al	109	82
New Model	23	21
%Reduction	79%	74%

CONCLUSIONS

A mathematical optimization model has been developed to establish the optimum project time-direct cost relationship. The problem is formulated in the standard form of zero-one programming. The objective function is to minimize project cost while the model is constrained by zero-one and network logic constraints. The model uses path constraints to satisfy logic sequence of project activities. All possible paths, which comprise the project network, are first generated, and then redundant paths are identified. Only activities comprising dominant paths (effective activities) are considered. Network logic constraints are formulated only for dominant paths. Redundant paths as well as unnecessary decision variables are eliminated, thus makes the model very powerful.. The discrete point utility data for activities are used. Overlapping between project activities is, also, permitted. The model has the following features: (1)It guarantees the optimum solution, (2)Precise activity time-cost relationship is used, (3)Overlapping activities are permitted, (4)The model is entirely formulated using an interface computer subroutine, and (5)The model significantly reduces the problem size.

APPENDIX I: REFERENCES

Adeli, H., and Karim, A. (1997). "Scheduling/Cost Optimization and Neural Dynamies Model for Construction". J. Constr. Engrg. and Mgmt., ASCE, 123(4), 450-458.

Ahuja, H. N. (1984). Project Management Techniques in Planning and Controlling Construetion Projects. John Wiley and Sons, New York.

Ammar, M. A. (1992). Network Compression with Discrete Utility Data. M. Sc. Thesis, Structural Engineering Dept., Faculty of Engineering,Mansoura University.

Antill, J. M., and Woodhead, R. W. (1982). Critical Path Method in Construetion Practice. 3rd Edition, Wiley Interscience Publication, New York.

Cusak, M (1985). "Optimization of Time and Cost." International Journal of Project Management, 3(1), 50-55.

Crowston, W. B. (1970). "Decision CPM: Network Reduction and Solution," Operation Research Quarterly, Vol. 21, 435-452.

Eldosouky, A. I., Abdelreheem, A. H., and Ammar, M. A. (1991). "An Optimization Approaeh to Project Cost/Time Problem." Fourth Arab Structural Engineering Conference, Cairo, Egypt, Part V, V61-V72.

Fulkerson, D. R. (1961). "A network Flow Computation for Project Cost Curves." J. Management Science, 7(2), 167-178.

Harris, R. B. (1978). Precedence and Arrow Networking Techniques for Construction. John Wiley and Sons, New York.

Kapur, K. C. (1973). "An Algorithm for Project Cost-Duration Analysis Problem with Quadratic and Convex Cost Functions." AIIE Transactions, 5(4), 314-322.

Kuester, J. L., and Mize, J. H. (1973). Optimization Techniques with FORTRAN. McGraw Hill, New York.

Li, H., and Love, P. (1997). "Using Improved Genetic Algorithms to Facilitate Time-Cost Optimization". J. Constr. Engrg. and Mgmt., ASCE, 123(3), 233-237.

Liu, L., Burns, S. A., and Feng, C. W. (1995). "Construction Time-Cost Trade-Off Analysis Using LP/IP Hybrid Method". J. Constr. Engrg. and Mgmt., ASCE, 121(4), 446-454.

Meyer, W. L., and Shaffer, L. R. (1965). "Extending CPM for Multiform Project Time-Cost Curves." Journal of the Construction Division, ASCE, 91(1), 45-67.

Panagiotakopoulos, D. (1977). "Cost-Time Model for Large CPM Project Networks." Journal of the Construction Division, ASCE, 103(2), 201-211.

Perera, S. (1982). "Compression of Overlapping Precedence Network." Journal of the Construction Division, ASCE, 108(1), 1-12.

Thompson, P. (1981). Organization and Economics of Construction. McGraw-Hill Book Company.

APPENDIX II: PROCEDURE FOR GENERATING NETWORK PATHS

Assuming that the number of successors for an activity (i) is NS(i) and the succeeding activities are SUC(i,j), where j = 1, 2, ..., NS(i). The procedure consists of the following systematic steps:

1. Determine the sequence step for each activity. Sequence step is defined by Harris (1978) as "The earliest logical position in the network that an activity can occupy while maintaining its proper dependencies".
2. Sort project activities in increasing order according to sequence step number.
3. Initially consider the first activity, the number of paths at this stage (K) equals number of successors of the first activity, NS(1), while the paths are the links between the first activity and its successors. Therefore:

$$K = NS(1)$$
$$P_1 = \{A(1), SUC(1,1)\}$$
$$P_2 = \{A(1), SUC(1,2)\}$$
$$\cdot$$
$$\cdot$$
$$P_K = \{A(1), SUC(1,K)\}$$

4. Consider the following activity (i):
   a. Check the last element in the each generated path; k, where k=1, 2, ..., K. If k = K, then got to step 5.
   b. If the last element in a path is identical to the activity under consideration then go to step 4-c, otherwise go to step 4-d.
   c. The number of paths (K) is to be increased by NS(i)-1. The new paths are identical to the checked path. Then the successors of the considered activity are to be added for the checked path and the new generated paths.
   d. If the last element in the checked path differs from the activity under consideration, then move the last element one position to the right and replace it by zero.
5. Repeat step 4 for all Activities.

After step 5, the set of all possible paths; {PATH}, and the total number of paths; (K), are specified.