Applied Mathematical Sciences, Vol. 4, 2010, no. 42, 2081 - 2094

The Discrete Time-Cost-Quality Trade-off Problem Using a Novel Hybrid Genetic Algorithm

N. Shahsavari Pour

Department of Industrial Engineering
Science and Research Branch, Islamic Azad University, Tehran, Iran
Email: Shahsavari_n@alum.sharif.edu

M. Modarres

Department of Industrial Engineering
Sharif University of Technology, Tehran, Iran

Mir B. Aryanejad

Department of Industrial Engineering
Iran University of Science and Technology, Narmak, Tehran, Iran

R. Tavakoli Moghadam

Department of Industrial Engineering
University of Tehran, Tehran, Iran

Abstract

Time, cost and quality are among the major objectives of any project .In recent years, the demands of project stakeholders regarding reductions in the total cost and time of a project along with achieving the desirable quality of the project have risen significantly. This leads researchers to developing models that incorporate the quality factor to previously existing time-cost trade-off models.
This study presents a model for the discrete time-cost-quality trade-off problem.In this model there are a number of execution modes to select for each activity, and the best execution mode (t, c, q) of the activities should be determined to optimize total cost and time objectives, subjected to project quality and other constraints.
To solve the model, a new meta-heuristic algorithm called *NHGA* is introduced .

This algorithm is much more efficient than classic genetic algorithm (*GA*) for solving the above-mentioned model. By presenting a case example, the efficiency of the proposed algorithm for solving the model and its flexibility for project managers' decision making is demonstrated.

Keywords: Time cost quality trade-off, Project scheduling, Meta heuristic algorithms

1 Introduction

In critical path method (*CPM*) computations, it is assumed that all activities can be performed in the normal duration .It is sometimes required for a project to be completed before the normal due time .Naturally, in such cases, the duration of performing some of the activities must be decreased. This decrease in time is achieved by an increase in resources or a change in the execution methods of the activities and by increased costs .This, on the other hand, leads to a change in the quality of the activities and consequently the quality of the project .The aim of time-cost-quality trade-off is to select a set of activities for crashing, such that the total cost and time of the project is minimized while the total quality of the project is maximized.

In most time-cost trade-off models, the relationship between the decrease in activity time and the increase in activity cost is assumed to be in the form of a linear function, and the aim is to finish the project at the expected due time and by minimizing the total cost (direct and indirect cost) .To solve such a linear model, numerous methods have been put forward [9,14,15,17,21,29,31]. Although other models have also been proposed for the cost function, such as concave cost function [12] , convex cost function[20] and continuous activity cost function [24], for real world applications the discrete time-cost trade-off problem (*DTCTP*) applies. Unlike linear models, there are few researches on the *DTCTP* .This is while any discrete time-cost relationship is more relevant to real world projects since resources, execution methods and technology types in real world projects are discrete items [26].

The solution methods of *DTCTP* are classified into exact and heuristic .Exact solution methods are based on dynamic programming [17,28], enumeration algorithms [25], or project network decomposition [8]. None of the exact solution methods can solve large problems with large numbers of activities .In fact, *DTCTP* is known as an *NP-hard* problem [7] .No case was found in exact solution methods to have the required efficiency for solving *DTCTP*, but there are many researches based on heuristic solution methods for solving *DTCTP* [2,3,5,6,13,19,23,30,34].

The total quality of the project is affected by the project crashing. Thus, it is necessary to include the quality factor in the time-cost trade-off problem, leading to the time-cost-quality trade-off problem. In the discrete case, this problem is represented by *DTCQTP* (discrete time-cost-quality trade-off problem); in which each project activity can be executed in one of many execution modes. The execution mode of any activity is related to the resources, execution methods and execution technology of the activity. For each mode of an activity, there is a triple combination (*t,c,q*) indicating the time, cost, and quality of the specific activity in that mode, such that $t\epsilon Z, c\epsilon Z, 0 < q\epsilon Z < 100$.

This kind of problem is called the multi-execution mode for activities in the literature, and the best execution mode (*t, c, q*) of the activities should be determined to optimize multi objective, subjected to some constraints [18].
In this research, measuring the quality of each activity as well as the total quality of the project are adapted the procedures used in [4].

The aim of the time-cost-quality trade-off optimization problem is to complete the project in minimum time with minimum cost and maximum quality .Little research has been carried out on this optimization problem [1,4,10,18,22,32,33] . By removing some non-realistic assumptions, new more realistic methods can be presented .In order to eliminate the linear relationship between time-cost and also between time-quality, this study considers the problem in the discrete case (i.e., *DTCQTP*), but unlike previous studies, it is not looking for Pareto solutions. This study is different from the previous research in two respects: different modeling of the problem, and innovation in solving the problem.

Different modeling of the problem : Considering the fact that most project managers carry out crashing of the activities with the aim of decreasing the total cost and time of the project alongside achieving acceptable quality, the model presented in this study applies to this aim, solving this model leads to reaching to an optimal point in three-dimensional space. By changing the value for the allowable quality and re-running the algorithm, a different optimal point can be obtained by the project manager. Having these optimal points on hand, project managers can make decisions more easily than when they are facing with a set of Pareto solutions .

Innovation in solving the model : In this paper, a novel hybrid genetic algorithm (*NHGA*) is introduced for solving the model, which is more efficient compared to other meta-heuristic algorithms such as the genetic algorithm .The high speed of the algorithm and quick convergence of the solutions make this model and its solution algorithm suitable for large projects with large numbers of activities.

2 Problem Formulation

In this research, the project is defined by a direct acyclic graph $G = (V,E)$, in which V (vertex) and E (edge) are the sets of nodes and arcs, respectively .Arcs and nodes represent activities and events, respectively .$G\ (V,E)$ is demonstrated as a matrix $A_{m\times n}$ in which m and n denote the number of nodes and arcs.
Matrix A_{mxn} is called the node -arc incidence matrix for graph $G(V,E)$. Matrix A has one row for each node of the network and one column for each arc .Each column of A contains exactly two nonzero coefficients :“+1” and “-1”. The column corresponding to arc j contains “+1” if i is the node starting arc j, “-1” if i is the node ending arc j, and “0” otherwise.

$A = [a_{ij}]$

$$a_{ij} = \begin{cases} 1 & if\ node\ i\ starts\ arc\ j \\ -1 & if\ node\ i\ ends\ arc\ j \\ 0 & otherwise \end{cases}$$

i=1,…………m *j =1,…………,n*

Each project activity *(E_j)* has different execution modes *(M_j)*, in which any $k\epsilon M_j$ involves time t_{jk}, cost c_{jk}, and quality q_{jk} of activity *j* .If *k* and *r* are two modes for activity *j* and *k<r*, it is assumed that $t_{jk}>t_{jr}$, $c_{jk}<c_{jr}$, and $q_{jk}\neq q_{jr}$ .Although in the literature it is assumed that a decrease in the activity time leads to a decrease in activity quality, it is noteworthy that, in real world projects, this is not always the case .For instance, if a new technology is employed to reduce the time required for an activity, this reduction can be accompanied by an increase in quality and cost.

The aim of this research was to obtain the optimal combination (t_{jk}, c_{jk}, q_{jk}) of each activity for crashing the project network, such that along with reducing the total time of the project, the total cost of the project (direct plus indirect) is minimized while the total quality of the project does not fall below a desired level . The following notation is used to describe the *DTCQTP*:

C_T: Total cost of project (direct plus indirect);

T_t: Total duration of project;

C_{Id}: Project indirect cost per time unit;

M_j: Set of available execution modes for activity *j;*

c_{jk}: Direct cost of activity *j* when performed the k_{th} execution mode;

t_{jk}: Duration of activity *j* when performed the k_{th} execution mode;

q_{jlk}: Performance of quality indicator (*l*) in activity *j* performed the k_{th} execution mode;

y_{jk}: The binary mode indicator which is 1 when mode *k* is assigned to activity *j* and 0 otherwise;

w_j: Weight of activity *j* compared to other activities in the project, $\sum_{j=1}^{n} w_j = 1$

$\acute{w}_{jl}$: Weight of Quality indicator (*l*) compared to other *(L)* indicator, in activity *j*, $\sum_{l=1}^{L} \acute{w}_{jl} = 1$

Q_{allow}: Lower bound for Project quality;

x_j: The binary path indicator that determines the flow on the arc *j* . if $x_j = 1$ the activity *j* is in the path. While $x_j = 0$ means not;

a_{ij} The entry of incidence matrix, as defined before;

b_i: The available supply in node *i*;

$T_{cpm}^{\bar{K}}$: Critical path duration if set of modes $\bar{K}$ are assigned to activities;

Mixed integer programming is used for modeling *DTCQTP*:

$$Min.C_t = \left(\sum_{j=1}^{n} \sum_{k\in M_j} c_{jk} * y_{jk}\right) + C_{Id} * T_{cpm}^{\bar{k}} \quad (1)$$

$$Min.T_t = T_{cpm}^{\bar{k}} \quad (2)$$

St:

$$\sum_{j=1}^{n} w_j \sum_{l=1}^{L} \acute{w}_{jl} \sum_{k\in M_j} q_{jlk} * y_{jk} \geq Q_{allow} \quad (3)$$

$$\sum_{k\in M_j} y_{jk} = 1 \qquad j = 1,2,\dots\dots\dots,n \quad (4)$$

$$y_{jk} \in \{0,1\} \qquad \forall\, j,k \quad (5)$$

The *CPM* problem can be considered as the reverse of the shortest path problem of the network, so $T_{cpm}^{\bar{k}}$is formulated considering the network matrix $A_{m\times n}$ as follows:

$$T_{cpm}^{\bar{k}} = Max \sum_{j=1}^{n} x_j \sum_{k\in M_j} t_{jk} * y_{jk} \quad (6)$$

St:

$$\sum_{j=1}^{n} a_{ij} * x_j = b_i \qquad i = 1,2,\dots\dots\dots,m \quad b_i = \begin{Bmatrix} 1 & if \;\; i=1 \\ -1 & if \;\; i=m \\ 0 & otherwise \end{Bmatrix} \quad (7)$$

$$x_j = \in \{0,1\} \qquad \forall\, j \quad (8)$$

$$\bar{K} = \{k_1, k_2, \dots\dots, k_n\} \quad (9)$$

Objective functions (1) and (2) minimize the project's total costs and duration respectively. Constrain (3) enforces that the total quality of project does not fall bellow the allowable level. In (4) one and only one execution mode is assigned to each activity and equations (5) and (8) are sign constrains. Objective function (6) and its constraints calculate duration of critical path. Constraint (7) defines the feasible path in the project network and is used for the precedence relations between activities. Constraint (9) enforces that activities on critical path are performed by set of modes $\bar{K}$.

3 Solution Procedure

In the *DTCQTP*, there are a number of execution modes to select for each activity . If the number of project activities is n and there are k execution modes for each activity to choose from, then there are k^n solution series, which is a very large search space for solving the problem .Therefore, it is necessary for solving the problem and obtaining the optimal solution to make use of evolutionary algorithms.

3.1 *Evolutionary Algorithms*

Evolutionary Algorithms such as genetic algorithm are proved to be able to balance exploration and exploitation of the solution search space .The basic operation of a genetic algorithm is simple .First, a population of possible solutions to a problem is developed .Then, the better solutions are recombined with each other to from some new solutions for the next generation .In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (crossover and mutation) to form a new population .The new population is then used in the next iteration of the algorithm .Finally, the new solutions are used to replace the poorer of the original solutions and the process is repeated . Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population [11].

3.2 *Novel Hybrid Genetic Algorithm*

In this research, in order to solve the problem, a new meta-heuristic algorithm, the novel hybrid genetic algorithm (*NHGA*), is introduced .This algorithm features much better efficiency in comparison with the genetic algorithm .The high speed of the algorithm and quick convergence of the solutions make it suitable for solving the above problem.

NHGA was developed by some modifications in *GA* (Fig .1) .These modifications are as follows:

- addition of hill climbing
- reduction in hill climbing selection rate by introducing a new function
- reduction in mutation rate in the next generations by using a function

3.3 Model implementation

A chromosome to the problem is a set of integer values (Genes) which shows the selected mode for each activity that is between 1 and the number of defined modes for that activity.

$$[m_1, m_{2,} ....m_n]: m_j \in [1, M_j]$$

In each chromosome, only one mode is selected for each activity, which leads to the combination of *(t,c,q)* for executing the activity .When reading the numbers for all genes of a chromosome is completed, then an execution mode is selected for all project activities and a chromosome is produced with feasible genes.

The steps of *NHGA* for solving the problem under consideration are as follows:

Step 1 :First, problem data are read and then *N* chromosomes with feasible genes are randomly produced as primary solutions .Problem data include project data and *NHGA* parameters.

The project data include:

- Project network matrix $(A_{mxn}=[a_{ij}]);$
- Available execution modes for each activity j and their expected impact on the activity cost, duration and quality (Mj and (c_{jk} ,t_{jk} ,q_{jk}));
- Weight of activity j compared to other activities in the project (W_j);
- Project indirect cost per time unit (C_{Id});
- Lower bound for project overall quality (Q_{allow});

The required *NHGA* parameters include:

- String size (n);
- Number of generation (G);

- Population size (N);
- Hill climbing Rate;
- Weight of exponential function (W_{exp});
- Weight of linear function (W_{lin});
- Two point crossover rate;
- Uniform crossover rate;
- Mutation rate;

Step 2 :For each one of the N chromosomes produced, the total direct cost of project $C_{d(s)}$, the total time of project $T_{(s)}$ and the total quality of project $Q_{(s)}$are calculated as follows:

Total direct cost of project :the sum of direct costs of all project activities.

$$C_{d_{(s)}} = \sum_{j=1}^{n} c_{sj} \qquad s = 1,2,\dots\dots\dots,N \tag{10}$$

c_{sj}: Direct cost of activity j for each chromosome S;

Total time of project :the sum of activities duration in the critical path.

$$T_s = Max \sum_{j=1}^{n} x_j * t_{sj} \qquad s = 1,2,\dots\dots\dots,N \tag{11}$$

St:

(7), (8),(9)

t_{sj}: Duration of activity j for each chromosome S;

Total quality of project :weighted sum of activities quality

$$Q_{(s)} = \sum_{j=1}^{n} w_j * q_{sj} \qquad q_{sj} = \sum_{l=1}^{L} \acute{w}_{jl} * q_{sjl} \qquad s = 1,2,\dots\dots\dots,N \tag{12}$$

q_{sj}: Quality of activity j for each chromosome S;

q_{sjl}: Performance of quality indicator (l) in activity j for each chromosome S;

The quality calculated for each chromosome should be checked at this step not to be lower than the allowed quality (Q_{allow}), so that the chromosome has feasible genes.

Step 3 :Determining the fitness function (F_s) and the probability of selection (P_s) for each parent chromosome *"S"* using the following equations: (13), (14)

$$F_{(s)} = C_{d_{(s)}} + C_{Id} * T_s - \left(C_{d_{min}} + C_{Id} * T_{min}\right) + 1 + \left[\!\left[W_t * C_{d_{min}}/T_{min}\right]\!\right] * \left[(T_s - T_{min})/\left(C_{d_{(s)}} - C_{d_{min}} + 1\right)\right] \tag{13}$$

$$P_{(s)} = \frac{\frac{\sum_{s=1}^{N} F_{(s)}}{F_{(s)}}}{\sum_{s=1}^{N} \frac{\sum_{s=1}^{N} F_{(s)}}{F_{(s)}}} \tag{14}$$

$C_{d_{min}}$: Minimum direct cost of population;

T_{min} : Minimum time of population; W_t: Scaling factor;

For the first objective (1), one attempts to obtain the best solution, which is absolutely superior to second objective (2). Therefore equation (13) is introduced for fitness function. If $\left(C_{d_{min}} + C_{Id} * T_{min}\right)$is not present in equation (13), then f_s's will become large numbers, P_s's will be so close to each other, and finally the selection process will lose the necessary efficiency.If f_s were equal to zero, then it would be impossible to calculate P_s .Hence, +1 is present in the equation (13).

Considering the fact that chromosomes with lower f_s are more desirable, P_s should be defined so that the lower f_s, the higher the probability of selecting chromosome *"S"*. So, equation (14) is introduced for P_s.

Step 4 :producing offspring chromosomes from parent chromosomes to enter the next generation is done at this step .Before applying crossover and mutation operators, first a small number of $(c_{(h)})$ of the best parents are directly transferred to the next generation (hill climbing).The number of $(c_{(h)})$, however, decreases from one generation to the next .The decline rate function is a combination of a linear function and an exponential function with pre-specified weights.

$$F_{HillRate} = \frac{w_{exp}*\left(1-\left(e^{(n_g-G)}\right)\right)+w_{line}*\left(1-\frac{n_g}{G}\right)}{w_{exp}+w_{line}} \tag{15}$$

Where G is the number of generations, n_g is the generation number index, W_{exp} is the weight of the exponential function, and W_{line} is the weight of the linear function.The direct transfer of the best parents to the next generation with the decline rate function (15) is an innovation for improving GA for this problem, which increased the efficiency of the algorithm .After hill climbing, in order to produce the rest of the offspring $(c_{(t)})$, crossover $(c_{c(t)})$ and mutation $(c_{m(t)})$ operators should be applied.The operators were designed such that, after they are applied, the chromosome genes are still feasible .In this problem, for the number of activities less than 50, combination of a two-point crossover and a uniform crossover with pre-specified weights was used and for the number of activities greater than 50, only uniform crossover was used. Uniform crossover has been introduced by [16] .Also, considering the number of activities, one point to multi-point mutation was used .For instance, in a project with nine activities, two random chromosomes with feasible genes can be as follows:

Parent1=[2,1,5,3,2,4,1,5,3]
Parent2=[4,3,2,5,4,1,3,2,5]

In this example, since the number of activities is small, two-point and uniform crossover and two-point mutation were used .The offspring produced from these parents by applying the above-mentioned operators are as follows:

Two-point crossover with random points (E_1=4,E_2=7)
Offspring1=[2,1,5,5,4,1,3,5,3]
Offspring2=[4,3,2,3,2,4,1,2,5]
Uniform crossover with random mask chromosome *[1,1,0,1,0,0,1,0,0]*
Offspring1=[4,3,5,5,2,4,3,5,3]
Offspring2=[2,1,2,3,4,1,1,2,5]
Two-point mutation with random points (E_1=6,E_2=9)
Offspring1=[2,1,5,3,2,3,1,5,1]

It should be mentioned that the mutation rate is decreasing and uses function (16), so that in the final generation, the mutation rate will be zero.

$$F_{MuRate} = 1 - \frac{n_g}{G} \tag{16}$$

Step 5 :Repeat steps 2-4 until the chromosomes do not change from one generation to the next.

The steps of *NHGA* are summarized in Fig .1.

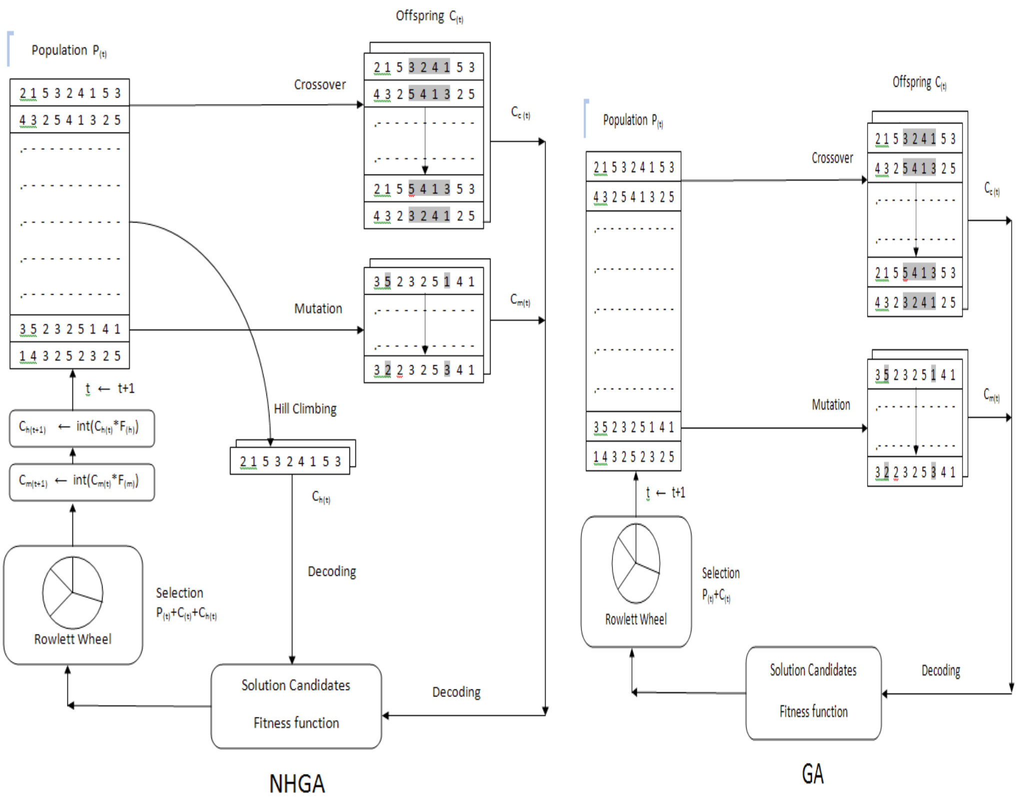

Fig.1. *NHGA* and *GA* flowchart

4 Application Example

As an example, a project including nine activities is presented in this section (Fig .2) .The network matrix is as presented in Table 1 .Each activity has different execution modes, each of which has the time, cost, and quality of the activity, as presented in Table 2 .The effect weight of each activity in the total quality *(w_j)* is also considered in Table 2.

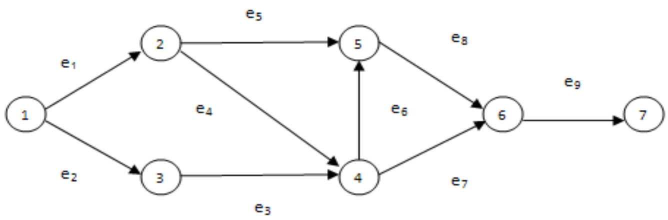

Fig.2. Project network

$$
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{array} \\
\left[\begin{array}{ccccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{array}\right]
\end{array}
$$

Table.1. Network Matrix

mode	Activities	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
1	**T**	7	8	8	10	14	8	11	11	11
	C	160	140	110	100	160	130	150	140	150
	Q	90	85	90	88	92	85	87	91	90
2	**T**	6	7	7	9	13	7	10	10	10
	C	180	150	120	130	170	140	180	150	170
	Q	85	82	85	90	90	82	90	88	88
3	**T**	5	6	6	8	12	6	9	9	9
	C	190	170	140	140	180	150	190	160	180
	Q	80	80	84	85	86	80	85	85	85
4	**T**	4	5	5	7	11	5	8	8	8
	C	200	180	150	150	200	170	200	170	200
	Q	70	75	80	75	70	85	90	75	90
5	**T**	3	4	4	6	10	4		7	
	C	230	200	170	165	220	190		265	
	Q	85	80	90	80	80	90		85	
6	**T**					9				
	C					240				
	Q					90				
	W_{qi}	0.1	0.1	0.14	0.11	0.12	0.15	0.08	0.12	0.08

Table.2. Execution modes of activities

The model is programmed in the Microsoft Excel using Visual Basic Application (*VBA*) .The project data including Tables (1,2), IC =20, and Q_{allow} =84 are entered . In this example, there are 1500000 solutions. To obtain the optimal solution, the proposed method was used .As the number of activities is small in the example, combination of a two-point crossover and a uniform crossover with pre-specified weights, and two-point mutation have been used.

NHGA parameters were set as following:

G=100, N=100, *two-point crossover rate*=0.6, *uniform crossover rate*=0.2, *mutation rate*=0.2, *hill climbing rate* =0.2, W_{exp}=8, W_{line} =2.

The program was run on a Pentium 4 *PC* with *CPU* 2.8 *GHz*, which took 227 seconds, and chromosome [4,2,2,1,1,5,1,4,4] and its corresponding project time, cost, and quality (T_t=34, C_d=1440, Q=84.5) were obtained as the output. Also total cost of project was obtained (C_t=2120).The project manager then obtained other optimal points by increasing Q_{allow}, which are presented in the results in Table 3 and Fig.3.

Q	C_t	T_t	C_d	*Solution chromosome*									Q_{allow}
84.48	2120	34	1440	4	2	2	1	1	5	1	4	4	0-84
86.18	2120	35	1420	3	2	1	1	1	5	1	4	4	85
86.18	2120	35	1420	3	2	1	1	1	5	1	4	4	86
87.48	2120	37	1380	1	1	1	1	1	5	1	4	4	87
88.18	2130	37	1390	2	1	1	1	1	5	1	3	4	88
89.04	2140	39	1360	1	1	1	1	1	5	1	2	4	89
89.4	2150	40	1350	1	1	1	1	1	5	1	1	4	89.2
89.62	2180	40	1380	1	1	1	2	1	5	1	1	4	89.6
89.86	2210	40	1410	1	1	1	2	1	5	2	1	4	89.8

Table.3. Final output of program

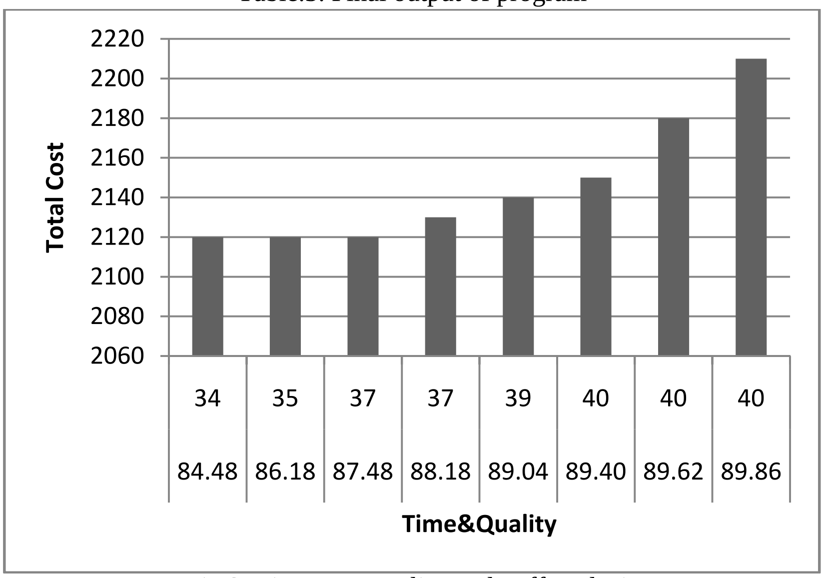

Fig.3. Time-cost-quality trade-off analysis

The project manager can make decisions by using the information in Table.3 and Fig.3 and analyzing the internal and external conditions of the project.

5. Calculations

Many project managers perform project crashing with the aim of reducing total costs and time along with achieving desirable quality .The model proposed in this paper addressed this real world issue .In order for the model to be as realistic as possible, the problem was considered in the discrete case (*DTCQTP*) .In this problem, each project activity can be executed in one of several modes. The execution mode of any activity was related to the resources, execution methods and execution technology of the activity. For each mode of an activity, there was a triple combination *(t,c,q)* indicating the time, cost, and quality of the specific activity in that mode. So, the point is which mode each activity should be executed in so that project time is reduced while total cost of project is minimized and project quality does not fall below a certain allowed level .Solving the problem gave an optimal solution including the time, cost, and quality of the project. By changing the allowable quality level for the project and re-running the algorithm, other optimal solutions could be obtained .Having these optimal solutions on hand, and analyzing the environmental conditions, project managers could make decisions effectively.

To solve the problem in this paper, *NHGA* was introduced, which not only takes much less time than classic *GA* to reach the solution, but also featured much higher accuracy compared to classic *GA*.

The example in this paper was once solved on a *PC* using classic *GA* and then using *NHGA* .With classic *GA*, (N=500 and G=100) had to be included as classic *GA* parameters and the algorithm took 1132 sec .On re-running, the final solutions were closed to each other. When the same example was executed on the *PC* with *NHGA*, in order to reach the solution N=100 and G=100 was sufficient, and the algorithm took 227 sec .Moreover, the obtained solutions on re-running indicated the high accuracy of the algorithm, such that the final solutions were exactly the same .The high speed of the proposed model and the quick convergence of the solutions make it desirable for large projects with a large number of activities . Considering uncertainty in each of the time, cost, and quality factors and even simultaneous uncertainty in more than one factor, this model can be extended to such cases, which makes it more realistic.

References

[1] A. Afshar, A. Kaveh and O.R. Shoghli, Multi-objective optimization of time-cost-quality using multi-colony ant algorithm, *Asian Journal of Civil Engineering*. 8(2) (2007), 113-124.

[2] A. Azaron, C. Perkgoz and M. Sakawa, A genetic algorithm approach for the time–cost trade-off in PERT networks, *Applied Mathematics and Computation* 168 (2005), 1317–1339.

[3] A. Azaron, R. Tavakoli, Multi-objective time–cost trade-off in dynamic PERT networks using an interactive approach , *European Journal of Operational Research* 180(3) (2007) 1186–1200

[4] A.J.G. Babu, N. Suresh, Project management with time, cost and quality considerations, *European Journal of Operational* 88 (1996), 320–327.

[5] M. Bagherpour, S. Noori and S. J. Sadjadi, Cost – Time trade-off models application to crashing flow shop scheduling problems, *ICCSA,* 3982 (2006), 546 – 553.

[6] X.M. Daisy, Applying parato ranking and niche formation to GA based multi objective time-cost optimization,. *J.Consr.Eng.Manage*., 131(1) (2005), 81-91

[7] P. De, E.J. Dunne, J.B. Ghosh, C.E. Wells, Complexity of the discrete time-cost trade-off problem for project networks, *Operations Research* 45 (1997), 302–306.

[8] E. Demeulemeester, W. Herroelen, S.E. Elmaghraby, Optimal procedures for the discrete time/cost trade-off problem in project networks, *European Journal of Operational Research* 88 (1996), 50–68.

[9] S.E. Elmaghraby, A. Salem, Optimal linear approximation in project compression. O.R. Technical Report 171, North Carolina State University at Raleigh (1981).

[10] K. El-Rayes, A. Kandil, Time-cost-quality trade-off analysis for highway construction, Journal of Construction. *Engineering Management*, 131(4) (2005), 477-485.

[11] E. Eshteharian, R. Abbasnia and A. Afshar, Optimization of uncertain construction time-cost trade-off problem,*The First International conference on Construction In Developing Countries,Pakistan* (2008)

[12] J.E. Falk, J.L. Horowitz, Critical path problems with concave cost–time curves, *Management Science* 19 (1972), 446–455.

[13] C.W. Feng, L. Liu and S.A. Burns, Using genetic algorithms to solve construction time-cost trade-off problems, *J. Comput. Civ. Eng.*, 11(3) (1997), 184-189

[14] D.R. Fulkerson, A network flow computation for project cost curves, *Management Science* 7(2) (1961),167–181.

[15] S.K. Goyal, A note on the paper: a simple CPM time/cost trade-off algorithm, *Management Science* 21 (1975),718–722.

[16] O. Hansancebi, F. Erbatur, Evaluation of crossover techniques in genetic algorithm based optimum structural design, *Computer and structures*, 78 (2000), 435-448

[17] T.J. Hindelang, J.F. Muth,A dynamic programming algorithm for decision CPM networks, *Operations Research* 27 (1979), 225–241

[18] H. Iranmanesh, M.R. Skandari and Allahverdiloo, Finding pareto optimal front for the multi_mode time, cost , quality trade-off in project scheduling,. *International Journal of Computer and Information and System Science, and*

Engineering. spring 2008
[19] C. G. Jin, Z. Ji *et al.*, Research on the fully fuzzy time-cost trade-off based on genetic algorithms, *Journal of Marine Science and Application*, 4(3) (2005), 18-23.
[20] K.C. Kapur, An algorithm for the project cost/ duration analysis problem with quadratic and convex cost functions, *IIE Transactions* 5 (1973), 314–322.
[21] J.E. Kelly, M.R. Walker, Critical path planning and scheduling: Mathematical basis, *Operation Research* 9(3) (1961),296-320
[22] D.B. Khang, Y.M. Myint, Time, cost and quality trade-off in project management: a case study, *International Journal of Project Management* 17 (4) (1999), 249–256.
[23] S. S, Leu, A. T. Chen and C. H. Yang, A GA-based fuzzy optimal model for construction time-cost trade-off, *International journal of Project Management,* 19 (2001),47-58.
[24] J.J. Moder, C.R. Phillips, E.W. Davis, Project Management with CPM, PERT and precedence diagramming, *third ed., Van Nostrand Reinhold Company*, 1983.
[25] J.H. Patterson, R.T. Harvey, An implicit enumeration algorithm for the time/cost trade-off problem in project network analysis, *Foundations of Control Engineering* 4 (1979), 107–117.
[26] D. Prabudha, E.J. Dunne, J.B. Ghosh, C.E. Wells, The discrete time-cost trade-off problem revisited, *European Journal of Operational* 81 (1995), 225–238.
[27] W. Prager, A structure method of computing project cost polygons, *Management science* 9,no3 (1963),394-404
[28] D.R. Robinson, A dynamic programming solution to the cost–time for CPM, *Management Science* 22 (1975), 158–166.
[29] N. Siemens, A simple CPM time/cost trade-off algorithm, *Management Science* 17 (1971),354–363.
[30] S.Tailiu, Fuzzy activity times in critical path and project crashing problems, *Cybernetics and Systems*, 34(2) (2003), 161 – 172.
[31] H.R. Tareghian, H. Taheri, An application of randomized minimum cut to the project time/cost trade-off problem, *Applied Mathematics and Computation* 173 (2005), 1200–1207.
[32] H.R. Tareghian, H. Taheri, A solution procedure for the discrete time, cost and quality trade-off problem using electromagnetic scatter search, *Applied Mathematics and Computation* 190 (2007), 1136–1145.
[33] H.R. Tareghian, H. Taheri, On the discrete time, cost and quality trade-off problem, *Applied Mathematics and Computation* 181 (2006), 1305–1312.
[34] D. X. Zheng, T. Ng and M. M. Kumaraswamy, Applying a genetic algorithm-based multiobjective approach for time-cost optimization, *Journal of Construction Engineering and Management*, 130(2) (2004), 168-176.

Received: November, 2009