Indian Journal of Science and Technology Vol. 4 No. 10 (Oct 2011) ISSN: 0974- 6846

Using genetic algorithms to solve industrial time-cost trade-off problems

Ghorbanali Mohammadi

Industrial Engineering Department, College of Engineering, Shahid Bahonar University of Kerman, Iran.
gmohammadi@mail.uk.ac.ir

Abstract

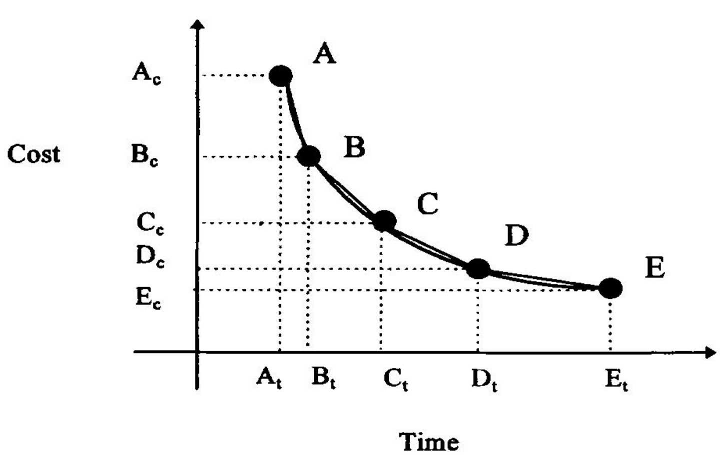
Time-cost trade-off analysis is one of the most important aspects of industrial project planning and control. There are trade-offs between time and cost to complete the activities of a project; in general, the less expensive the resources used, the longer it takes to complete an activity. Existing methods for time-cost trade-off problems focus on using heuristics or mathematical programming. These methods, however, are not efficient enough to solve large scale CPM problems. This paper presents a Multi-Objective Genetic Algorithm (MOGA) approach to time-cost trade-off problems (TCTP). Finding optimal decisions is difficult and time-consuming considering the numbers of permutations involved. This type of problem is NP-hard, hence attainment of IP/LP solutions, or solutions via Total Enumeration (TE) is computationally prohibitive. The MOGA approach searches for locally Pareto-optimal or locally non-dominated frontier where simultaneously optimization of time-cost is desired. The application of the proposed algorithm is demonstrated through an example project a real life case. The results illustrate the promising performance of the proposed algorithm.

Keywords: Time-cost trade-off, Genetic Algorithms, project management

Introduction

Industrial projects management face the decisions of selecting appropriate resources, including crew sizes, equipment, methods, and technologies to perform the activities of a project. In general there is a trade-off between the time and the cost to complete a task; the less expensive the resources, the large duration they take to complete an activity (Feng *et al.,* 1997). For example, using more productive equipment or hiring more workers may save time, but the cost could increase. Fig. 1 presents a typical relationship between time and cost to complete an activity. This figure indicates that the activity can be completed by either option A, B, C, D or E. Each option represents a different method of constructing the activity in which some of the resources are changed or a different technology is used. Ultimately, resource assignment decisions made at the activity level control the overall duration and cost of a project. If a project is running late the schedule plan, planners can perform a so called time -cost trade-off problems (TCTP) analysis. One method is to compress some of the activities on the critical path to save time cost (Siemens, 1971). The results of this analysis are: (1) a time-cost trade - off curve; and (2) the selection of different methods that provide the optimal balance of project duration and cost.

Fig 1. Typical relationship between time and cost of activity



With real-life projects involving hundreds of activities, finding optimal time-cost trade-off decisions is difficult and time consuming considering the number of permutations involved (Liu *et al.,* 1995).

Existing time-cost trade-off techniques

Since the late 1950s, critical path method (CPM) techniques have become widely recognized as valuable tools for the planning and scheduling of large projects. In a traditional CPM analysis, the major objective is to schedule a project assuming deterministic durations. However, project activities must be scheduled under available resources, such as crew sizes, equipment and materials. The activity duration can be looked upon as a function of resource availability. Moreover, different resource combinations have their own costs. Ultimately, the schedule needs to take account of the trade-off between project direct cost and project completion time. For example, using more productive equipment or hiring more workers may save time, but the project direct cost could increase.

In CPM networks, activity duration is viewed either as a function of cost or as a function of resources committed to it. The well-known time-cost trade-off problem (TCTP) in CPM networks takes the former view. In the TCTP, the objective is to determine the duration of each activity in order to achieve the minimum total direct and indirect costs of the project. Studies on TCTP have been done using various kinds of cost functions such as linear (Fulkerson, 1961; Kelly, 1961), discrete (Demeulemeester *et al.,* 1993), convex (Lamberson & Hocking, 1970; Berman *et al*., 2005), and concave Falk and Horowitz, (1972). Recently, some researchers have adopted computational optimization techniques such as genetic algorithms to solve TCTP. Chau *et al*. (1997) and Azaron *et al.* (2005) proposed models using genetic algorithms and the Pareto front approach to solve construction time-cost trade-off problems The existing techniques for time-cost trade-off problems (TCTP) can be categorized into two areas: mathematical programming and heuristic methods.

Mathematical programming models

These methods convert the TCTP to mathematical models and utilize linear programming, integer programming, or dynamic programming to solve them. Kelly 1961 formulated TCTP as a linear time -cost relationship within activities. Other researches, such as Hendrickson and Au 1989 and Pagnoni 1990, they used linear programming as a tool to solve the TCTP. These approaches is suitable for problems with linear time-cost relationship, but fail to solve those with discrete time-cost relationships. Burnes *et al*. (1996) took a hybrid approach, which used linear programming to find a lower bound of the trade-off curve and integer programming to find the exact solution for any desired duration. Elmagraby (1993) used dynamic programming to solve TCTP for networks that can be decomposed to pure series or parallel sub networks.

An alternative way to determine the total duration and find critical paths is by using LP technique (Hiller & Lieberman, 2001; Taha, 2003). The idea is based on the concept that a CPM problem can be thought of as the opposite of the shortest path problem, to determine a critical path in the project network it is sufficient to find the longest path from start to finish. Then the length of this longest path is the total duration time of the project network. The LP formulation assumes that a unit flow enters the project network at the start node and leaves at the finish node. Let x_{ij} be the decision variable denoting the amount of flow in activity (*i*, *j*). Since only one unit of flow could be in any arc at any one time, the variable x_{ij} must assume binary values (0 or 1) only.

Heuristic methods

These methods provide good solutions, but do not guarantee optimality. Example of heuristic approaches include Fondahl`s method (Fondahl, 1961). Siemens's effective cost slope model (Siemens, 1971) and Moselhi`s structural stiffness method (Moselhi, 1993). Most heuristic methods assume only linear time-cost relationships within activities.

Both Heuristic methods and mathematical models show the strengths and weaknesses in solving TCTP. The heuristic approaches select the activities to be shortened or expanded based on certain selection criteria, which do not guarantee optimal solutions. On the other hand, mathematical models require great computational effort and some approaches do not provide the optimal solution.

Meta-heuristic and evolutionary algorithms have shown relatively higher efficiency in received more attention. In recent works, Feng *et al*. (1997), Li *et al.* (1999) and Hegazy handling these problems. Although they do not necessarily guarantee the global optimal solutions, their ability to search the solutions space intelligently, rather than completely, makes them capable of producing relatively good solutions to large-sized problems. Among them algorithms, the genetic algorithms (GAs) and ant colony algorithm (ACO) have (1999) adopted GAs for Time- cost optimization problem. There are trade-offs between time and cost to complete the activities of a project; in general, the less expensive the resources used, the longer it takes to complete an activity. Using critical path method (CPM), the overall project cost can be reduced by using less expensive resources for noncritical activities without impacting the project duration. Existing methods for time-cost trade-off analysis focus on using heuristics or mathematical programming. This paper presents: (1) an algorithm based on the principles of GAs for Industrial time-cost trade-off optimization; and (2) a computer program that can execute the algorithm efficiently.

Notation

$D_{n(ij)}$ Normal activity time for node *i-j* ; $D_{f(i-j)}$ Crash activity time for *i-j;* K_n Total direct cost; d_{i-j} Pant time for activity *i-j;* n Total project node; t_i Plant time for activity *i*

Multi-objective optimization

A Multi-objective Optimization Problem (MOP) can be defined as determining a vector of design variables within a feasible region to minimize a vector of objective functions that usually conflict with each other. Such a problem takes the form:

$$\text{Minimize } \left\{f_1(X), f_2(X), \ldots, f_m(X)\right\} \tag{1}$$

$$\text{S.T.} \qquad g(X) \le 0,$$

Where X is vector of decision variable; $f_i(X)$ is the *i*th objective function; and $g(X)$ is constraint vector. A decision vector X is said to *dominate* a decision vector Y ($X \succ Y$) if:

$$f_i(X) \le f_i(Y) \quad \text{for all} \quad i \in \{1, 2, \ldots, m\} \tag{2}$$

and

$$f_i(X) \prec f_i(Y) \quad \text{for at least one} \quad i \in \{1, 2, \ldots, m\} \tag{3}$$

All decision vectors that are not dominated by any other decision vector are called *non-dominated* or *pareto-optimal*. These are solutions for which no objective can be improved without detracting from at least one other objective.

Indian Journal of Science and Technology Vol. 4 No. 10 (Oct 2011) ISSN: 0974- 6846

There are various solution approaches for solving the MOP. Among the most widely adopted techniques are: sequential optimization, *e*-constraint method, weighting method, goal programming, goal attainment, distance based method and direction based method. For a comprehensive study of these approaches, readers may refer to Szidarovsky *et al*. (1986).

Evolutionary Algorithms (EAs) seem particularly desirable to solve multi-objective optimization problems because they deal simultaneously with a set of possible solutions (the so-called population) which allows to find an entire set of Pareto-optimal solutions in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, EAs are less susceptible to the shape or continuity of the Pareto-optimal frontier, whereas these two issues are a real concern for mathematical programming techniques. However, EAs usually contain several parameters that need to be tuned for each particular application, which is in many cases highly time consuming. In addition, since the EAs are stochastic optimizers, different runs tend to produce different results. Therefore, multiple runs of the same algorithm on a given problem are needed to statistically describe their performance on that problem. These are the most challenging issues with using EAs for solving MOPs. The detailed discussion on application of EAs in multi-objective optimization has already been reported (Deb, 2001; Coello *et al.,* 2002).

The multi-objective TCTP

Objective functions

The following variables are defined for the muli-objective TCTP:

$$\mathrm{Min}\ Z = H\left(t_n - t_1\right) + \sum_i \sum_j C_{ij}\left(D_{n(ij)} - d_{ij}\right) + K_n$$

ST

$$t_j - t_i \geq d_{ij} \quad i, j = 1, 2, \ldots, n \tag{4}$$

$$D_{f(ij)} \leq d_{ij} \leq D_{n(ij)} \tag{5}$$

$$t_i \geq 0 \tag{6}$$

In constraint (4) actual activity time cannot be greater than differences between starting and ending of each activity time. In constraint (5) actual activity time is grater or equal than crash time and less than or equal normal time of each activity. In constraint (6) event activities time are always greater than or equal zero.

Pareto Front-Non-dominated Set

Solutions to a multi-*objective* optimization problem can be mathematically expressed in terms of non-dominated or superior points. If solution S1 is better than S2 in terms of all objective values, we say that the solution SI dominates S2 or the solution S2 is inferior to SI. Any member of the feasible region that is not dominated by any other member is said to be non-dominated or non-inferior. This non-dominated set is the so-called Pareto front. The members of the Pareto front are not dominated by any other members in the solution space; therefore, these solutions have the least objective conflicts of any other solutions, which provide the best alternatives for decision making.

Basically, we can treat TCTP as a multi-objective optimization process, which tries to minimize both project duration and cost. Each member in the population has its own total project duration and cost; therefore, a non-dominated set (a trade-off curve) can be determined such that there are no other members in the population that have better objective values in both time and cost than the members in the non-dominated set.

Genetic algorithms

Genetic search process

Initialization: The schedule generated is represented by chromosome *c*. The genetic search process starts with a randomly generated set of chromosomes called the initial population. The size of the population (*pop_size*) depends on the solution space.

Population evaluation: The fitness parameter (*fit* (*c*)) considered is VPC.

Selection of new population: The process of selecting the chromosomes to represent the next generation has the following steps (Ponnambalam *et al.,* 2003):

Step 1 Conversion of the fitness parameter value to a fitness value (*new_fit* (c)), a Parameter suitable for minimization objective.

New_fit (*c*) = 1- (*fit* (*c*)/F)

Where *fit* (*c*) =VPC corresponding to chromosome and F is the sum of the fitness values of all chromosomes, given by

$$F = \sum_{c=1}^{pop\_size} fit(c)$$

Step 2 Conversion of the new fitness parameter to an expected frequency of selection $(p(c))$, given by

$$p(c) = new\_fit(c) / \sum_{c=1}^{pop\_size} new\_fit(c)$$

Step 3 Calculation of the cumulative probability of survival $(cp(c))$

$$cp(c) = \sum_{c=1}^{pop\_size} p(c)$$

A random number between 0 and 1, *r* is obtained and a chromosome *c* is selected which satisfies the following condition:

$$cp(c-1) \leq r \leq cp(c)$$

This selection process is repeated as many times as the size of the population.

Crossover. Select a pair of chromosomes for crossover operation, if the random number generated is less than the probability of crossover.

Mutation. For the mutation, generate a uniform random number (*r*), if the uniform random number satisfies the following equation:

$$r \le p_m$$

Where p_m is the probability of mutation.

Termination: The above process will be repeated for the fixed number of generations. The crossover operator used in this algorithm is "order crossover (OX)" and the mutation operator is "inversion mutation (IV)" (Michalewicz, 1992).

Genetic operators

In this work Order Crossover (OX) and inversion (INV) operators (Michalewicz, 1992) are used. Order crossover is explained in Fig. 2. The intermediate string is obtained by reversing the second parent string at the second cut point. The respective positions of 1, 2, 3 and 4 are identified in intermediate string from the selected portion in first parent, and these are deleted from intermediate string. The elements remaining in intermediate string are copied into offspring first after the second cut point and next before the first cut point. The middle string is copied as it is from parent 1. By interchanging the first parent and second parent we can get the other offspring. The probability of crossover used in this paper is 0.8. INV is a unary operator. The INV operator first chooses two random cut points in a parent. The elements between the cut points are then reversed. An example of the inversion operation is presented in Fig.3. The substring that is cut and reversed is enclosed in the thick lined box and P and O denote parent and offspring, respectively. After conducting the analysis the number of generations is fixed as 1500, the probability of crossover as 0.8 and the probability of mutation as 0.2.

Fig.2. Order crossover (OX)

Parent 1: $\left(1\ 2\ 2 \middle| 2\ 1\ 3\ 3 \middle| \ 3\ 3\right)$

Parent 2: $\left(1\ 3\ 3 \middle| 1\ 2\ 3\ 2 \middle| \ 2\ 3\right)$

Intermediate string: $\left(2\ 3\ 1\ 3\ 3\ 1\ 2\ 3\ 2\ \right)$

Offspring: $\left(3\ 2\ 3 \middle| 2\ 1\ 3\ 3 \middle| \ 2\ 1\right)$

Fig 3. INV operator

$$\mathrm{P} = \left(1\ 2\ 2 \middle| \ 2\ 1\ 3\ 3\ \middle| \ 2\ 2\right)$$

$$\mathrm{O} = \left(1\ 2\ 2 \middle| \ 3\ 3\ 1\ 2\ \middle| \ 3\ 3\right)$$

Algorithm

The following genetic operations are employed to generate and handle a population in our Multi-objective genetic algorithm:

Step 0: (Initialization) randomly generate a parent population containing N_{pop} strings where N_{pop} is the number of strings in each population.

Step 1: (Reproduction) A second generation, O, the offspring population is created from *P* by selecting strings probabilistically relative to their *f* values with replacement.

Step 2: (Crossover) For each selected pair, apply a crossover operation to generate an offspring with the crossover probability P_c. N_{pop} strings should be generated from a pair of parent strings in the crossover operation.

Step 3: (Mutation) For each string generated by the crossover operation, apply a mutation operation with a pre-specified mutation probability P_m.

Step 4: (Elitist strategy) Randomly remove N_{elite} string from the N_{pop} strings generated by the above operations, and add the same number of strings from a tentative set of Pareto optimal solution to the current population.

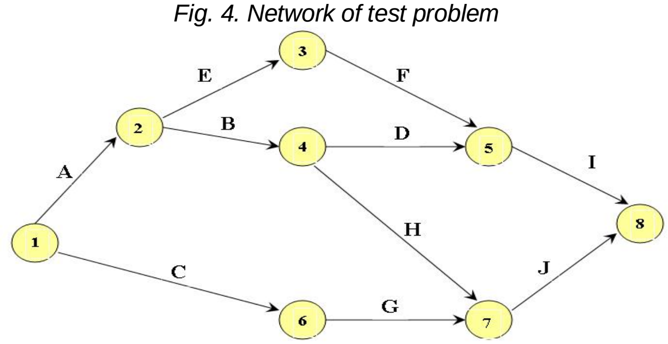
Step 5: (Termination test) If a pre-specified stopping condition is not satisfied, return to step1-4.

Step 6: (User selection) The multi-objective GA the final set of Pareto optimal solutions to the decision maker.

Test problem

Many test caused were generated to verify the accuracy of the algorithm. As an example, the verification of a 10 -activity CPM network is described in the following sections. Fig.4 shows the precedence relationships of the network and Table 1 shows the associated time cost for the options of each activity. Table 2 shows the result of test problem. Last column of the Table 2 indicates the priority of the methods used.

Fig. 4. Network of test problem



An initial generation of 300 strings is randomly selected and the initial generations is distributed over the solution space and does not gather in one region. The final generation occurred in the 60th iterations and the trade-off curve obtained from the final generation (see Fig 5).

After finding the trade-off curve, planners can determine the total cost and the direct cost from the total cost and the direct cost from the trade-off curve. Indirect cost is usually assumed to be proportional to the project duration.

Indian Journal of Science and Technology Vol. 4 No. 10 (Oct 2011) ISSN: 0974- 6846

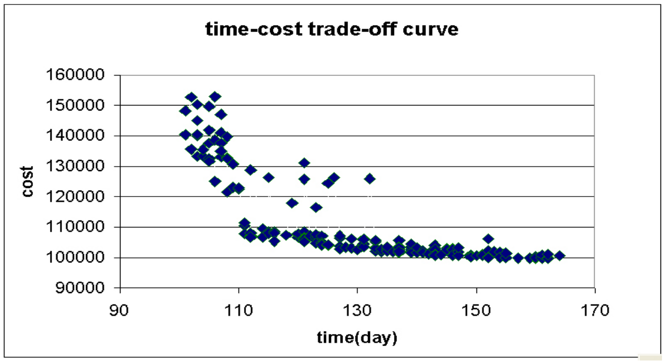
Table 1. Options of test problem

ID	Description	Priority		Possible Method	First Method		Second Method		Third Method		Fourth Method		Fifth Method	
		P	S	M	$1	D1	$2	D2	$3	D3	$4	D4	$5	D5
1	A	1	2	5	5500	12	2200	15	1900	18	2000	21	200	23
2	B	2	4	5	2500	10	2400	18	1800	20	1500	23	1000	22
3	C	1	6	3	8700	15	4000	22	3200	33				
4	D	4	5	3	45000	12	45000	16	30000	20				
5	E	2	3	4	30000	15	17500	24	15000	28	10000	45		
6	F	3	5	3	45000	16	32000	18	18000	24				
7	G	6	7	3	23000	8	24000	15	23000	18				
8	H	4	7	5	200	14	215	15	200	16	300	21	120	24
9	I	5	8	5	300	15	240	18	180	20	150	23	100	30
10	J	7	8	3	450	15	450	21	320	33				

Table 2. Test problem result (printout)

Individual	Duration	Cost	Tardiness	Resource	Fitness	Method	Priority
1	149	100790	1	178235	6.33E+01	5533433322	3863193987
2	124	116208	1	115088	6.33E+01	4352213542	3465897865
3	113	136020	1	152249	6.33E+01	1254324532	1326549781
4	144	113930	1	101226	6.33E+01	4423325421	9856231455
5	159	99940	1	173264	6.33E+01	1125543212	2319786523
6	162	99740	1	165338	6.33E+01	5532452132	1325649785
7	124	107360	1	167893	6.33E+01	4552325412	9856437853
8	133	108558	1	108934	6.33E+01	5423254123	1326594787
9	120	116808	1	129676	6.33E+01	3254232541	2365897854
10	114	119658	1	170653	6.33E+01	3325425354	9856123546
11	118	116908	1	167129	6.33E+01	4452325412	9995874412
12	107	149820	1	373881	6.33E+01	5523554125	3863193998
13	113	121408	1	466565	6.33E+01	2135423521	3467853264
14	102	159820	1	455782	6.33E+01	1125432541	2356998454
15	118	120158	1	165452	6.33E+01	5523254123	2569837461
16	115	118908	1	169147	6.33E+01	2235423542	2561113546
17	118	114010	1	182294	6.33E+01	4253254125	1119986542
18	101	163808	1	179902	6.33E+01	3252354215	1452368975
19	146	100920	1	194815	6.33E+01	3325423254	2563589654
20	107	140670	1	658510	6.33E+01	1452325542	3569874152
21	131	116208	1	101010	6.33E+01	5523254123	4125558754
22	125	116208	1	113159	6.33E+01	5425325412	4445255632
23	123	115290	1	130670	6.33E+01	1232545232	3336589655
24	164	100790	1	138633	6.33E+01	2354232145	2546354455
25	155	102090	1	123578	6.33E+01	2532541235	2223655475
26	127	106460	1	182590	6.33E+01	4521325421	7854632587
27	154	100440	1	218436	6.33E+01	5533433322	6665522477
28	155	100790	1	142819	6.33E+01	5523542514	2221456985
29	141	101490	1	284586	6.33E+01	4521354254	1112544856
30	140	101590	1	230715	6.33E+01	4251325412	2365899785

Fig.5. Optimal trade-off curve of test problem



Optimal choice to perform the project would be the lowest total cost. Using trade-off curve as the objective function allows for much more efficient evaluation of various indirect cost rates without performing another GA run. This is an improvement over treating the total cost as the objective in the GA.

Conclusions

Industrial time-cost trade-off problems (TCTP) are large scale optimization problems. The existing techniques using heuristic and mathematical programming are not efficient or accurate enough to solve TCTP of real-life Industrial projects. The present study develops a GA Pareto front approach to solve the CPM time-cost trade-off problem in most industrial decisions. The proposed algorithm is easy to implement and capable to treating any type of TCTP.

References

1. Azaron A Perkgoz C and Sakawa M (2005) A genetic algorithm approach for the time-cost trade-off in PERT networks, *Appl. Maths Comput.* 168, 1317-1339.
2. Burns S Liu L and Feng C (1996) The LP/IP hybrid method for construction time-cost trade-off analysis. *Constr. Mgmt. Econ.* 14, 265-276.
3. Coello CAC Van Veldhuizen DA and Lamont GB (2002) Evolutionary algorithms for solving multi-objective Problems. Kluwer academic publishers, NY.
4. Chau DKH, Chan W and Govindan K (1997) A time-cost trade-off model with resource consideration using genetic algorithm. *Civil Engg. Sys*. 14, 291-311.
5. Deb K (2001) Multi-Objective Optimization using Evolutionary Algorithms. John Wiley, Chichester, UK. Demeulemeester E Herroelen W, Elmaghraby S (1993) Optimal procedures for the discrete time-costtrade-off problem in project networks, Research Report, Department of Applied Economics, Katholieke Universiteit Leuven, Leuven, Belgium.
6. Elmograby SE (1993) Resource allocation via dynamic programming in activity networks. *Euro. J. Operational Res*. 64, 199-215.
7. Feng CW Liu LY and Burns SA (1997) Using genetic Algorithms to Solve Construction Time-cost trade-off Problems. *J. Compt. Civil Engg*. 11, 184-189.
8. Fulkerson D (1961) A network flow computation for project cost curves, *Mangt. Sci.* 7, 167-178.
9. Fondall JW (1961) A non-computer approach to the critical path method for the construction Industry. The Rep. No. 9, the Constr. Inst. Dept. of Civil Engg; Stanford Univ., Stanford California.
10. Falk J and Horowitz J (1972) Critical path problem with concave cost curves. *Mangt. Sci.* 19, 446-455.
11. Hendrickson C and Au T (1989) Project management for construction. Fundamental concept for owners, engineers, architect, and builders. Englewood Cliffs (NJ) prentice-Hall Inc.
12. Hegazy T (1999) Optimization of time-cost trade-off analysis: using genetic algorithms*. Can. J. Civil Engg*. 26, 685-697.
13. Hillier FS and Lieberman GJ (2001) Introduction to Operations Research. 7th ed, McGraw-Hill, Singapore.
14. Kelly J (1961) Critical path planning and scheduling: mathematical basis. *Operations Res*. 9, 296-320.
15. Liu LY Burns SA and Feng CW (1995) Construction time-cost trade-off analysis using LP/IP hybrid method. *J. Constr. Engg*. 12, 446-54.
16. Lamberson L and Hocking R (1970) Optimum time compression in project scheduling. *Mangt. Sci*. 16, 597-606.
17. Li H, Cao JN and Love PE (1999) Using machine learning and GA to solve time-cost trade-off problems.*J. of Construct. Engg. and Mangt*. 125, 347-353.
18. Michalewicz Z (1992) Genetic algorithms + data structure = evolution programs. Berlin: springer.
19. Moselhi O (1993) Schedule compression using the direct stiffness method. *Can. J. Civil Engg.* 20, 65-72.
20. Pagnoni A (1990) Project engineering: Computer oriented planning and operational decision making, Spring-Verlag. KG. Berlin.
21. Ponnambalam SG Aravindan P and Subba Rao M (2003) Genetic algorithms for sequencing problem in mixed model assembly lines. *Computers & industrial Engg.* 45, 669-690.
22. Siemens N (1971) A simple CPM time-cost trade-off Algorithms. *Mangt. Sci*. 17B, 354-363.
23. Szidarovsky F Gershon ME and Dukstein L (1986) Techniques for multi objective decision making in systems Management. Elsevier, NY.
24. Taha HA (2003) Operations research: an introduction. 7th ed., Prentice-Hall, NJ.