KSCE Journal of Civil Engineering (2017) 21(4):1047-1058

DOI 10.1007/s12205-016-1615-x

Construction Management

pISSN 1226-7988, eISSN 1976-3808
www.springer.com/12205

Network Analysis Algorithm for the Solution of Discrete Time-Cost Trade-off Problem

Önder Halis Bettemir* and M. Talat Birgönül**

Received August 10, 2015/Revised February 7, 2016/Accepted April 28, 2016/Published Online July 29, 2016

Abstract

Optimum solution of time-cost trade-off problem has significant importance since it provides the highest profit opportunity. For this reason, exact, heuristic, and meta-heuristic algorithms are adapted to obtain the optimum or near-optimum solution. However, heuristic algorithms may not always converge into the global optimum, while meta-heuristic algorithms require significant computation to converge into global optimum and exact methods are complex for construction planners to implement. Therefore, minimum cost-slope based fast converging network analysis algorithm, which provides optimum or near-optimum solutions, is proposed for discrete time-cost trade-off problem. The algorithm searches the global optimum through the feasible crashing options. Number of feasible crashing options increase tremendously in large projects. Therefore, an elimination algorithm is embedded to reduce the number of crashing options. The crashing option with the lowest unit crashing cost is executed and global optimum is searched by stepwise crashing. Tests on 18 and 63-Activity projects revealed that the network analysis algorithm converges to optimum or near-optimum solution by only one percent of the computational demand of meta-heuristic algorithms. Consequently, the proposed heuristic algorithm is a convenient optimization method for the solution of time-cost trade-off problem.

Keywords: *time-cost trade-off, optimization, project planning, network analysis algorithm*

1. Introduction

Competition in the construction industry is increasing day by day as new firms are entering into market and the existing companies are enlarging their job opportunities. In order to gain competitive advantage against rivals, the construction companies aim to minimize the costs. However, this goal requires excellent planning and scheduling of construction projects.

Project duration can be shortened by expediting critical activities with additional expense. Crashing a critical activity increases the construction cost of the activity, while decreases project duration. Decrease in project duration reduces the indirect cost. Minimum sum of direct and indirect project costs are searched in Time-Cost Trade-off (TCT) problems (Hegazy, 1999a).

Importance of the TCT problem was recognized more than half a century ago, almost simultaneously with the development of project analysis techniques by Fulkerson (1961), Kelly and Walker (1959) and Kelly (1961). Several heuristic algorithms, which aim to achieve optimum solution of TCT problems, were developed. First considerable attempt to solve TCT problem can be noted as the heuristic algorithm derived by Siemens (1971) and later his algorithm was improved by Goyal (1975, 1996), and Siemens and Gooding (1975). The algorithms provide minimum total project cost for a specific project completion duration. Later, Barber and Boardman (1988) and Chiu and Chiu (2005) proposed heuristic algorithms for the solution of TCT with linear cost curves. The aforementioned algorithms provide optimum solution for continuous crashing functions but cannot guarantee convergence into global optimum for the non-linear or discrete crashing alternatives.

Berman (1964) developed an algorithm for the optimization of the networks that have concave continuous cost functions. Falk and Horowitz (1972) examined the concave continuous cost functions by branch-and-bound (BaB) algorithm. In addition to this, Vanhoucke (2005) proposed BaB algorithm for the TCT problem. Network decomposition algorithms were also presented to solve TCT (Schwarze, 1980; De *et al*., 1995; Demeulemeester *et al*., 1996; Demeulemeester *et al.*, 1998; Vanhoucke and Debels, 2007; Hazır *et al*., 2010a).

Neural Network was adapted by Pathak and Srivastava (2014) for the solution of TCT. Integer programming-linear programming (IP/LP) and mixed integer programming provide lower bounds of the time-cost relationships. Thus, IP/LP and mixed-integer programming algorithms are adapted by Liu *et al.* (1995), Moussourakis and Haksever (2004), Sakellaropoulos and Chassiakos (2004), Liberatore and Pollack-Johnson (2006), Bidhandi (2006),

*Assistant Professor, Dept. of Civil Engineering, Inonu University, Malatya 44280, Turkey; Previously, Dept. of Civil Environmenta Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, USA (Corresponding Author, E-mail: onder.bettemir@inonu.edu.tr)
**Professor, Middle East Technical University, Dept. of Civil Engineering, Ankara, Turkey (E-mail: birgonul@metu.edu.tr)

Moussourakis and Haksever (2007), Hazır *et al.* (2010b), Son *et al.* (2013), and Al Haj and El-Sayegh (2015).

TCT is considered as NP-Hard (De *et al.*, 1997); i.e., search domain increases significantly faster than the project size. Thus, memory requirement and computational demand of the exact algorithms increase significantly. Therefore, meta-heuristic algorithms are implemented for the solution of the TCT problem even though the presence of exact algorithms. GA is the most preferred meta-heuristic algorithm for the solution of TCT (Feng *et al*., 1997; Li *et al*., 1999; Hegazy, 1999a; Zheng *et al.*, 2004 and 2005a; Bettemir, 2009; Lee *et al*., 2010). Hybrid Genetic Algorithm and Shuffled Frog Leaping successfully solve small projects in reasonable computation time (Elbeltagi *et al*., 2005; Sonmez and Bettemir, 2012). However, optimum or near optimum solution requires approximately half a million schedule evaluations for large projects (Bettemir, 2009). Moreover, Ng and Zhang (2008), and Zhang and Thomas Ng (2012) implemented Ant Colony Optimization, Yang (2007) employed Particle Swarm Optimization, Anagnostopoulos and Kotsikas (2010) used Simulated Annealing, Rogalska *et al.* (2008) utilized hybrid evolutionary algorithm, and Geem (2010) implemented Harmony Search for TCT problem. Cha and Lee (2015) analyzed TCT by Building Information System software.

Trade-off during the project planning is not limited to time and cost. Therefore, variants of time-cost trade-off problem are also analyzed. Babu and Suresh (1996), Khang and Myint (1999), Tareghian and Taheri (2006), Zhang and Xing (2010), Kim *et al.* (2012), Mungle *et al.* (2013), Tavana *et al.* (2014), and Monghasemi *et al.* (2015) added quality to TCT problem and solved the time-cost-quality trade-off.

Cost and duration options are assumed deterministic in TCT. On the contrary, both activity costs and durations have uncertainty and Leu *et al.* (2001), Zheng and Ng (2005), Azaron *et al.* (2005), Yang (2005), Eshtehardian *et al.* (2009), Li and Wang (2009), Ke *et al.* (2009), Kalhor *et al.* (2011), Chen and Tsai (2011), Yang (2011), Xu *et al.* (2012), and Said and Haouari (2015) considered the uncertainties of the crashing options.

TCT problem is solved by assuming infinite resource availability. If resources are available in limited quantities, then the problem is named multi-mode resource constrained project scheduling. In order to take availability of resources into account, Hegazy (1999b), Liu and Wang (2008), Ghoddousi *et al.* (2013), Afruzi *et al.* (2014), Rostami *et al.* (2014) and Cheng and Tran (2016) solved TCT with limited resource.

Elazouni and Metwally (2007) solved TCT by minimizing the negative cash flow and similarly, Ammar (2011) solved TCT to maximize the net present value. In addition to this, Fathi and Afshar (2010) maximize the net present value of the profit by GA. Koo *et al.* (2015) introduced environmental impact and solved Time-Cost-Environment trade-off.

To sum up, heuristic algorithms do not always converge into exact solution, meta-heuristic algorithms have significant computational demand, and exact algorithms are complex for construction planners. As a result, a robust solution algorithm for the solution of discrete TCT problem still lacks. Minimum cost-slope method provides optimum solution for continuous TCT problem. However, the algorithm is not suitable for the discrete TCT problem because, discrete crashing options prevent formation of linear cost functions. In this study; fast, simple, and optimum converging network analysis algorithm, inspired by minimum cost-slope method, is proposed for the solution of discrete TCT problem.

2. Methodology

Network Analysis Algorithm (NAA) searches the optimum schedule by crashing the activities step-by-step. Selection of the crashing options is not based on the maximum benefit because the maximum benefit may not be the best opportunity when the penalty conditions are disappeared. On the contrary, reward becomes permanent payment when the project is completed earlier than the deadline. In this case, crashing cost is compared with the sum of the indirect cost and the reward.

Flowchart of the network analysis algorithm is represented in Fig. 1. The algorithm initially identifies the paths of the network. Then critical path or paths are determined by backward pass. If there is one critical path, the critical activity with the least unit crashing cost is selected as candidate among the activities on the critical path. If there are multiple critical paths, crashing options that expedite every critical path are determined. The crashing option with the least unit crashing cost is selected as candidate. If

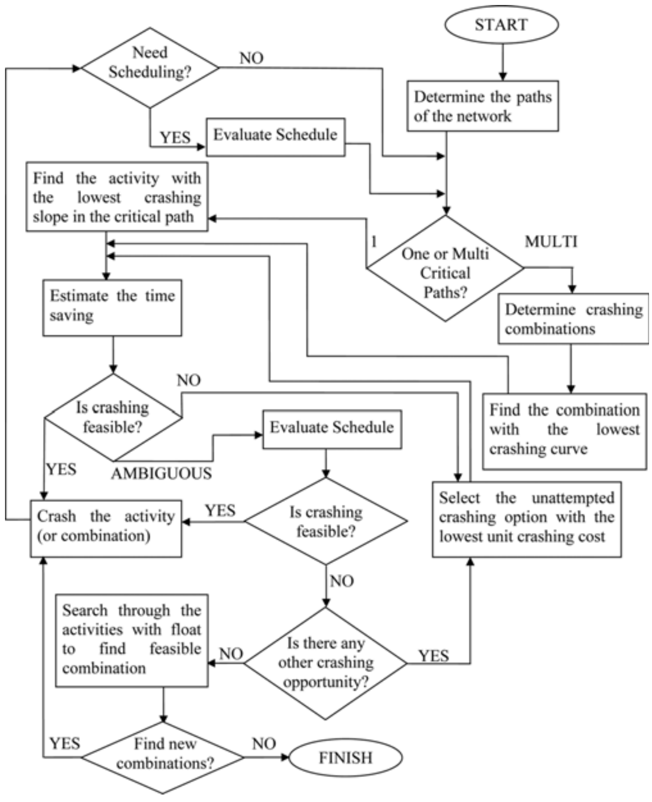

Fig. 1. Flowchart of the Network Analysis Algorithm

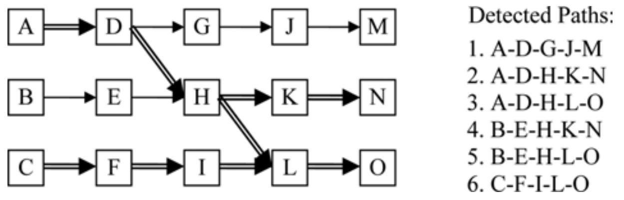

Fig. 2. Illustration of Path Detection Algorithm on an Example Project

the candidate crashing option is feasible, the crashing option is executed, else an unattempted crashing option with the lowest unit crashing cost becomes new candidate option. Iteration is terminated if there is not any feasible crashing option. The network analysis consists of path detection, determination of crashing options, and assessment of feasibility modules, which are explained in detail in the following sub-sections.

2.1 Path Detection

Path detection is executed in the beginning and only once. If an activity, x, has n ($n > 1$) successors, the path including the activity x is duplicated n - 1 times and the successor activities are chained to the duplicated paths one-by-one. A path is completed if an activity with no successor is concatenated. A sample path determination is illustrated in Fig. 2.

The path detection algorithm starts with the activities, which has no predecessors and adds the successor activities in sequence. The algorithm starts with activity A then adds activity D and obtains the path A-D. The path A-D is duplicated once since the activity D has two successors, G and H respectively. The successors are added to immature path A-D and paths A-D-G and A-D-H are obtained. Path 1, A-D-G-J-M, is obtained by adding the activities J and M to A-D-G. Activity H has two successors, K and L. For this reason, immature path A-D-H is duplicated once and K and L activities are added which forms the immature paths A-D-H-K and A-D-H-L respectively. These paths are completed by adding the activities N and O respectively. Remaining paths are detected by applying the same procedure for the activities B and C. Path detection algorithm identifies the six paths in the network by systematically concatenating the successor activities to their predecessors. The detected paths are shown in Fig. 2.

Critical path or paths are identified among the detected paths. Critical path is defined as the path, which consists of only critical activities. Therefore, paths containing both critical and uncritical activities are not classified as critical path.

2.2 Determination of Crashing Combinations

Search for the feasible crashing options is conducted in three situations. The first one is the existence of one critical path, the second one is the presence of multi-critical paths and the last one is the necessity of crashing uncritical activities together with the critical ones.

If there is on critical path, activities on the critical path are sorted according to their unit crashing costs. The activity with the lowest unit crashing cost is assigned as candidate and feasibility of it is examined. If the crashing option is feasible it is executed, otherwise it is eliminated and next crashing option is examined. If all options are examined without obtaining a feasible option, new crashing options are formed by combining critical and uncritical activities.

If number of critical paths is more than one, feasible crashing options are determined among the combination of critical activities on the critical paths. Number of possible crashing options for the n-critical path network is equal to the number of n^{th} combinations of the critical activities. If the crashing options were not eliminated, there would be $N_1 \times N_2 \times ... \times N_n$ crashing options where N_i is the number of activities on the i^{th} critical path. Number of crashing options grows significantly fast and causes memory allocation problems. Therefore, an elimination method, which searches for critical activities that exist on every critical path is implemented. If such a critical activity is detected, it is recorded as a feasible crashing option and erased from the critical paths. After the removal of common critical activities, critical paths are clustered according to their common critical activities. Common critical activities are again recorded separately and removed from the combination list. A search algorithm for the detection of analogous critical paths is not needed since the path detection algorithm, represented in Fig. 2, lists the similar paths in sequence.

To illustrate the combinatorial algorithm, paths 2, 3 and 6 of the network given in Fig. 2 are assumed critical paths. There is not a common critical activity between the three paths, so no activity is eliminated. Paths 2 and 3 are matched and activities A, D and H are separated from the paths A-D-H-K-N and A-D-H-L-O. The remaining activities K-N and L-O are combinatorial matched and K_L, K_O, N_L and N_O matches are obtained. Common critical paths, A, D, and H are added and seven crashing options are obtained for the critical paths 2 and 3. Combinations of the seven crashing options and the critical path C-F-I-L-O are matched. Path C-F-I-L-O has the common activities of L and O with the 4 of the 7 crashing options. Activity L exists in K_L and N_L options while activity O exists in K_O and N_O options. Combination of C-F-I with K_L, K_O, N_L and N_O will not provide a superior crashing option, since the aforementioned 4 combinations already have common critical activities. Combination of A-D-H and C-F-I-L-O produce 15 crashing options. These are added to the 4 existing crashing options and finally 19 crashing options are obtained compared with 125 crashing options if no elimination is performed.

Obtaining a feasible crashing option may require crashing the same activity more than once or crashing more than one activity in the same critical path. If these possibilities are examined, the search domain will increase significantly for large and complex networks. Throughout the determination of the crashing options, each option has to be kept in memory. In order to keep memory requirements within reasonable limits, the search algorithm skips crashing the alternatives multiple times with the increased risk of

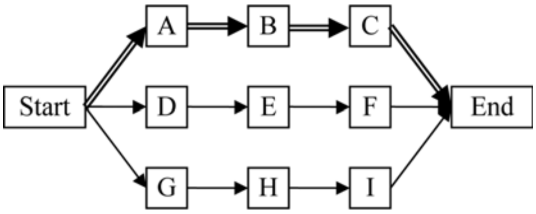

Fig. 3. Illustration of Crashing Critical and Uncritical Activities Simultaneously

missing the global optimum.

The third case is the generation of crashing options from both critical and uncritical activities. If crashing options are rejected because of the inadequate total floats of the uncritical activities, the search algorithm tries to produce feasible crashing options by attaching uncritical activities. If feasibility of the option is vague, the uncertainty is clarified by schedule evaluations.

The third case is illustrated in Fig. 3 with a hypothetical project. Path, A-B-C is the critical path, and paths D-E-F and G-H-I have total float durations m and n days respectively where $m < n$. If all of the crashing alternatives of the path A-B-C expedites more than n days with a cost more than the indirect cost of m-days, crashing only A, B, or C becomes infeasible. In order to obtain feasible crashing options, paths D-E-F and G-H-I should be included in the search domain. Feasible crashing options can be obtained by only including path D-E-F, or D-E-F and G-H-I. Furthermore, the feasible option may require crashing an activity more than one at a time or crashing more than one activity on the same path. Consequently, discrete crashing options significantly expand the search domain. Similar to the second case, examining such a large domain requires tremendous memory even for a medium size project. For this reason, uncritical activities are examined only one-by-one without considering the second or further combinations with the risk of missing the best crashing option.

Some of the cost curves of discrete crashing alternatives may not be monotonic decreasing. In this case, crashing the activity by only one crashing mode may not be feasible while, expediting by two or more crashing modes at the same time can be feasible. If NAA cannot find any feasible crashing option, the further crashing modes of the critical activity are examined. If a decrease in the crashing cost slope is detected at the next crashing mode, feasibility of crashing the critical activity twice is examined.

2.3 Assessment of Feasibility

Search algorithm determines significantly many crashing alternatives, which requires being evaluated with a considerable computational demand. However, the proposed method is aimed to be fast converging. Consequently, the crashing options need to be assessed without a schedule evaluation. A crashing option does not provide the expected shortening in the project duration if an uncritical path with inadequate total float becomes critical path. On the other hand, the expedited activity may also shorten the uncritical paths and the expected shortening in the project duration might be achieved. To explain this statement, activities A and C of the network given in Fig. 2, are assumed to be expedited. Expediting activity A may not shorten the duration of the path A-D-H-L-O as it is shortened if the total floats of the activities B and E are less than the expedition of activity A. In this case, activity H waits for the completion of activity E, instead of D and the path B-E-H-L-O becomes the new critical path. As a result, shortening in the project duration is limited by the total float of path B-E. Conversely, if total float of the path B-E is adequate, project duration will not be affected from the path B-E-H-L-O.

This dilemma can be solved by executing a trial schedule evaluation. In order to keep the number of schedule evaluations in a reasonable limit, a project duration predictor is embedded. The algorithm predicts the expedition of each path and the overall network by only considering the expedited activities in the paths and the activities with minimum float amounts.

Shortening in the project duration is predicted by examining each path of the network. Paths, which contain a crashed activity, are certain to be shortened. However, the remaining paths may or may not be shortened. Situation of them are predicted by considering total floats of the activities. Minimum of the total float values of the activities on an uncritical path, which is greater than zero, is assigned as the *critical total float* (CTF) of the corresponding uncritical path. This float value is assigned as the maximum safe crashing duration, which assures the corresponding path will not be a critical path when a crashing within CTF duration is executed. Amount of shortening in project duration is estimated conservatively by the expression $min(M_i^1 - M_{i+1}^1, ..., M_j^k - M_{j+1}^k, ..., M_l^m - M_{l+1}^m, CTF_1, CTF_2, ..., CTF_n)$. In the expression M_i^k represents the duration of k^{th} activity for its i^{th} crashing option, m is the number of crashed activities, CTF_j is the critical total float of the j^{th} path and n is the number of uncritical paths. The estimated shortening duration is conservative since crashing a critical activity may also expedite the paths, in which the crashed activity does not exist. However, determination of exact shortening of the paths requires extensive analysis of the network. In order to compare the schedule evaluation numbers fairly, networks are not analyzed in detail and computational demand for one crashing step is kept reasonable. If $min(M_i^1 - M_{i+1}^1, ..., M_j^k - M_{j+1}^k, ..., M_l^m - M_{l+1}^m) \geq$ *minimum expedition* $> min(M_i^1 - M_{i+1}^1, ..., M_j^k - M_{j+1}^k, ..., M_l^m - M_{l+1}^m, CTF_1, CTF_2, ..., CTF_n)$ then a schedule evaluation is performed with only forward pass and the conflict is resolved.

3. Case Studies

NAA is tested on 6 small-size projects ranging from 7 to 10 activities and it converged into global optimum in all of the projects in 4 to 8 crashing attempts. The algorithm is also tested on the two case problems of 18-Activity project (Hegazy, 1999a) and two case problems of 63-Activity project (Bettemir, 2009). First case problem of 18-Activity project is daily $200 indirect cost without reward or penalty, and the latter case problem is $200 indirect cost with $20,000 penalty/day for later or $1000

Table 1. Precedence Relations and Crashing Options of 18-Activity Project

Act ID	Pred.	Normal		Crash 1		Crash 2		Crash 3		Crash 4	
		Dur.	Cost	Expd.	A.C.	Expd.	A.C.	Expd.	A.C.	Expd.	A.C.
1	–	24	1200	3	300	5	400	1	250	1	250
2	–	25	1000	2	500	3	300	2	600	3	600
3	–	33	3200	11	800	7	500	–	–	–	–
4	–	20	30000	4	5000	4	10000	–	–	–	–
5	1	30	10000	2	5000	4	2500	2	2500	–	–
6	1	24	18000	6	14000	4	8000	–	–	–	–
7	5	18	22000	3	2000	6	6000	–	–	–	–
8	6	24	120	3	88	5	-8	1	15	1	5
9	6	25	100	2	50	3	30	2	60	3	60
10	2, 6	33	320	11	80	7	50	–	–	–	–
11	7, 8	20	300	4	50	4	100	–	–	–	–
12	5, 9,10	30	1000	2	500	4	250	2	250	–	–
13	3	24	1800	6	1400	4	800	–	–	–	–
14	4, 10	18	2200	3	200	6	600	–	–	–	–
15	12	16	3500	4	1000	–	–	–	–	–	–
16	13, 14	30	1000	2	500	4	250	2	250	2	1000
17	11, 14, 15	24	1800	6	1400	4	800	–	–	–	–
18	16, 17	18	2200	3	200	6	600	–	–	–	–

Table 2. Precedence Relations and Crashing Options of 63-Activity Project

Activity ID	Pred	Normal		Crash 1		Crash 2		Crash 3		Crash 4	
		Expd.	C.C.	Expd	C.C.	Expd	C.C.	Expd	C.C.	Expd	C.C.
1	NULL	14	3750	2	500	2	1150	1	850	-	-
2	NULL	21	11250	3	3550	1	1400	2	3450	-	-
3	NULL	24	22450	2	2450	3	3050	2	3700	-	-
4	NULL	19	17800	2	1600	2	2200	-	-	-	-
5	NULL	28	31180	2	3020	3	4050	2	3150	-	-
6	1	44	54260	2	4190	4	4775	3	4925	-	-
7	1	39	47600	3	3150	3	4050	3	4950	-	-
8	2	52	62140	5	7560	3	2900	5	9150	-	-
9	3	63	72750	4	6700	4	6800	4	5250	2	8000
10	4	57	66500	4	3750	3	5550	4	4950	5	5700
11	5	63	83100	4	6350	4	8350	5	6450	5	8150
12	6	68	75500	6	6500	4	5500	5	4300	4	4750
13	7	40	34250	3	4250	4	5450	2	4800	-	-
14	1, 8	33	52750	3	5700	3	4950	2	2850	-	-
15	9	47	38140	7	3360	5	6150	3	6450	-	-
16	9, 10	75	94600	5	6650	4	11500	5	11750	4	8350
17	10	60	78450	5	6050	6	6750	2	3390	-	-
18	10, 11	81	127150	8	16100	7	11350	5	7300	-	-
19	11	36	82500	2	12300	4	6900	-	-	-	-
20	12	41	48350	4	4900	3	6200	2	7350	-	-
21	13	64	85250	4	7350	3	7200	4	7700	4	6250
22	14	58	74250	5	4850	3	7600	3	4800	5	5900
23	15	43	66450	2	3350	4	6000	4	5600	3	7050
24	16	66	72500	4	6000	4	5200	5	5650	4	7050
25	17	54	66650	4	3450	3	4700	4	4700	3	7300
26	18	84	93500	5	9000	6	8750	5	8500	6	8750
27	20	67	78500	7	7950	3	2650	1	2400	3	3250
28	21	66	85000	3	4750	3	2750	2	4300	4	3700
29	22	76	92700	5	5800	4	6100	3	5300	4	5700
30	23	34	27500	2	2300	3	1950	2	2050	1	2400

Table 2. (continued)

Activity ID	Pred	Normal		Crash 1		Crash 2		Crash 3		Crash 4	
		Expd.	C.C.	Expd	C.C.	Expd	C.C.	Expd	C.C.	Expd	C.C.
31	19, 25	96	145000	7	9800	6	13850	6	10850	5	9600
32	26	43	43150	3	5150	3	3150	2	3150	2	6850
33	26	52	61250	3	3100	5	4400	3	5750	3	5000
34	28, 30	74	89250	3	4550	5	5950	4	5350	5	9150
35	24,27,29	138	183000	12	18500	11	36500	12	45750	5	13750
36	24	54	47500	5	3250	7	6050	4	5950	5	5500
37	31	34	22500	2	1600	3	2650	2	3050	3	1800
38	32	51	61250	4	4550	3	5450	3	5250	3	3900
39	33	67	81150	6	6450	4	4500	5	5350	3	5350
40	34	41	45250	2	3150	3	2800	3	3500	2	3500
41	35	37	17500	6	3700	4	5650	4	5450	-	-
42	36	44	36400	3	3350	3	3050	6	5500	2	1950
43	36	75	66800	6	4400	6	5200	4	4900	5	4900
44	37	82	102750	6	6750	6	17500	4	9800	3	9200
45	39	59	84750	4	6650	4	9900	4	25200	4	16250
46	39	66	94250	3	5250	4	8750	4	10250	5	17500
47	40	54	73500	3	5000	4	5100	3	5100	3	4700
48	42	41	36750	2	3050	2	4000	3	4700	3	5450
49	38,41,44	173	267500	14	22200	12	22300	9	40500	17	45250
50	45	101	47800	27	13500	11	15500	14	14700	-	-
51	46	83	84600	6	9050	5	4850	7	6100	4	8600
52	47	31	23150	3	4450	2	2200	2	2950	3	2450
53	43, 48	39	31500	3	2750	3	3550	4	3450	3	3350
54	49	23	16500	1	1300	1	1950	1	1450	2	3100
55	52, 53	29	23400	2	1850	1	1650	2	2500	2	3100
56	50, 53	38	41250	3	3400	2	3150	2	3600	2	4050
57	51, 54	41	37800	3	3450	3	4350	3	4150	2	3650
58	52	24	12500	2	1100	2	1650	2	1550	2	2650
59	55	27	34600	3	2900	2	3750	3	5500	2	4000
60	56	31	28500	2	2000	2	2750	2	4750	4	5800
61	56, 57	29	22500	2	2250	2	2500	3	2550	2	3700
62	60	25	38750	2	2450	2	3550	2	5050	2	1300
63	61	27	9500	1	200	1	400	1	700	2	1900

reward/day for earlier finishes than 110 days. Case problems of 63-Activity project are daily $2300 and $3500 indirect costs without any penalty or reward respectively. Precedence relationships and crashing alternatives of the activities are given in Table 1 and Table 2 for the 18 and 63-Activity projects respectively.

18-Activity project consists of 1 activity with 2-construction options, 10 activities with 3-construction options, 2 activities with 4-construction options, and 5 activities with 5-construction options. Number of possible scheduling alternatives of the 18-Activity project is; $5^5*4^2*3^{10}*2^1 \approx 5.9$ Billion. 63-activity project consists of 2 activities with 3-construction options, 15 activities with 4-construction options, and 46 activities with 5-construction options. As a result, total number of possible schedule combinations of the 63-activity project is $3^2*4^{15}*5^{46} \approx 1.37*10^{42}$. The significant difference between the numbers of scheduling combinations makes the optimization of 63-Activity project more difficult.

Crashing sequences provided by NAA are presented in Tables 3 to 6. Crash number column represents the sequence of crashing; crashed activity presents ID; project duration column gives the project duration obtained by the current construction modes of the alternatives; crashing cost column represents the cost of the executed crashing option; obtained benefit column represents the summation of the saved money by avoidance of penalty, reduced indirect costs or earned rewards; direct cost column represents the summation of the direct costs of the activities according to current construction modes; indirect cost column presents the summation of indirect costs and penalties; and total cost column provides the summation of direct and indirect costs.

Crashing sequence of the first case problem of the 18-Activity project is given in Table 3. At the end of the 10th crashing, a feasible crashing option is not found and decrease in unit crashing costs is searched in the subsequent crashing modes. Cost slope of the 12th activity decreases, therefore, first and

Table 3. Crashing Sequence of the First Case Problem of the 18-Activity Project

Crash Number	Crashed Activity	Project Duration	Crashing Cost	Obtained Benefit	Direct Cost	Indirect Cost	Total Cost
0	----	169	----	----	99740	33800	133540
1	10	161	80	1600	99820	32200	132020
2	9	159	50	400	99870	31800	131670
3	9	158	30	200	99900	31600	131500
4	10	156	50	400	99950	31200	131150
5	9	154	60	400	100010	30800	130810
6	9	151	60	600	100070	30200	130270
7	18	148	200	600	100270	29600	129870
8	1	145	300	600	100570	29000	129570
9	1	140	400	1000	100970	28000	128970
10	18	134	600	1200	101570	26800	128370
11	12*	128	750	1200	102320	25600	127920
12	12	126	250	400	102570	25200	127770

Table 4. Crashing Sequence of the Second Case Problem of the 18-Activity Project

Crash Number	Crashed Activity	Project Duration	Crashing Cost	Obtained Benefit	Direct Cost	Indirect Cost	Total Cost
0	----	169	----	----	99740	1213800	1313540
1	10	161	80	161600	99820	1052200	1152020
2	9	159	50	40400	99870	1011800	1111670
3	9	158	30	20200	99900	991600	1091500
4	10	156	50	40400	99950	951200	1051150
5	9	154	60	40400	100010	910800	1010810
6	9	151	60	60600	100070	850200	950270
7	18	148	200	60600	100270	789600	889870
8	1	145	300	60600	100570	729000	829570
9	1	140	400	101000	100970	628000	728970
10	18	134	600	121200	101570	506800	608370
11	17	128	1400	121200	102970	385600	488570
12	17	124	800	80800	103770	304800	408570
13	1	123	250	20200	104020	284600	388620
14	1	122	250	20200	104270	264400	368670
15	12	120	500	40400	104770	224000	328770
16	12	116	250	80800	105020	143200	248220
17	12	114	250	40400	105270	102800	208070
18	15	110	1000	80800	106270	22000	128270

second crashing options are executed together and the activity is crashed 6 days with a cost of $750. To represent dual crashing, the activity is shown as 12* in the 11th crashing in Table 3. NAA converges into global optimum for the first case problem of 18-Activity project in 12 crashing.

Table 4 represents the crashing sequence of the second case problem of the 18-Activity Project. Global optimum is obtained in 18 crashing attempts. In this case, first and second crashing options of the 12th activity are executed separately although crashing slope of the first option is $250/day. When the first crashing option of 12th activity is executed, project duration is 120 days and due to the $20,000 daily penalty, the aforementioned crashing option is feasible.

Global optima are obtained with very little computational demand for the two case problems. The algorithm obtains the global optimum in 12 and 18 crashing attempts for the first and second cases respectively. Trial schedule evaluation is not performed in both cases, which demonstrates that convergence of the heuristic algorithm is significantly faster than the meta-heuristic algorithms.

Table 5 represents the crashing sequence for the solution of the first case problem of the 63-Activity project. In this case, the heuristic algorithm cannot converge into global optimum and finds a local optimum, which is $2450 worse than the global optimum. NAA performs 20 crashing and 12 trial schedules. Computational demand increases slightly although the search domain of the 63-Activity project is significantly higher than the 18-Activity project.

Table 5. Crashing Sequence of the First Case Problem of the 63-Activity Project

Crash Number	Crashed Activity	Project Duration	Crashing Cost	Obtained Benefit	Direct Cost	Indirect Cost	Total Cost
0	---	708	---	---	3858620	1628400	5487020
1	63	707	200	2300	3858820	1626100	5484920
2	63	706	400	2300	3859220	1623800	5483020
3	41	700	3700	13800	3862920	1610000	5472920
4	63	699	700	2300	3863620	1607700	5471320
5	63	697	1900	4600	3865520	1603100	5468620
6	22	692	4850	11500	3870370	1591600	5461970
7	61	690	2250	4600	3872620	1587000	5459620
8	54	689	1300	2300	3873920	1584700	5458620
9	54	688	1950	2300	3875870	1582400	5458270
10	54	687	1450	2300	3877320	1580100	5457420
11	61	685	2500	4600	3879820	1575500	5455320
12	61	682	2550	6900	3882370	1568600	5450970
13	61	680	3700	4600	3886070	1564000	5450070
14	57	677	3450	6900	3889520	1557100	5446620
15	57	674	4350	6900	3893870	1550200	5444070
16	57	671	4150	6900	3898020	1543300	5441320
17	57	669	3650	4600	3901670	1538700	5440370
18	54	667	3100	4600	3904770	1534100	5438870
19	49	653	22200	32200	3926970	1501900	5428870
20	49	641	22300	27600	3949270	1474300	5423570

Table 6. Crashing Sequence of the Second Case Problem of the 63-Activity Project

Crash Number	Crashed Activity	Project Duration	Crashing Cost	Obtained Benefit	Direct Cost	Indirect Cost	Total Cost
0	---	708	0	0	3858620	2478000	6336620
1	63	707	200	3500	3858820	2474500	6333320
2	63	706	400	3500	3859220	2471000	6330220
3	41	700	3700	21000	3862920	2450000	6312920
4	63	699	700	3500	3863620	2446500	6310120
5	63	697	1900	7000	3865520	2439500	6305020
6	22	692	4850	17500	3870370	2422000	6292370
7	61	690	2250	7000	3872620	2415000	6287620
8	57	687	3450	13500	3876070	2404500	6280570
9	61	685	2500	7000	3878570	2397500	6276070
10	61	682	2550	10500	3881120	2387000	6268120
11	54	681	1300	3500	3882420	2383500	6265920
12	54	680	1950	3500	3884370	2380000	6264370
13	54	679	1450	3500	3885820	2376500	6262320
14	54	677	3100	7000	3888920	2369500	6258420
15	61	675	3700	7000	3892620	2362500	6255120
16	57	672	4350	10500	3896970	2352000	6248970
17	57	669	4150	10500	3901120	2341500	6242620
18	57	667	3650	7000	3904770	2334500	6239270
19	49	653	22200	49000	3926970	2285500	6212470
20	49	641	22300	42000	3949270	2243500	6192770
21	41	639	5650	7000	3954920	2236500	6191420
22	37	637	1600	7000	3956520	2229500	6186020
23	41, 25	633	8900	14000	3965420	2215500	6180920

Table 6 represents the crashing sequence for the solution of the second case problem of 63-Activity project. NAA cannot converge into global optimum and finds a local minimum, which is $4750 worse than the global optimum in 23 crashing and 73 trial-schedules. In the 23rd crashing step, a feasible crashing option cannot be obtained because of the uncritical paths with inadequate total float. The search algorithm attaches an uncritical algorithm, and produces a crashing option, which reduces the total project cost $5100.

Computational demand of executing a crashing step is more than the schedule evaluation of meta-heuristic algorithms. Backward pass of the schedule, formation of crashing options and checking feasibility of crashing options are required only in NAA. In order to compare computational demands, one crashing is assumed equivalent of three schedule evaluations of a meta-heuristic algorithm. As a result, computational demands of the case problems of 63-Activity project becomes 75 and 145 forward pass schedule evaluations. Similarly, computational demands of the case problems of the 18-Activity project become 39 and 57 forward pass schedules respectively. This comparison reveals that computational demand of NAA is significantly less than the meta-heuristic algorithms.

4. Discussion of Results

NAA is tested on 6 small-size, 1 medium-size, and 1 large-size networks. Proposed algorithm obtains global optimum in all of the case problems for the small and medium-size networks. NAA has significantly less computational and memory demand than meta-heuristic algorithms. However, meta-heuristic algorithms can provide better results if unlimited schedule evaluation is performed.

The 18-Activity project is previously analyzed extensively in the literature and global optimum of the problem is obtained by a few thousand-schedule evaluations by meta-heuristic algorithms (Elbeltagi *et al*., 2005; Bettemir, 2009). The 63-Activity project is previously analyzed by Bettemir (2009) and Sonmez and Bettemir (2012) and the global optimum of the network is obtained with 40% success rate in 500,000 schedule evaluations.

NAA converged into global optimum for the 2 case problems of 18-Activity project but missed the global optimum by $2450 and $4750 for the case problems of 63-Activity project. Obtained local optima for both cases are still satisfactory when the computational demand and the memory requirement of the algorithm are considered. Divergences from the optimum solution are 0.046% and 0.077% respectively. On the other hand, genetic algorithm and simulated annealing meta-heuristic algorithms diverged 2.61% and 2.50% at the end of 50,000 trials (Sonmez and Bettemir, 2012). Approximately, 200,000 schedule evaluations are required by Genetic Algorithm and Simulated Annealing in order to obtain the same local minima. This reveals that computational demand of the proposed NAA is less than one thousandth of the meta-heuristic algorithms.

Apart from convergence, elimination of local minima is also important. Meta-heuristic algorithms assign probability of survival according to the fitness of individuals. Measure of fitness for the TCT problems is the obtained saving in the total project cost. Probability of survival may not be distinctive for the good fit and poor fit individuals when the crashing costs range between $50 and $20,000. In this case, erroneously selected crashing option with a very low crashing cost may not be discerned. However, NAA clearly distinguishes the extremes of crashing costs and can be a useful tool for the elimination of local minimum.

NAA can be adapted to eliminate infeasible portions of the search domain to decrease the computational demand of meta-heuristic algorithms. Infeasible crashing options can easily be detected and selection of them can be restricted similar to Tabu search algorithm.

If meta-heuristic algorithms are implemented, number of required trials to converge into optimum or near-optimum solution increases exponentially when project size increases (Elbeltagi *et al*., 2005; Bettemir, 2009). On the contrary, case studies reveal that computational demand of NAA increases linearly. This property of the heuristic algorithm makes it more suitable for the optimization of large projects.

Number of critical paths increases through the step-by-step crashing procedure, and the number of crashing options, either. Due to elimination procedure, number of crashing combinations is kept within reasonable limits. At the final step of the second case of 18-Activity project, four critical paths are detected with only two feasible crashing options. Elimination keeps the memory and computational demands in reasonable limits; however, increases the risk of convergence into local-minima.

Convergence of the NAA into global optima is not guaranteed due to elimination and neglection of the crashing combinations. If crashing options were formed without narrowing the search space, more than billions of crashing options would be formed for large projects. Consequently, this situation is not feasible in terms of memory and processing power. On the other hand, NAA can be merged with network decomposition algorithm to detect feasible crashing options without neglecting. If this can be achieved NAA can be able to obtain global optimum for all projects in any size and complexity.

5. Conclusions

In this study, discrete TCT problem is solved by minimum cost-slope method based network analysis algorithm. The proposed algorithm is developed for Activity-on-Node (AoN) type networks, where definition of activity precedence relationships is easier. NAA searches the global optimum by a step-by-step crashing procedure. Crashing options are determined by searching through the feasible portion of the search domain. Presence of discrete crashing alternatives requires the uncritical activities to be included in the search domain. Even though the existence of a large search domain, NAA eliminates infeasible options and the crashing option with the minimum cost-slope is selected.

The proposed algorithm is tested in terms of capability of finding the global optimum and its rate of convergence. Test results revealed that the algorithm converges to optimum or near-optimum solutions significantly fast. Computational demand of NAA is considerably less than the meta-heuristic algorithms. In addition to this, implementation of the method is easier than integer-programming or branch-and-bound algorithms, and the method provides better solutions than heuristic algorithms. Furthermore, adaptation of NAA for the AoN type networks provides the possibility of combining it with meta-heuristic algorithms and developing new hybrid meta-heuristic algorithms. Consequently, analysis results show that the proposed NAA is a proper optimization method for the discrete TCT problem.

References

Afruzi, E. N., Najafi, A. A., Roghanian, E., and Mazinani, M. (2014). "A multi-objective imperialist competitive algorithm for solving discrete time, cost and quality trade-off problems with mode-identity and resource-constrained situations." *Computers and Operations Research*, Vol. 50, pp. 80-96, DOI: 10.1016/j.cor.2014.04.003.

Al Haj, R. A. and El-Sayegh, S. M. (2015). "Time–cost optimization model considering float-consumption impact." *Journal of Construction Engineering and Management*, Vol. 141, No. 5, DOI: 10.1061/(ASCE)CO.1943-7862.0000966.

Ammar, M. A. (2011). "Optimization of project time-cost trade-off problem with discounted cash flows." *Journal of Construction Engineering and Management*, Vol. 137, No. 1, pp. 65-71, DOI: 10.1061/(ASCE)CO.1943-7862.0000256.

Anagnostopoulos, K. P. and Kotsikas, L. (2010). "Experimental evaluation of simulated annealing algorithms for the time–cost trade-off problem." *Applied Mathematics and Computation*, Vol. 217, No. 1, pp. 260-270, DOI: 10.1016/j.amc.2010.05.056.

Azaron, A., Perkgoz, C., and Sakawa, M. (2005). "A genetic algorithm approach for the time-cost trade-off in PERT networks." *Applied Mathematics and Computation*, Vol. 168, No. 2, pp. 1317-1339, DOI: 10.1016/j.amc.2004.10.021.

Babu, A. J. G. and Suresh, N. (1996). "Project management with time, cost, and quality considerations." *European Journal of Operational Research*, Vol. 88, No. 2, pp. 320-327, DOI: 10.1016/0377-2217(94)00202-9.

Barber, T. J. and Boardman, J. T. (1988). "Knowledge–based project control employing heuristic optimisation." *IEE Proceedings*, Vol. 135, No. 8, pp. 529-538, DOI: 10.1049/ip-a-1.1988.0083.

Berman, E. B. (1964). "Resource allocation in a PERT network under continuous activity time-cost functions." *Management Science*, Vol. 10, No. 4, pp. 734-745, DOI: 10.1287/mnsc.10.4.734.

Bettemir, Ö. H. (2009). *Optimization of time-cost-resource trade-off problems in project scheduling using meta-heuristic algorithms*, Doctoral dissertation, Middle East Technical University.

Bidhandi, H. M. (2006). "A new approach based on the surrogating method in the project time compression problems." *Annals of Operations Research*, Vol. 143, No. 1, pp. 237-250, DOI: 10.1007/s10479-006-7385-y.

Cha, H. S. and Lee, D. G. (2015). "A case study of time/cost analysis for aged-housing renovation using a pre-made BIM database structure." *KSCE Journal of Civil Engineering*, Vol. 19, No. 4, pp. 841-852.

Chen, S. P. and Tsai, M. J. (2011). "Time–cost trade-off analysis of project networks in fuzzy environments." *European Journal of Operational Research*, Vol. 212, No. 2, pp. 386-397, DOI: 10.1016/j.ejor.2011.02.002.

Cheng, M. Y. and Tran, D. H. (2016). "An efficient hybrid differential evolution based serial method for multimode resource-constrained project scheduling." *KSCE Journal of Civil Engineering*, Vol. 20, No. 1, pp. 90-100.

Chiu, Y. S. P. and Chiu S. W. (2005). "Incorporating expedited time and cost of the end product into the product structure diagram." *International Journal of Machine Tools and Manufacture*, Vol. 45, No. 7, pp. 987-991, DOI: 10.1016/j.ijmachtools.2004.10.015.

De, P., Dunne, E. J., Ghosh, J. B., and Wells, C. E. (1995). "The discrete time-cost tradeoff problem revisited." *European Journal of Operational Research*, Vol. 81, No. 2, pp. 225-238, DOI: 10.1016/0377-2217(94)00187-H.

De, P., Dunne, E. J., Ghosh, J. B., and Wells, C. E. (1997). "Complexity of the discrete time-cost tradeoff problem for project networks." *Operations Research*, Vol. 45, No. 2, pp. 302-306, DOI: 10.1287/opre.45.2.302.

Demeulemeester, E. L., Herroelen, W. S., and Elmaghraby, S. E. (1996). "Optimal procedures for the discrete time/cost trade-off problem in project networks." *European Journal of Operational Research*, Vol. 88, No. 1, pp. 50-68, DOI: 10.1016/0377-2217(94)00181-2.

Demeulemeester, E., De Reyck, B., Foubert, B., Herroelen, W., and Vanhoucke, A. M. (1998). "New computational results on the discrete time/cost trade-off problem in project networks." *Journal of the Operational Research Society*, Vol. 49, No. 11, pp. 1153-1163, DOI: 10.2307/3010096.

Elazouni, A. M. and Metwally, F. G. (2007). "Expanding finance-based scheduling to devise overall-optimized project schedules." *Journal of Construction Engineering and Management*, Vol. 133, No. 1, pp. 86-90.

Elbeltagi, E., Hegazy, T., and Grierson, D. (2005). "Comparison among five evolutionary-based optimization algorithms." *Advanced Engineering Informatics*, Vol. 19, No. 1, pp. 43-53, DOI: 10.1016/j.aei.2005.01.004.

Eshtehardian, E., Afshar, A., and Abbasnia, R. (2009). "Fuzzy-based MOGA approach to stochastic time–cost trade-off problem." *Automation in Construction*, Vol. 18, No.5, pp. 692-701, DOI: 10.1016/j.autcon.2009.02.001.

Falk, J. E. and Horowitz, J. L. (1972). "Critical path problems with concave cost-time curves." *Management Science*, Vol. 19, No. 4-part-1, pp. 446-455, DOI: 10.1287/mnsc.19.4.446.

Fathi, H. and Afshar, A. (2010). "GA-based multi-objective optimization of finance-based construction project scheduling." *KSCE Journal of Civil Engineering*, Vol. 14, No. 5, pp. 627-638.

Feng, C. W., Liu, L., and Burns, S. A. (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." *Journal of Computing in Civil Engineering*, Vol. 11, No. 3, pp. 184-189, DOI: 10.1061/(ASCE)0887-3801(1997)11:3(184).

Fulkerson, D. R. (1961). "A network flow computation for project cost curves." *Management Science*, Vol. 7, No. 2, pp. 167-178, DOI: 10.1287/mnsc.7.2.167.

Geem, Z. W. (2010). "Multiobjective optimization of time-cost trade-off using harmony search." *Journal of Construction Engineering and Management*, Vol. 136, No. 6, pp. 711-716, DOI: 10.1061/(ASCE)CO.1943-7862.0000167.

Ghoddousi, P., Eshtehardian, E., Jooybanpour, S., and Javanmardi, A. (2013). "Multi-mode resource-constrained discrete time–cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm." *Automation in Construction*, Vol. 30, pp. 216-

227, DOI: 10.1016/j.autcon.2012.11.014.

Goyal, S. K. (1996). "A simple time-cost tradeoff algorithm." *Production Planning and Control*, Vol. **7**, No. 1, pp. 104-106, DOI: 10.1080/09537289608930331.

Goyal, S. K. (1975). "A note on a simple CPM time-cost tradeoff algorithm." *Management Science*, Vol. 21, No. 6, pp. 718-722, DOI: 10.1287/mnsc.21.6.718.

Hazır, Ö., Haouari, M., and Erel, E. (2010a). "Discrete time/cost trade-off problem: A decomposition-based solution algorithm for the budget version." *Computers and Operations Research*, Vol. 37, No. 4, pp. 649-655, DOI: 10.1016/j.cor.2009.06.009.

Hazır, Ö., Haouari, M., and Erel, E. (2010b). "Robust scheduling and robustness measures for the discrete time/cost trade-off problem." European Journal of Operational Research, Vol. 207, No. 2, pp. 633-643, DOI: 10.1016/j.ejor.2010.05.046.

Hegazy, T. (1999a). "Optimization of construction time-cost trade-off analysis using genetic algorithms." *Canadian Journal of Civil Engineering*, Vol. 26, No. 6, pp. 685-697, DOI: 10.1139/l99-031.

Hegazy, T. (1999b). "Optimization of resource allocation and leveling using genetic algorithms." *Journal of Construction Engineering and Management*, Vol. 125, No. 3, pp. 167-175, DOI: 10.1061/(ASCE)0733-9364(1999)125:3(167).

Kalhor, E., Khanzadi, M., Eshtehardian, E., and Afshar, A. (2011). "Stochastic time–cost optimization using non-dominated archiving ant colony approach." *Automation in Construction*, Vol. 20, No. 8, pp. 1193-1203, DOI: 10.1016/j.autcon.2011.05.003.

Ke, H., Ma, W., and Ni, Y. (2009). "Optimization models and a GA-based algorithm for stochastic time-cost trade-off problem." *Applied Mathematics and Computation*, Vol. 215, No. 1, pp. 308-313, DOI: 10.1016/j.amc.2009.05.004.

Kelley Jr, J. E. and Walker, M. R. (1959). "Critical-path planning and scheduling." *In Proc. of eastern joint IRE-AIEE-ACM computer conference Association for Computing Machinery December 1-3*, 1959, Vol. 1, pp. 160-173.

Kelley Jr, J. E. (1961). "Critical-path planning and scheduling: Mathematical basis." *Operations Research*, Vol. 9, No. 3, pp. 296-320, DOI: 10.1287/opre.9.3.296.

Khang, D. B. and Myint, Y. M. (1999). "Time, cost and quality trade-off in project management: A case study." *International Journal of Project Management*, Vol. 17, No. 4, pp. 249-256, DOI: 10.1016/S0263-7863(98)00043-X.

Kim, J., Kang, C., and Hwang, I. (2012). "A practical approach to project scheduling: considering the potential quality loss cost in the time–cost tradeoff problem." *International Journal of Project Management*, Vol. 30, No. 2, pp. 264-272, DOI: 10.1016/j.ijproman.2011.05.004.

Koo, C., Hong, T., and Kim, S. (2015). "An integrated multi-objective optimization model for solving the construction time-cost trade-off problem." *Journal of Civil Engineering and Management*, Vol. 21, No. 3, pp. 323-333, DOI: 10.3846/13923730.2013.802733.

Lee, H. S., Roh, S., Park, M. S., and Ryu, H. G. (2010). "Optimal option selection for finishing works of high-rise building." *KSCE Journal of Civil Engineering*, Vol. 14, No. 5, pp. 639-651.

Leu, S. S., Chen, A. T., and Yang, C. H. (2001). "A GA-based fuzzy optimal model for construction time–cost trade-off." *International Journal of Project Management,* Vol. 19, No. 1, pp. 47-58, DOI: 10.1016/S0263-7863(99)00035-6.

Li, H., Cao, J. N., and Love, P. E. D. (1999). "Using machine learning and GA to solve time-cost trade-off problems." *Journal of Construction Engineering and Management*, Vol. 125, No. 5, pp. 347-353, DOI: 10.1061/(ASCE)0733-9364(1999)125:5(347).

Li, C. and Wang, K. (2009). "The risk element transmission theory research of multi-objective risk-time-cost trade-off." *Computers and Mathematics with Applications*, Vol. 57, No. 11, pp. 1792-1799, DOI: 10.1016/j.camwa.2008.10.051.

Liberatore, M. J. and Pollack-Johnson, B. (2006). "Extending project time–cost analysis by removing precedence relationships and task streaming." *International Journal of Project Management,* Vol. 24, No. 6, pp. 529-535, DOI: 10.1016/j.ijproman.2006.04.004.

Liu, L., Burns, S. A., and Feng, C. W. (1995). "Construction time-cost trade-off analysis using LP/IP hybrid method." *Journal of Construction Engineering and Management*, Vol. 121, No. 4, pp. 446-454, DOI: 10.1061/(ASCE)0733-9364(1995)121:4(446).

Liu, S. S. and Wang, C. J. (2008). "Resource-constrained construction project scheduling model for profit maximization considering cash flow." *Automation in Construction*, Vol. 17, No. 8, pp. 966-974, DOI: 10.1016/j.autcon.2008.04.006.

Monghasemi, S., Nikoo, M. R., Fasaee, M. A. K., and Adamowski, J. (2015). "A novel multi criteria decision making model for optimizing time–cost–quality trade-off problems in construction projects." *Expert Systems with Applications*, Vol. 42, No. 6, pp. 3089-3104, DOI: 10.1016/j.eswa.2014.11.032.

Moussourakis, J. and Haksever, C. (2004). "Flexible model for time/cost tradeoff problem." *Journal of Construction Engineering and Management*, Vol. 130, No. 3, pp. 307-314, DOI: 10.1061/(ASCE)0733-9364(2004)130:3(307).

Moussourakis, J. and Haksever, C. (2007). "Models for accurate computation of earliest and latest start times and optimal compression in project networks." *Journal of Construction Engineering and Management,* Vol. 133, No. 8, pp. 600-608, DOI: 10.1061/(ASCE)0733-9364(2007)133:8(600).

Mungle, S., Benyoucef, L., Son, Y. J., and Tiwari, M. K. (2013). "A fuzzy clustering-based genetic algorithm approach for time–cost–quality trade-off problems: A case study of highway construction project." *Engineering Applications of Artificial Intelligence*, Vol. 26, No. 8, pp. 1953-1966, DOI: 10.1016/j.engappai.2013.05.006.

Ng, S. T. and Zhang, Y. (2008). "Optimizing construction time and cost using ant colony optimization approach." *Journal of Construction Engineering and Management*, Vol. 134, No. 9, pp. 721-728, DOI: 10.1061/(ASCE)0733-9364(2008)134:9(721).

Pathak, B. K. and Srivastava, S. (2014). "Integrated ANN-HMH approach for nonlinear time-cost tradeoff problem." *International Journal of Computational Intelligence Systems*, Vol. 7, No. 3, pp. 456-471, DOI: 10.1080/18756891.2014.891368.

Rogalska, M., Bożejko, W., and Hejducki, Z. (2008). "Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling." *Automation in Construction*, Vol. 18, No. 1, pp. 24-31, DOI: 10.1016/j.autcon.2008.04.002.

Rostami, M., Moradinezhad, D., and Soufipour, A. (2014). "Improved and competitive algorithms for large scale multiple resource-constrained project-scheduling problems." *KSCE Journal of Civil Engineering*, Vol. 18, No. 5, pp. 1261-1269.

Said, S. S. and Haouari, M. (2015). "A hybrid simulation-optimization approach for the robust Discrete Time/Cost Trade-off Problem." *Applied Mathematics and Computation*, Vol. 259, pp. 628-636, 10.1016/j.amc.2015.02.092.

Sakellaropoulos, S. and Chassiakos, A. P. (2004). "Project time–cost analysis under generalised precedence relations." *Advances in Engineering Software*, Vol. 35, No. 10, pp. 715-724, DOI: 10.1016/j.advengsoft.2004.03.017.

Schwarze, J. (1980). "An algorithm for hierarchical reduction and decomposition of a directed graph." *Computing*, Vol. 25, No. 1, pp. 47-57, DOI: 10.1007/BF02243881.

Siemens, N. (1971). "A Simple CPM Time-Cost trade off algorithm." *Management Science*, Vol. 17, No. 6, pp. B354-B363, DOI: 10.1287/mnsc.17.6.B354.

Siemens, N. and Gooding, C. (1975). "Reducing project duration at minimum cost: A time-cost tradeoff algorithm." *Omega*, Vol. 3, No. 5, pp. 569-581, DOI: 10.1016/0305-0483(75)90030-4.

Son, J., Hong, T., and Lee, S. (2013). "A mixed (continuous+ discrete) time-cost trade-off model considering four different relationships with lag time." *KSCE Journal of Civil Engineering*, Vol. 17, No. 2, pp. 281-291, DOI: 10.1007/s12205-013-1506-3.

Sonmez, R. and Bettemir, Ö. H. (2012). "A hybrid genetic algorithm for the discrete time–cost trade-off problem." *Expert Systems with Applications*, Vol. 39, No. 13, pp. 11428-11434, DOI: 10.1016/j.eswa.2012.04.019.

Tareghian, H. R., and Taheri, S. H. (2006). "On the discrete time, cost and quality trade-off problem." *Applied Mathematics and Computation*, Vol. 181, No. 2, pp. 1305-1312, DOI: 10.1016/j.amc.2006.02.029.

Tavana, M., Abtahi, A. R., and Khalili-Damghani, K. (2014). "A new multi-objective multi-mode model for solving preemptive time–cost–quality trade-off project scheduling problems." *Expert Systems with Applications,* Vol. 41, No. 4, pp. 1830-1846, DOI: 10.1016/j.eswa.2013.08.081.

Vanhoucke, M. (2005). "New computational results for the discrete time/cost trade-off problem with time-switch constraints." *European Journal of Operational Research*, Vol. 165, No. 2, pp. 359-374, DOI: 10.1016/j.ejor.2004.04.007.

Vanhoucke, M. and Debels, D. (2007). "The discrete Time/Cost trade off problem: Extensions and heuristic procedures." *Journal of Scheduling*, Vol. 10, Nos. 4-5, pp. 311-326, DOI: 10.1007/s10951-007-0031-y.

Xu, J., Zheng, H., Zeng, Z., Wu, S., and Shen, M. (2012). "Discrete time–cost–environment trade-off problem for large-scale construction systems with multiple modes under fuzzy uncertainty and its application to Jinping-II Hydroelectric Project." *International Journal of Project Management*, Vol. 30, No. 8, pp. 950-966, DOI: 10.1016/j.ijproman.2012.01.019.

Yang, I. T. (2005). "Chance-constrained time–cost tradeoff analysis considering funding variability." *Journal of Construction Engineering and Management*, Vol. 131, No. 9, pp. 1002-1012, DOI: 10.1061/(ASCE)0733-9364(2005)131:9(1002).

Yang, I. T. (2007). "Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis." *Journal of Construction Engineering and Management*, Vol. 133, No. 7, pp. 498-505, DOI: 10.1061/(ASCE)0733-9364(2007)133:7(498).

Yang, I. T. (2011). "Stochastic time–cost tradeoff analysis: A distribution-free approach with focus on correlation and stochastic dominance." *Automation in Construction*, Vol. 20, No. 7, pp. 916-926, DOI: 10.1016/j.autcon.2011.03.008.

Zhang, H. and Xing, F. (2010). "Fuzzy-multi-objective particle swarm optimization for time–cost–quality tradeoff in construction." *Automation in Construction*, Vol. 19, No. 8, pp. 1067-1075, DOI: 10.1016/j.autcon.2010.07.014.

Zhang, Y. and Thomas Ng, S. (2012). "An ant colony system based decision support system for construction time-cost optimization." *Journal of Civil Engineering and Management*, Vol. 18, No. 4, pp. 580-589, DOI: 10.3846/13923730.2012.704164.

Zheng, D. X., Ng, S. T., and Kumaraswamy, M. M. (2004). "Applying a genetic algorithm-based multiobjective approach for time-cost optimization." *Journal of Construction Engineering and Management*, Vol. 130, No. 2, pp. 168-176, DOI: 10.1061/(ASCE)0733-9364(2004)130:2(168).

Zheng, D. X., Ng, S. T., and Kumaraswamy, M. M. (2005). "Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time–cost optimization." *Journal of Construction Engineering and Management,* Vol. 131, No. 1, pp. 81-91, DOI: 10.1061/(ASCE)0733-9364(2005)131:1(81).

Zheng, D. X. and Ng, S. T. (2005). "Stochastic time–cost optimization model incorporating fuzzy sets theory and nonreplaceable front." *Journal of Construction Engineering and Management*, Vol. 131, No. 2, pp. 176-186, DOI: 10.1061/(ASCE)0733-9364(2005)131:2(176).