

Multi-objective optimization of discrete time–cost tradeoff problem in project networks using non-dominated sorting genetic algorithm

Mohammadreza Shahriari¹

Received: 11 February 2016 / Accepted: 17 March 2016 / Published online: 29 March 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract The time–cost tradeoff problem is one of the most important and applicable problems in project scheduling area. There are many factors that force the managers to crash the time. This factor could be early utilization, early commissioning and operation, improving the project cash flow, avoiding unfavorable weather conditions, compensating the delays, and so on. Since there is a need to allocate extra resources to short the finishing time of project and the project managers are intended to spend the lowest possible amount of money and achieve the maximum crashing time, as a result, both direct and indirect costs will be influenced in the project, and here, we are facing into the time value of money. It means that when we crash the starting activities in a project, the extra investment will be tied in until the end date of the project; however, when we crash the final activities, the extra investment will be tied in for a much shorter period. This study is presenting a two-objective mathematical model for balancing compressing the project time with activities delay to prepare a suitable tool for decision makers caught in available facilities and due to the time of projects. Also drawing the scheduling problem to real world conditions by considering nonlinear objective function and the time value of money are considered. The presented problem was solved using NSGA-II, and the effect of time compressing reports on the non-dominant set.

Keywords Time–cost tradeoff · Time value of money · Crashing · NSGA-II · Multi-objective problem · AOA network

Introduction

Discrete time–cost tradeoff problem (DTCTP) has many applications, and a lot of research has been conducted on this area. In these studies, customer needs to get services in shorter time periods and necessity of reducing the project cost were considered together, and this approach raised the importance of DTCTP for business owners and researchers. In 1991, Hindelang and Muth presented the DTCTP for the first time (Hindelang and Muth 1979). Prabuddha et al. (1997) and Vladimir et al. (2001) showed that this problem belongs to NP-hard problem. One of the basic assumptions in this problem is that the activities cost is a function of the activities duration, which is a decision variable. The lower limit of duration is crash duration, and the upper limit of duration is normal duration. Kelley and Walker (1959), Fulkerson (1961), Kelly (1961), Ford and Fulkerson (1962), Siemens (1971), Goyal (1975), and Elmaghraby and Salem (1981) presented the linear mathematical models and Moder et al. (1983) considered continuous activities cost. DTCTP was solved using many different exact methods, such as dynamic programming (Hindelang and Muth 1979), enumeration algorithm (Patterson and Harvey 1979), and branch and bound (Demeulemeester et al. 1996, 1998; Erenguc et al. 2001), but none of the methods could solve the DTCTP in large scale, so many researchers decided to use heuristics and metaheuristic algorithms to solve this problem. Akkan (1998) used a heuristic algorithm based on Lagrange released, while Liu et al. (2000) used genetic algorithm (GA) for solving DTCTP and Elmaghraby and

✉ Mohammadreza Shahriari
shahriari@iau.ae; Shahriari.mr@gmail.com

¹ Faculty of Management, South Tehran Branch, Islamic Azad University, Tehran, Iran

Kamburowski (1992) considered reward and penalty on objective function. Ann and Erenguc (1998) introduced activities compression, and Van Slyke was the first researcher who used the Monte Carlo simulation method for compressing activities. It has many advantages, such as longer projects time, flexibility of selecting distribution functions of activities time, and ability to calculate the value of the path critically (Van Slyke 1963), and DTCTP was considered with time switch in it (Vanhoucke et al. 2002). All the presented researchers have only one objective function, and the aim of all is decreasing the project cost or time. The researchers found that for determining the duration of activities, many things must be considered. For example, one may wish to finish the project in shortest time with minimum cost. To this effect, two finishing time and project cost objective functions can be considered in a scalar unique objective function (Sasaki and Gen 2003a, b), and can solve this multi-objective problem using the LP-metric method (Pasandideh et al. 2011). Mohammadreza Shahriari et al. introduced a mathematical model for time–cost tradeoff problem with budget limitation based on a mixed-integer programming, considering the time value of money (Mohammadreza Shahriari et al. 2010). Jaeho Son et al. (2013) presented a new formulation technique is introduced to merge the two independent scenarios mathematically, and it guarantees the optimal solution. Öncü - Hazır (2014) presented a mathematical model to support project managers from a wide range of industries in scheduling activities to minimize deviations from project goals. Tiwari and Johari (2015) introduced an approach, which was experimented on several case studies that proved its usefulness. The intertwined approach using simple and popular Microsoft Office tools (Excel and MSP) is logical, fast, and provides a set of feasible project schedule meeting the deadline and that do not violate resource limits. Mario Vanhoucke extend the standard electromagnetic metaheuristic with problem specific features and investigate the influence of various EM parameters on the solution quality (Vanhoucke 2014).

Kaveh et al. (2015) showed that two new metaheuristic algorithms, charged system search (CSS) and colliding body optimization (CBO), are utilized for solving this problem. The results show that both of these algorithms find reasonable solutions; however, CBO could find the result in a less computational time having a better quality.

Ke et al. proposed a model to deal with an intelligent algorithm combining stochastic simulations and GA, where a stochastic simulation technique is employed to estimate random functions and GA is designed to search optimal schedules under different decision-making criteria (Ke et al. 2012).

In this study, we used NSGA-II for solving this two-objective problem.

The assumption of activities compressing in time–cost tradeoff problem was presented to achieve the specific solutions. In this solution, with spending money, the project time will decrease. Generally, delay in finishing the projects is inevitable. The World Commission on Dams (WCD) made a research on 99 large projects and reported that only 50 of these projects finished in due time, 30 % of them have 1–2 years delay, and 20 % have more than 2 years delay (4 projects have more than 10 years delay), and the main reasons of delays are financial problems, inefficiency of contractors and operation management, unreal scheduling, and employers dissatisfaction (WCD 2000). Unlike the studies in this area, we consider the “maximum duration” for the upper level of activities duration, whatever moving from normal time to compressed time caused more compressing cost, and whatever moving from normal time to presented upper bound of the time caused more saving of money. The final front of the solutions in NSGA-II presents a variety of solutions for decision makers, and they have opportunity to select a proper solution based on the available budget and appropriate time for the project. Another problem considered in this study is the time value of money. In large projects, it is important to know the proper time of spending or saving the money due to the time value of money. This study has five parts. The second part presents a mathematical model. Third part deals with solving algorithm. In part 4, a numerical example is presented, and the final part belongs to conclusion and further studies.

Mathematical model

Nomenclatures

t_i	Happening time of event i
$D_{f(ij)}$	Minimum allowed time of activity $i - j$
$D_{n(ij)}$	Normal time for activity $i - j$
$D_{m(ij)}$	Maximum allowed time of activity $i - j$
d_{ij}	Scheduled (actual) time of activity $i - j$
H	Indirect (overhead) project cost
K_n	Direct project cost
$C_{f(ij)}$	Compressing cost of activity $i - j$
$C_{n(ij)}$	Normal cost of activity $i - j$
$C_{m(ij)}$	The cost of delaying in activity $i - j$
C_{ij}	Compressing cost rate of activity $i - j$
C'_{ij}	Saving rate of delaying for activity $i - j$
t_{Max}	Maximum allowed time for finishing the project
C_{Max}	Maximum available budget
I_0	Interest rate

$$\begin{aligned}
 y_{ij} &= \begin{cases} 1 & \text{If the activity } i - j \text{ compressed} \\ 0 & \text{Otherwise} \end{cases} \\
 y'_{ij} &= \begin{cases} 1 & \text{If the activity } i - j \text{ has delay} \\ 0 & \text{Otherwise} \end{cases} \\
 y''_{ij} &= \begin{cases} 1 & \text{If the activity } i - j \text{ has been done in normal time} \\ 0 & \text{Otherwise} \end{cases}
 \end{aligned}$$

Problem definitions

When the beginning activities of a project compressed, the compressing budget would be involved until the finishing project time, and this value of money would be involved for shorter time. This effect is obvious for the money that is involved with delaying in activities time, so the time value of money is an effective factor in this area. In the presented model, we used the time value of money to calculate the best time for compressing or delaying the activities. We added the compressing cost to the total cost function and subtract the venue of saving money of delaying on the activities from the total cost function, so the Pert network make better tradeoff between time and cost, considering the time value of money. The cost function contains direct cost, overhead cost, compressing cost, and delaying cost as follows:

$$\begin{aligned}
 Z_1 &= H(t_n - t_1) + K_n + \sum_i \sum_j C_{ij} \cdot y_{ij} \{D_{n(ij)} - d_{ij}\} \\
 &\quad - \sum_i \sum_j C'_{ij} \cdot y'_{ij} \{D_{n(ij)} - d_{ij}\} \tag{1}
 \end{aligned}$$

For interpolation of the equations that contain the time value of money for compressing, we act as follows. Consider some of the activities have been compressed, and for each time unit of compression, C_{ij} is the unit of money spend, and therefore, $\sum_i \sum_j C_{ij}(t_n - t_i)$ is the total money that spends for compression involved from day t_i to day t_n (finishing time of project). For calculating the saving money of delaying, we act the same, so we have:

$$\begin{aligned}
 Z_1 &= H(t_n - t_1) + K_n \\
 &\quad + \sum_i \sum_j C_{ij} \cdot y_{ij} \{ \{D_{n(ij)} - d_{ij}\} + (t_n - t_i)I_0 \} \\
 &\quad - \sum_i \sum_j C'_{ij} \cdot y'_{ij} \{ \{D_{n(ij)} - d_{ij}\} + (t_n - t_i)I_0 \} \tag{2}
 \end{aligned}$$

Much research has been conducted on the effects of the project cost and the cost of compression. Ameen (1987) presented the definition of cost gradient and offered a new technique, ‘‘CAPERTSIM’’, for decision making under uncertainty in time–cost tradeoff in compressing and the relation between them. In this research, a cost gradient index is defined as the ratio of money spend to compressing

value and considering the time duration of each activity as a probabilistic variable presents a simple simulation-based model for time compressing.

In this study, we present a nonlinear relation between activities durations and C_{ij} and C'_{ij} , as shown in Fig. 1:

According to Fig. 1, if the relation between compressing time and compressing cost was an exponential distribution, then we would have:

$$C(d_{ij}) = \alpha e^{-\beta \cdot d_{ij}} \tag{3}$$

If we had the coordination of D_n and D_f , the values of α and β could be calculated as follows:

$$\beta = \frac{Ln\left(\frac{C_n}{C_f}\right)}{D_f - D_n} \tag{4}$$

$$\alpha = e^{\{Ln(C_n) + \beta \cdot D_n\}} \tag{5}$$

Using the same approach for the saving coefficient, we have:

$$C'(d_{ij}) = \alpha' e^{-\beta' \cdot d_{ij}} \tag{6}$$

$$\beta' = \frac{Ln\left(\frac{C'_m}{C'_n}\right)}{D_n - D_m} \tag{7}$$

$$\alpha' = e^{\{Ln(C'_n) + \beta' \cdot D_m\}} \tag{8}$$

The second objective function is considered as the finishing time of the project as follows:

$$Z_2 = t_n \tag{9}$$

Mathematical model

$$\begin{aligned}
 \text{Min } Z_1 &= H(t_n - t_1) + K_n + \sum_i \sum_j y_{ij} \cdot \alpha_{ij} e^{-\beta_{ij} \cdot d_{ij}} \\
 &\quad \{1 + (t_n - t_i)I_0\} \\
 &\quad - \sum_i \sum_j y'_{ij} \cdot \alpha'_{ij} e^{-\beta'_{ij} \cdot d_{ij}} \{1 + (t_n - t_i)I_0\} \tag{10}
 \end{aligned}$$

$$\text{Min } Z_2 = t_n \tag{11}$$

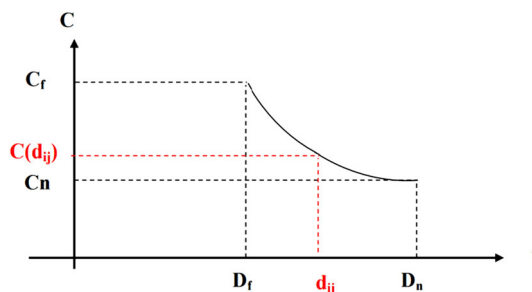


Fig. 1 Relation between activities durations and C_{ij} (C'_{ij})

$$S.t : \tag{12}$$

$$t_j - t_i \geq d_{ij} \tag{12}$$

$$y_{ij} \cdot D_{f(ij)} \leq y_{ij} \cdot d_{ij} \leq y_{ij} \cdot D_{n(ij)}; \quad \forall i,j \tag{13}$$

$$y'_{ij} \cdot D_{f(ij)} \leq y'_{ij} \cdot d_{ij} \leq y'_{ij} \cdot D_{n(ij)}; \quad \forall i,j \tag{14}$$

$$y''_{ij} \cdot d_{ij} = D_{n(ij)} \cdot y''_{ij}; \quad \forall i,j \tag{15}$$

$$y_{ij} + y'_{ij} + y''_{ij} = 1; \quad \forall i,j \tag{16}$$

$$t_n \leq t_{Max} \tag{17}$$

$$H(t_n - t_1) + \sum_i \sum_j y_{ij} \cdot \alpha_{ij} e^{-\beta_{ij} \cdot d_{ij}} \{1 + (t_n - t_i)I_0\} - \sum_i \sum_j y'_{ij} \cdot \alpha'_{ij} e^{-\beta'_{ij} \cdot d_{ij}} \{1 + (t_n - t_i)I_0\} \leq C_{Max} \tag{18}$$

$$t_i \geq 0 \tag{19}$$

Equation (10) is the first objective function that deals with minimizing project cost, considering the time value of money. Equation 11 is the second objective function that minimizes the finishing time of project. Considering this two objective function together moves the problem to balance the compression and delaying of the project activities.

Solving algorithm

Because the presented problem belongs to NP-hard problem, the metaheuristic algorithms were used to solve the problem. One of these algorithms is GA. Sou-Sen Leu (2000) used GA based on fuzzy theory and considered the effects of uncertainty of the parameters in time–cost tradeoff. Li and Cao (1999) created “MLGAS” technique by combining GA and Learning machines method and claimed that when the relation between activities and cost is nonlinear, the presented technique has better solutions. Heng et al. (Burns 1994) presented a new algorithm based on GA and prepared a computational program to evaluate the efficiency of the presented algorithm. This research is the most complete research in this area that used GA for time–cost tradeoff. Chau and Chan (1997) claimed that exact methods, such as DP and LP, have very long solving time and are not suitable for solving time–cost tradeoff, so he developed the GA for this problem and considered the resource constraints for each activity.

The single objective optimization algorithm could find the better solution for one objective function, and if more than one objective considered for a problem, these algorithms have no efficiency. When the problem has more than one objective function, the results can be shown as a Pareto front of non-dominant solutions. This Pareto front contains the solutions that are

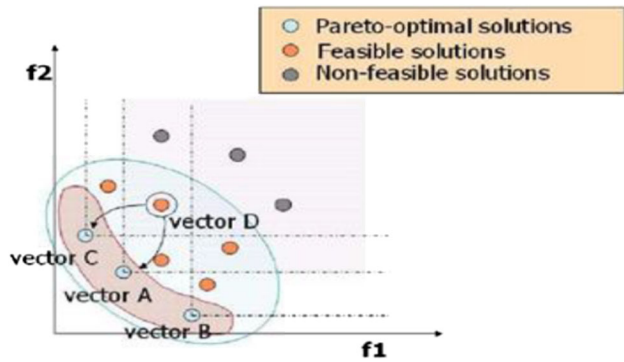


Fig. 2 Pareto sets in multi-objective problems (Deb 2001)

acceptable operation for all objective functions. When we have Pareto front of solutions, none of the solutions in Pareto frons has better result for all objective function comparing with other solutions in Pareto front, and we have not a single optimal solution (Tavakkoli-Moghaddam et al. 2008). Figure 2 shows the Pareto sets in multi-objective problems (Deb 2001).

NSGA-II algorithm

Non-dominated sorting genetic algorithm (NSGA-II) is one of the most efficient and famous multi-objective algorithms, which was presented by Deb (2001) and Zade et al. (2014) and proved its usefulness in multi-objective problems (Deb et al. 2002). The NSGA-II can convergent with Pareto sets of solutions, and the results could spread to all sets. NSGA-II uses non-dominant sorting for convergent confidence and also crowding distance for cutting the bad solutions for earning better solutions (Gen and Cheng 1997; Amiri and Khajeh 2016). Totally, its higher

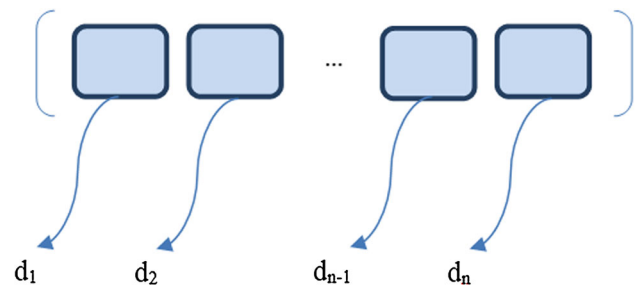
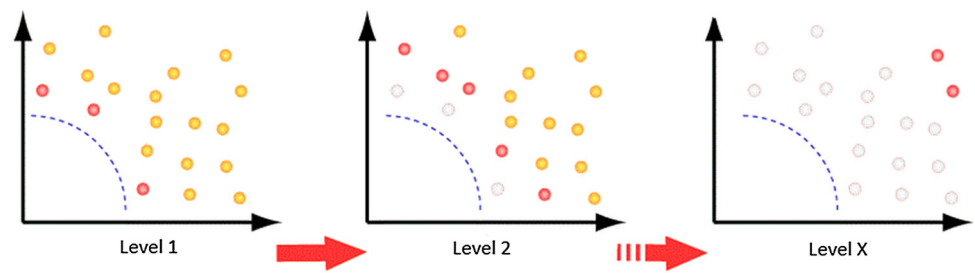


Fig. 3 Problem chromosome structure

Table 1 NSGA-II parameters value

<i>npop</i>	50
<i>P_C</i>	0.7
<i>P_M</i>	0.3
<i>nIt</i>	100

Fig. 4 Procedure on non-dominant sorting



competence makes this algorithm a good selection for multi-objective problems.

The steps of NSGA-II are as follows.

Initialization

The structure of the chromosome that used in this study is shown in Fig. 3. This chromosome is a $1 \times n$ matrix, and each genome in this chromosome is duration of activities.

In addition, the initial information of this algorithm is:

- npop* The population size represents the number of chromosomes in each iteration of algorithm.
- P_C Probability of crossover operator that represents the number of parents in the mating pool.
- P_M Probability of mutation operator that represents the number of chromosomes mutating in each iteration of algorithm.
- nIt* Maximum algorithm iterations.

The values of these parameters are shown if Table 1.

Fast non-dominant sorting and crowding distance

In this step, all chromosomes ranked using fast non-dominant sorting and crowding distance concepts. In fast non-dominant, sorting the population is sorted based on domination concept. Each solution in this step is compared with all other chromosomes and determines which one is dominant or non-dominant. Finally, we have a set of non-dominant solutions that forms the first boundary of solutions. For determining the second boundary, the solutions that located in first boundary ignore and the procedure is repeated again. This procedure runs until all solutions are located in solution boundaries. In this procedure, the worst situation happened when each boundary contains one solution. In this situation, the complexity of algorithm is

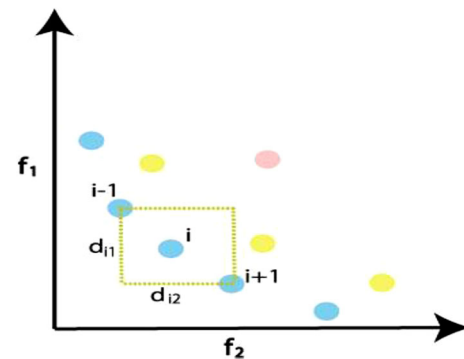


Fig. 5 Crowding distance of a specific solution (Deb 2001)

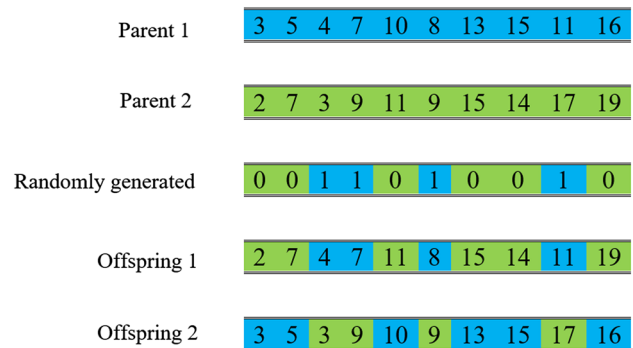


Fig. 6 Crossover operator

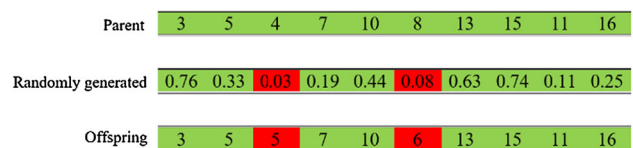


Fig. 7 Mutation operator

Fig. 8 Mechanism of NSGA-II evolution

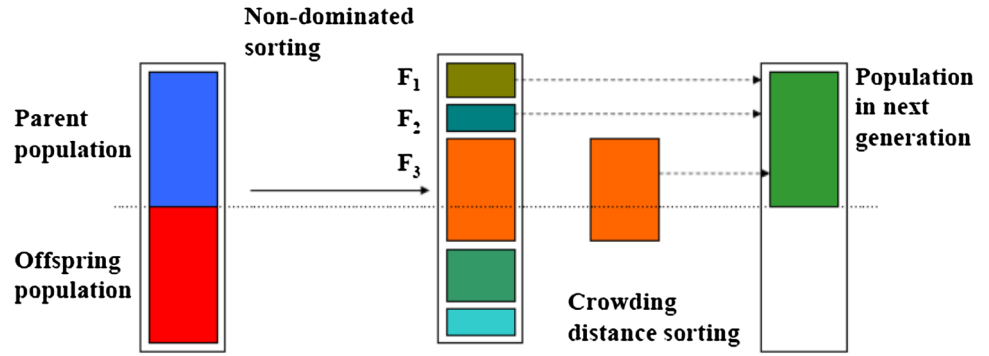


Table 2 Value of example parameters

Act	(D_f)	(D_n)	(D_m)	(C_f)	(C_n)	(C_m)	Act	(D_f)	(D_n)	(D_m)	(C_f)	(C_n)	(C_m)
A1	3	8	10	2000	500	375	A22	4	9	11	4050	1900	1425
A2	3	6	8	2050	850	637	A23	6	9	11	6020	500	375
A3	3	5	6	2480	480	360	A24	6	8	10	1900	360	270
A4	10	15	19	1830	380	285	A25	2	5	6	2070	870	652
A5	4	8	10	2100	300	225	A26	5	7	9	4000	500	375
A6	11	12	15	1890	690	517	A27	2	6	8	15,020	1180	885
A7	5	9	11	320	200	150	A28	6	15	19	64,600	7000	5250
A8	4	11	14	25,000	1200	900	A29	5	12	15	3710	700	525
A9	5	6	8	770	230	172	A30	2	4	5	470	370	277
A10	6	8	10	615	175	131	A31	4	8	10	7720	920	690
A11	3	7	9	6420	2300	1725	A32	14	19	24	16,800	5000	3750
A12	12	19	24	19,980	5700	4275	A33	3	5	6	1110	940	705
A13	3	5	6	3150	550	412	A34	7	12	15	27,550	5300	3975
A14	2	4	5	525	125	94	A35	5	7	9	5600	3400	2550
A15	11	13	16	3520	660	495	A36	9	11	14	3580	300	225
A16	6	9	11	2535	885	664	A37	4	8	10	8690	770	577
A17	4	7	9	12,660	3000	2250	A38	6	11	14	2190	690	517
A18	6	9	11	610	250	187	A39	1	5	6	6520	3000	2250
A19	10	13	16	9980	980	735	A40	6	8	10	2940	2200	1650
A20	5	7	9	3440	1440	1080	A41	4	9	11	12,600	1100	825
A21	8	11	14	3970	280	210	A42	3	8	10	3850	600	450

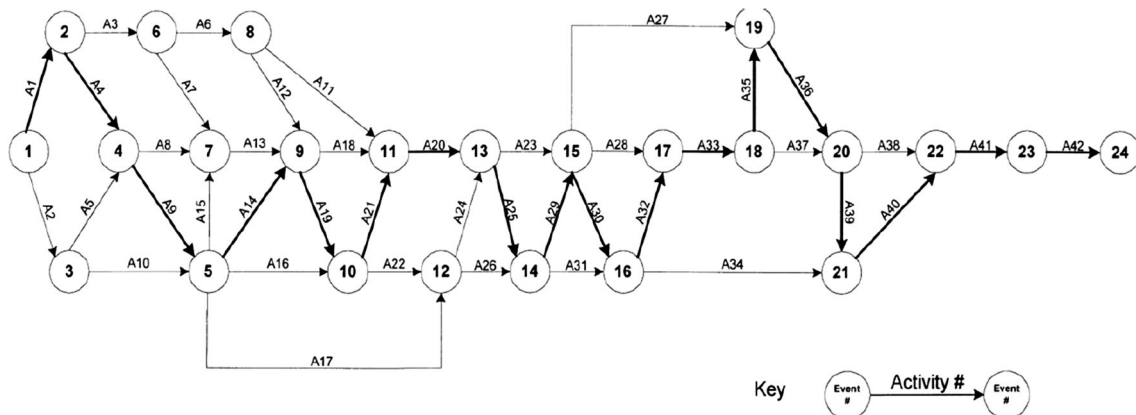


Fig. 9 AOA network of presented example

$O(MN^2)$, where M is the number of objectives and N is the population size. The procedure on non-dominant sorting is shown in Fig. 4.

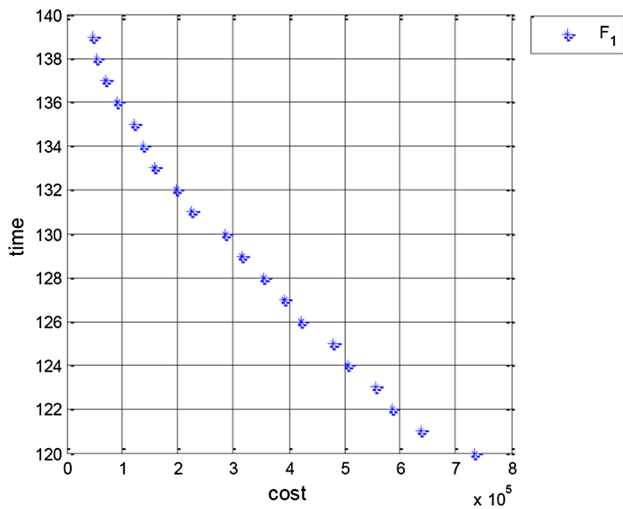


Fig. 10 Pareto front of the results for presented example

Table 3 Result of the first and second objective functions for 50 runs of NSGA-II and the values of compressing and delaying on activities

No.	Time	Cost	Crash	Delay	No.	Time	Cost	Crash	Delay
1	120	735,183	65	20	26	129	315,744	50	24
2	120	735,183	65	20	27	130	284,427	48	24
3	121	637,017	61	22	28	130	284,427	48	24
4	122	585,702	59	24	29	130	284,427	48	24
5	123	555,798	59	23	30	131	225,226	48	25
6	124	504,895	54	23	31	131	225,226	48	25
7	124	504,895	54	23	32	132	197,302	46	24
8	124	504,895	54	23	33	132	197,302	46	24
9	124	504,895	54	23	34	132	197,302	46	24
10	125	479,169	53	23	35	132	197,302	46	24
11	126	422,887	52	23	36	133	158,326	46	26
12	126	422,887	52	23	37	133	158,326	46	26
13	126	422,887	52	23	38	133	158,326	46	26
14	126	422,887	52	23	39	133	158,326	46	26
15	127	392,340	50	23	40	134	138,559	44	27
16	127	392,340	50	23	41	135	121,757	46	28
17	127	392,340	50	23	42	136	92,339	44	27
18	127	392,340	50	23	43	136	92,339	44	27
19	128	355,261	50	25	44	136	92,339	44	27
20	128	355,261	50	25	45	136	92,339	44	27
21	128	355,261	50	25	46	137	71,646	42	28
22	128	355,261	50	25	47	137	71,646	42	28
23	129	315,744	50	24	48	137	71,646	42	28
24	129	315,744	50	24	49	138	53,281	42	29
25	129	315,744	50	24	50	139	47,021	40	28

For determining the density of the solutions around a specific solution, the average distance between the specific solutions and two adjacent solutions is calculated and considered as a crowding distance. In other words, if we draw a rectangle that two adjacent solutions located on its vertex, sum of its length and width would be the crowding distance of the specific solution, as shown in Fig. 5. A specific solution with less crowding distance means less solution density around the specific solution, so for selecting the solutions for the next generation, the more crowding distance is better than less crowding distance (Deb 2001).

Parents

After non-dominant sorting and calculating crowding distance of each solution, the parents are ready for crossover and mutation operators based on selecting strategy.

Selecting strategy Parents are selected for crossover and mutation operators based on crowded tournament selection operator. In this operator, two solutions are compared with

each other, and winner is selected. The i th solution has two properties:

1. Has a rank or degree on non-domination that shows by r_i ,
2. Has a crowding distance that shows by d_i ,

The i th solution is the winner of the competition comparing with j th solution if and only if, one of two below conditions is established (Deb 2001):

1. The i th solution has better rank in a non-dominant sorting procedure ($r_i < r_j$) that means this solution has

Table 4 Duration of activities in solutions 1 to 25

Solution	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
A2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
A3	4	4	6	4	5	4	4	6	6	6	5	4	4	4	6	4	5	6	5	6	4	5	4	5	6
A4	13	13	13	13	13	13	13	13	13	13	14	13	13	13	13	13	14	13	14	13	13	14	13	14	13
A5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A6	11	11	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
A7	8	8	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A8	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
A9	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
A10	8	8	8	8	8	8	8	8	8	8	8	9	8	8	8	8	8	8	8	8	8	8	8	8	8
A11	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A12	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
A13	4	4	6	4	4	4	4	6	6	6	4	4	4	4	6	4	4	6	4	6	4	4	4	4	6
A14	4	4	5	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
A15	12	12	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
A16	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A17	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A18	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A19	11	11	13	11	11	11	13	14	13	14	11	13	13	11	13	11	13	13	13	13	13	11	11	13	13
A20	5	5	7	5	7	5	5	8	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
A21	8	8	9	8	9	9	8	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A22	5	5	9	5	9	5	5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A23	9	9	10	9	9	9	9	9	9	10	10	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A24	7	7	7	7	7	9	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
A25	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
A26	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A27	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A28	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
A29	6	6	8	6	6	6	6	6	8	8	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
A30	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
A31	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A32	15	15	20	15	15	15	15	20	20	20	20	20	15	15	15	20	20	20	15	15	20	20	15	15	20
A33	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
A34	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
A35	5	5	8	5	5	5	5	5	5	8	5	5	5	5	5	5	5	5	5	5	5	5	5	5	8
A36	9	9	10	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A37	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A38	7	7	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A39	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
A40	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
A41	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
A42	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

better non-domination degree comparing with j th solution,

2. If i th solution has more crowding distance comparing with j th solution ($d_i > d_j$), when the rank of both solutions are equal.

Crossover operator

For crossover operator, two parents were randomly selected, and two offspring produced using a uniform crossover operator. In this operator, for each genome of the parent's

Table 5 Duration of activities in solutions 26–50

Solution	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
A1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
A2	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
A3	4	5	4	6	6	6	6	6	4	5	4	5	5	4	6	6	6	4	4	5	6	4	5	6	6
A4	13	14	13	13	13	13	13	13	13	14	13	14	14	13	13	13	13	13	13	14	13	13	14	13	13
A5	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A6	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
A7	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A8	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
A9	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
A10	9	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A11	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A12	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
A13	4	4	4	6	6	6	6	6	4	4	4	4	4	4	6	6	6	4	4	4	6	4	4	6	6
A14	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
A15	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
A16	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A17	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A18	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A19	13	13	11	13	13	13	13	13	11	11	13	13	13	11	13	13	13	11	13	13	13	11	13	13	13
A20	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
A21	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A22	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A23	9	9	9	9	10	9	10	9	9	9	9	9	9	9	9	9	10	9	9	9	9	9	9	9	9
A24	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
A25	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
A26	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A27	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A28	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
A29	6	6	6	6	8	6	8	6	6	6	6	6	6	6	6	6	8	6	6	6	6	6	6	6	6
A30	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
A31	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A32	20	20	15	20	20	20	20	15	20	20	15	15	20	15	20	20	20	15	15	15	15	20	20	20	20
A33	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
A34	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
A35	5	5	5	8	8	5	8	5	5	5	5	5	5	5	8	5	8	5	5	5	5	5	5	5	8
A36	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
A37	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
A38	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
A39	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
A40	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
A41	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
A42	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

chromosome, a binary random variable is produced. If this variable is 1, the genome of the parents is changed with each other, and if this variable is 0, the genome stays in its place (Chau and Chan 1997; Deb et al. 2000). The crossover operator is shown in Fig. 6.

Mutation operator

The mutation operator is done in all four matrices of solution chromosome. For each genome, a uniform random variable between 0 and 1 is produced. If the value of this random variable is less than mutation rate, the genome mutates randomly, and if the value of the random variable is greater than mutation rate, the genome does not change (Chau and Chan 1997). The mutation operator is shown in Fig. 7.

Offsprings evaluation and combining with parents

In this section, we evaluate the offsprings that are created with crossover and mutation operators and assign a fitness quantity to each offspring. Then, we combine the parents and offsprings, and create a new population. This combination keeps the better solutions in new population. In multi-objective optimization problems, elitism has ambiguity. In these cases, we use a non-domination rank for each solution, so that each solution is rated based on non-domination.

After combination of the parents and offsprings, each solution is ranked based on the fast non-dominant sorting and crowding distances.

The mechanism of NSGA-II evolution is shown in Fig. 8.

Stop condition

The last step of NSGA-II is checking the stop condition. In multi-objective metaheuristic algorithms, there is no standard stop condition, so we consider a predefined algorithm iteration.

Numerical example

For illustrating the steps of presented algorithm, we used the numerical example of the paper entitled “Crashing PERT network using mathematical Programming” published in “International Journal of Project Management” in 2001. This example has been used in many studies as an authentic example for time compressing. In this example, all activities are assumed to be done in normal or compressed time. We extend this example for considering the delay in project activities. The values of example parameters are presented in Table 2.

The other parameters are $K_n = 100,000$, $H = 2000$, $I_0 = 0.1$, $C_{Max} = 1000,000$, and $t_{Max} = 140$.

The AOA network of presented example is shown in Fig. 9.

The results of solving the example with NSGA-II are presented as a Pareto front in Fig. 10.

As it is shown in Fig. 10, for faster finishing of projects, we need to pay more money, and in this situation, we must compress the activities more than normal situations. On the other hand, if we want to finish the projects in maximum acceptable due time, we have more money saving. The result of the first and second objective functions for 50 runs of NSGA-II and the values of compressing and delaying on activities are presented in Table 3.

As it is shown in Table 3, only 20 results are unique in 50 obtained ones, and the other results are repetitive. In addition, from the result no. 1 to no. 50, the time of finishing project increases, the project cost increases, and the compression and delaying activities show increasing trends. The duration of activities in each solution presented in Tables 4 and 5 consequently.

Conclusion and further studies

In this study, we showed that adding some assumptions to DTCTP can draw the problem nearer to real-world situations. One of these assumptions is adding the time value of money, because in many projects scheduling, the time value of money has a very important effect on making decision about compressing of the activities. In addition, adding the ability of delaying on project activities is another important factor appended to time–cost tradeoff problems. Moreover, presenting a Pareto front of results to decision makers gives them the opportunity to select the better solution due to project limitations and make the decision making procedure more flexible.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Akkan C (1998) A Lagrangian heuristic for the discrete time–cost tradeoff problem for activity-on-arc project networks. Working Paper, Koc University, Istanbul
- Ameen (1987) A computer assisted pert simulation. J Syst Manage
- Amiri Maghsoud, Khajeh Mostafa (2016) Developing a bi-objective optimization model for solving the availability allocation

- problem in repairable series–parallel systems by NSGA II. *J Indus Eng Int* 12(1):61–69
- Ann T, Erenguc SS (1998) The resource constrained project scheduling problem with multiple crashable modes: a heuristic procedure. *Eur J Operat Res* 107(2):250–259
- Burns SA (1994) The LP/IP hybrid method for construction time–cost trade off analysis. *Construct Manage Econ J*
- Chau DKH, Chan WT, Govindan K (1997) A time-cost trade-off model with resource consideration using genetic algorithm. *Civil Eng Syst* 14:291–311
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester
- Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Proceedings of the parallel problem solving from nature VI (PPSN-VI) conference, pp 849–858
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Deineko VG, Woeginger GJ (2001) Hardness of approximation of the discrete time–cost tradeoff problem. *Operat Res Lett* 29(5):207–210
- Demeulemeester E, Herroelen W, Elmaghraby SE (1996) Optimal procedures for the discrete time/cost trade-off problem in project networks. *Eur J Operat Res* 88(1):50–68
- Demeulemeester E, De Reyck B, Foubert B et al (1998) New computational results on the discrete time/cost trade-off problem in project networks. *J Operat Res Soc* 49(6):1153–1163
- Elmaghraby SE, Kamburowski J (1992) The analysis of activity network under generalized precedence relations. *Manage Sci* 38(9):1245–1263
- Elmaghraby SE, Salem A (1981) Optimal linear approximation in project compression. OR Technical Report 171, North Carolina State University at Raleigh
- Erenguc SS, Ahn T, Conway DG (2001) The resource constrained project scheduling problem with multiple crashable modes: an exact solution method. *Naval Res Log* 48(2):107–127
- Ford LR, Fulkerson DR (1962) Flows in networks. Princeton University Press, Princeton
- Fulkerson DR (1961) A network flow computation for project cost curves. *Manage Sci* 7:167–178
- Gen M, Cheng R (1997) Genetic algorithms and engineering design. Wiley, New York
- Goyal SK (1975) A note on the paper: a simple CPM time/cost trade-off algorithm. *Manage Sci* 21:718–722
- Hazir O, Haouari M, Erel E (2014) Robust optimization for the discrete time-cost tradeoff problem with cost uncertainty. *Handbook on project management and scheduling, vol 2, Part of the series International Handbooks on Information Systems* pp 865–874
- Hindelang TJ, Muth JF (1979) A dynamic programming algorithm for decision CPM networks. *Operat Res* 27(2):225–241
- Kaveh A, Mahdavi VR (2015) Resource allocation and time-cost trade-off using colliding bodies optimization. *Colliding Bodies optimization*, pp 261–277
- Ke H, Ma W, Chen X (2012) Modeling stochastic project time–cost trade-offs with time-dependent activity durations. *Appl Math Comput* 218:9462–9469
- Kelley JE (1961) Critical path planning and scheduling: mathematical basis. *Oper Res* 9:296–320
- Kelley JE, Walker MR (1959) Critical path planning and scheduling: an introduction. Mauchly Associates, Ambler
- Leu SS (2000) A GA-based fuzzy optimal model for construction time- cost trade off. *Int J Project Manage*
- Li H, Cao JN (1999) Using machine learning and GA to solve time-cost trade-off problems. *J Project Manage*
- Liu SX, Wang MG, Tang LX et al (2000) Genetic algorithm for the discrete time/cost trade-off problem in project network. *J North-east Univ China* 21(3):257–259
- Moder JJ, Phillips CR, Davis EW (1983) Project management with CPM, PERT and precedence diagramming, 3rd edn. Van Nostrand Reinhold Company, New York
- Pasandideh SHR, Niaki STA, Hajjipour V (2011) A multi-objective facility location model with batch arrivals: two parameter-tuned meta-heuristic algorithms. *J Intell Manuf.* doi:10.1007/s10845-011-0592-7
- Patterson JH, Harvey RT (1979) An implicit enumeration algorithm for the time/cost tradeoff problem in project network analysis. *Found Control Eng* 4(2):107–117
- Prabuddha DE, Dunne EJ, Ghosh JB et al (1997) Complexity of the discrete time–cost tradeoff problem for project networks. *Operat Res* 45(2):302–306
- Sasaki M, Gen M (2003a) A method of fuzzy multi-objective nonlinear programming with GUB structure by hybrid genetic algorithm. *Int J Smart Eng Des* 5(4):281–288
- Sasaki M, Gen M (2003b) Fuzzy multiple objective optimal system design by hybrid genetic algorithm. *Appl Softw Comput* 2(3):189–196
- Shahriari M et al (2010) A new mathematical model for time cost trade-off problem with budget limitation based on time value of money. *Appl Math Sci* 4(63):3107–3119
- Siemens N (1971) A simple CPM time/cost trade-off algorithm. *Manage Sci* 17:B-354–B-363
- Son J, Hong T, Lee S (2013) A mixed (continuous + discrete) time-cost trade-off model considering four different relationships with lag time. *KSCE J Civil Eng* 17(2):281–291
- Tavakkoli-Moghaddam R, Safari J, Sassani F (2008) Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliab Eng Syst Safe* 93:550–556
- The report of the world commission on Dams (WCD) (2000) Dams and development. Earthscan publications L + d, London and sterling
- Tiwari S, Johari S (2015) Project scheduling by integration of time cost trade-off and constrained resource scheduling. *J Inst Eng (India) Ser* 96(1):37–46
- Van Slyke RM (1963) Monte Carlo methods and the PERT problem. *Operat Res.* 33:141–143
- Vanhoucke M (2014) Generalized Discrete Time-Cost Tradeoff Problems. *Handbook on project management and scheduling, vol. 1 Part of the series International Handbooks on Information Systems* pp 639–658
- Vanhoucke M, Demeulemeester E, Herroelen W (2002) Discrete time/cost trade-offs in project scheduling with time-switch constraints. *J Operat Res Soc* 53(7):741–751
- Zade AE, Sadegheih A, Lotfi M (2014) A modified NSGA-II solution for a new multi-objective hub maximal covering problem under uncertain shipments. *J Indus Eng Int* 10(4):185–197

