Taylor & Francis
Taylor & Francis Group

International Journal of Construction Management

ISSN: 1562-3599 (Print) 2331-2327 (Online) Journal homepage: https://www.tandfonline.com/loi/tjcm20

Optimizing multi-mode time-cost-quality trade-off of construction project using opposition multiple objective difference evolution

Duc-Long Luong, Duc-Hoc Tran & Phong Thanh Nguyen

To cite this article: Duc-Long Luong, Duc-Hoc Tran & Phong Thanh Nguyen (2021) Optimizing multi-mode time-cost-quality trade-off of construction project using opposition multiple objective difference evolution, International Journal of Construction Management, 21:3, 271-283, DOI: 10.1080/15623599.2018.1526630

To link to this article: https://doi.org/10.1080/15623599.2018.1526630

Published online: 28 Dec 2018.

Submit your article to this journal

Article views: 698

View related articles

View Crossmark data

Citing articles: 26 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tjcm20

INTERNATIONAL JOURNAL OF CONSTRUCTION MANAGEMENT
2021, VOL. 21, NO. 3, 271–283
https://doi.org/10.1080/15623599.2018.1526630

Optimizing multi-mode time-cost-quality trade-off of construction project using opposition multiple objective difference evolution

Duc-Long Luong[a], Duc-Hoc Tran[a] and Phong Thanh Nguyen[b]

[a]Department of Construction Engineering and Management, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam;
[b]Department of Project Management, Faculty of Civil Engineering, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam

ABSTRACT

The time, cost and quality are crucial, conflicting aspects to construction project management. Tradeoff optimization among project duration (time), project cost, and the project quality within the project scope is necessary to enhance overall construction project benefit. A novel optimization algorithm, Opposition-based Multiple Objective Differential Evolution (OMODE), is presented to solve the time-cost-quality tradeoff (TCQT) problem. This novel algorithm employs an opposition-based learning technique for population initialization and for generation jumping. Opposition numbers are used to improve the exploration and convergence performance of the optimization process. A numerical high-way construction project case study is analyzed to illustrate the use of the algorithm and demonstrate its capabilities in generating non-dominated solutions. Comparisons with non-dominated sorting genetic algorithm (NSGA-II), multiple objective particle swarm optimization (MOPSO), multiple objective differential evolution (MODE) and previous results verify the efficiency and effectiveness of the proposed algorithm. This study is expected to provide an alternative solving methodology for the TCQT problem and help project manager plan construction methods with optimal time-cost-quality tradeoff.

KEYWORDS

Construction management; time-cost-quality tradeoff; multi-objective analysis; opposition; differential evolution

Introduction

In today’s competitive business environment, companies cannot survive without efficiently planning and scheduling. Resource management is a crucial part of project planning of any company. Time and Cost are two essential factors in construction project planning (Zahraie and Tavakolan 2009). In general, less expensive resources or technologies would result in longer project duration. For instance, using productive resources or technologies may save time, but this causes an increase in the cost. On the other hand, reduction of either time or cost may decrease quality of construction projects. Consequently, the relevant construction projects such as highways, tunnels, and bridges may age or deteriorate faster than expected, increasing the maintenance and rehabilitation cost (Zhang and Xing 2010). A time, cost and quality trade-off problem (TCQTP) is an extension of the time cost trade-off problem. This problem assumes that all project activities can be accomplished in the different modes of cost, time, and quality. The overall goal in TCQTPs is to select each activity’s mode such that the project can meet the deadline with the minimum possible cost and the maximum achievable quality. TCQTPs are NP-hard problems because the time, cost, and quality objectives are in conflict with each other (Khalili-Damghani et al. 2015).

Construction projects are built with a complex technology, open-air work environment, tight constraints of time and cost, and thus the accidents due to poor quality and safety occurred frequently (Fu and Zhang 2014). It is considerably significant to ensure project quality and safety under the construction time, cost and other constraints. Understanding the ability to minimize time, cost and improve quality of a project could determine the level of success of a construction company. Researchers have established various models and applied many different methods to reach time-cost-quality tradeoff from previous research works of the literature. Methodologies applied to optimize TCTQP are classified into mathematical methods (Babu and Suresh 1996; Khang and Myint 1999) and metaheuristic models (Kim et al. 2012; Tareghian and Taheri 2007). Heuristics are approximate rules-of-thumb, developed using problem-specific information, and tend to easily get trapped at local optima (Tareghian and Taheri 2006;

CONTACT Duc-Hoc Tran ✉ tdhoc@hcmut.edu.vn

© 2018 Informa UK Limited, trading as Taylor & Francis Group

Wood 2017). Evolutionary algorithms applied to TCQTP also including genetic algorithms (El-Rayes and Kandil 2005) ant colony ant algorithm (Das and Suganthan 2011) and artificial bee colony (Tran et al. 2015). Some evolutionary and other algorithms are configured to also evaluate TCQTP, considering limited availability of resources (Khalili-Damghani et al. 2015), or including more objectives (Tran et al. 2017).

Differential Evolution (DE) is currently one of the most popular evolutionary algorithms. DE may be used in a wide variety of highly nonlinear and complex optimization problems. This algorithm is simply structured and easy to use, while demonstrating great robustness and fast convergence in solving single objective global optimization problems (Price et al. 2005; Storn and Price 1997; Zheng et al. 2013). In light of these advantages, several researchers have applied DE to solve multi-objective problems (Das and Suganthan 2011; Zhou et al. 2011), with superior performance to other multiple objective algorithms widely reported and verified (Ali et al. 2012; Reddy and Kumar 2007; Wu et al. 2010). Therefore, this paper applies the DE algorithm in a model designed to solve the TCQT problem.

Based on the above, this study developed the novel Opposition Multiple Objective Differential Evolution (OMODE) algorithm to facilitate a TCQT analysis. In line with classical DE, this proposed algorithm applies an opposition-based learning technique for both population initialization and generation jumping. Opposition numbers are used to improve the exploration and convergence capabilities of optimization process. The objective is to demonstrate that the proposed algorithm attains fast convergence without losing solution diversity on the Pareto front. In the remaining parts of this paper, the related works and formulation on time-cost-quality tradeoff problem are presented. In the next section, the detailed descriptions of the proposed optimization model for the TCQT problem are presented in details. Subsequently, the performance of the newly developed model is demonstrated using a numerical case study and result comparisons. The final section presents conclusions and suggestions for future work.

Related works on project time-cost-quality problem

Many methodologies have been proposed to solve the time-cost tradeoff (TCT) problem by deciding an optimal combination of construction methods for all the activities (Ogunsemi and Jagboro 2006; Wood 2017; Zhang and Li 2010). Analytical methods use mathematical approaches such as linear and dynamic programming to solve problems (Burns et al. 1996; De et al. 1995; Jiang and Zhu 2010). The meta-heuristic approaches have shown relatively higher efficiency in this domain. Although such solutions do not necessarily guarantee the global optimal solution, their ability to search a solution space intelligently rather than completely makes them capable of producing relatively good solutions to large-scale problems (Kalhor et al. 2011). Li et al. (1999) proposed the application of genetic algorithms (GAs) and Pareto front approach to solve TCT problems. Yang (2007) developed a new algorithm using particle swarm optimization (PSO) to deal with time cost tradeoffs. Ng and Zhang (2008), Afshar et al. (2009) presented a Pareto-based multi-objective ant colony optimization (ACO) for TCT problems. Bettemir and Talat Birgönül (2017) based network analysis algorithm for the solution of discrete time-cost Trade-off Problem. Nabipoor et al. (2013) used an adjusted fuzzy dominance genetic algorithm to deal with multi-mode resource-constrained discrete time–cost tradeoff problem.

Trade-off during the project planning is not limited to time and cost. Therefore, variants of time-cost trade-off problem are also analyzed. Conducting sufficient analysis and multi-objective optimization of projects' time-cost-quality tradeoffs are therefore an essential requirement in the effective planning of all construction projects of any size. Babu and Suresh (1996) proposed a framework to study the tradeoff among time, cost, and quality for the first time using three inter-related linear programming models. Then Khang and Myint (1999) applied Babu's method in a cement factory construction project and verified the validity of the method. Attempting to use multiple objective evolutionary algorithms, El-Rayes and Kandil (2005) presented a multi-objective model to search for an optimal resource utilization plan that minimizes construction cost and time while maximizing its quality jointly, applied genetic algorithm (GA) to provide the capability of quantifying and considering quality, and visualized optimal tradeoffs among construction time, cost, and quality by a highway construction example. Zhang and Xing (2010) indicated that time, cost, and especially quality were difficult to describe by precise numbers; therefore, they proposed a Pareto-based multi-objective Particle Swarm Optimization (PSO) for the fuzzy TCQT problem. Tran et al. (2015) successfully used the hybridization evolutionary algorithms to solve the time-cost-quality tradeoff problem. Khalili-Damghani et al. (2015) modeled a multi-mode time–cost–quality trade-off project scheduling problem under generalized precedence relations using mixed-integer mathematical programming. Kazaz et al. (2016) outline a new two-step methodology for TCQ trade-off problems. Wood (2017) analyzed stochastic duration-cost-quality tradeoff problems with fuzzy memetic optimization algorithm.

Problem formulation

The objectives faced by a project manager are to minimize the overall time and cost and maximize the overall quality of the project. The problem is then to select an optimal combination of construction methods for all activities in order to arrive at an optimal compromise among time, cost and quality for the project. Three major objective functions are formularized in the following to enable the evaluation of project performance in construction time, cost and quality, respectively (El-Rayes and Kandil 2005).

The first objective, minimization of total project duration, may be expressed as follows:

$$\begin{gathered} Minimize\ project\ time = \sum_{i=1}^{l} T_i = Max_{\forall i}(ES_i + d_i) \\ ES_i = \underset{\text{all predecessors } j \text{ of } i}{Maximum} \left(ES_j + d_j\right) \end{gathered} \tag{1}$$

where T_i is duration of the activity $i\{i = 1, 2, \ldots, l\}$ on the critical path for a specific option of execution methods; l is the total number of critical activities on a specific critical path. ES_i is the earliest start of activity i, d_i is the duration of activity i. In this study, project duration is estimated based on the procedure used by El-Rayes and Moselhi (2001). In general, project duration is calculated based on precedence constraints and activity duration. The project information determines the precedence constraints and the selection alternatives determine activity duration.

The second objective, minimization of total project cost, may be calculated as follows:

$$Minimize\ project \cos t = \sum_{i=1}^{n} \cos t_i \tag{2}$$

where $\cos t_i$ is the cost of activity i for a specific option of execution methods and n is the total number of activities.

The third, which is also the last objective to be considered, is the maximization of project's overall quality. The project quality value is equal to the sum of all activities' quality. Therefore, the value of this objective is expressed as following equation:

$$Maximiz\ eproject\ quality = \sum_{i=1}^{l} wt_i \sum_{k=1}^{K} wt_{i,k} x Q_{i,k}^{n} \tag{3}$$

where $Q_{i,k}^{n}$ is performance of quality indicator (k) in activity (i) using resource utilization (n); $wt_{i,k}$ is weight of quality indicator (k) compared to other indicators in activity (i); and $wt_i$is weight of activity (i) compared to other activities in the project. Normally, in an activity has several indicators such as material, equipment, labor, etc… that effect its quality. Each indicator has a weight which impact to activity quality. For example, an activity has three indicators with quality performance as 90; 80; 95, respectively. The corresponding weights of quality indicator are 50%; 30%; 20%. The weight of activity in total project is 0.03 (3%). Therefore, the quality of activity is calculated as [(90*50%+80*30%+95*20)]*0.03 = 2.64.

Differential evolution

Differential evolution (DE) is a simple population-based, direct-search method used to solve global optimization problems (Price et al. 2005; Storn and Price 1997). DE procedures are similar to the genetic algorithm (GA), as both use mutation, crossover, and selection processes. However, the nature of the mutation process used in the DE distinguishes this method from the GA. The DE mutation process selects three random solutions from the previous generation to create a mutant solution. The original DE algorithm is described briefly as follows:

Let $S \subset \Re^n$ be the search space of the problem under consideration. DE utilizes NP and D-dimensional parameter vectors $X_{i,G} = \{x_{i,G}^{1}, x_{i,G}^{2}, \ldots, x_{i,G}^{D}\}$, $i = 1, 2, \ldots, NP$ as a population for each algorithm generation. The initial population is generated randomly and should cover the entire parameter space. At each generation, DE applies two operators, namely mutation and crossover (recombination) to yield one trial vector $U_{i,G+1}$ for each target vector $X_{i,G}$. Then, a selection phase takes place to determine whether the trial vector enters the population of the next generation or not. For each target vector $X_{i,G}$, a mutant vector $V_{i,G+1}$ is determined using the following equation.

$$V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \tag{4}$$

where, $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$ are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$, and F is a scaling factor such that $F \in [0, 1]$.

Following the mutation phase, the crossover operator is applied to increase the diversity. For each mutant vector $V_{i,G+1}$, a trial vector $U_{i,G+1} = \{u^1_{i,G+1}, u^2_{i,G+1}, \ldots, u^D_{i,G+1}\}$ is generated using the following scheme.

$$u^j_{i,G+1} = \begin{cases} v^j_{i,G+1}\, if (rand_j[0,1) \leq CR\, or\, j = j_{rand} \\ x^j_{i,G}\, otherwise \end{cases} j = 1, 2, \ldots, D \tag{5}$$

$CR \subset [0,1]$ is a user-defined crossover constant; j_{rand} is a randomly chosen index from $\{1, 2, \ldots, D\}$ that ensures trail vector $U_{i,G+1}$ differs from its target $X_{i,G}$ by at least one parameter.

To decide whether the trial vector $U_{i,G+1}$ should be a member of the population in the next generation, it is compared to the corresponding target vector $X_{i,G}$ using the greedy criterion. The selection operator is expressed as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G+1}\, iff(U_{i,G+1}) < f(X_{i,G}) \\ X_{i,G} \quad otherwise \end{cases} \tag{6}$$

Once the memberships of the next generation are selected, the evolutionary cycle of the DE iterates until a stopping condition is satisfied.

Opposition-based learning

Opposition-based strategy in the optimization algorithms uses the concept of opposition-based learning (OBL), introduced by Tizhoosh (2005). The main idea behind OBL is the simultaneous consideration of an estimate and its corresponding opposition estimate (i.e., guess and opposition guess) in order to achieve a better approximation of the current candidate solution. In the absence of *a priori* information about the solution, we usually start with *random guesses*. The computation time, among others, is related to the distance of these initial guesses from the optimal solution. We may improve our chance of starting with a closer (fitter) solution by simultaneously checking the opposite solution. By doing this the fitter one (guess or opposition guess) may be chosen as an initial solution. In fact, according to probability theory, a guess will be further from the solution than its opposite guess 50% of the time. Therefore, starting with the closer of the two guesses (as judged by its fitness) has the potential to accelerate convergence. The same approach can be applied not only to initial solutions, but also continuously to each solution in the current population (Ahandani and Alavi-Rad 2012; Rahnamayan et al. 2008).

The proposed opposition based multiple objective differential evolution for time-cost-quality tradeoff model (OMODE-TCQ)

This section describes the opposition-based multiple objective differential evolution (OMODE) for solving the TCQT problem developed in this study based on the original DE algorithm (Storn and , Price, 1997; Price et al., 2005). In OMODE, potential candidate solutions are addressed at initialization and the exploration and exploitation capabilities during various optimization processes are mainly concerned. OBL may be used in two stages of MODE. Firstly, in the initialization stage in order to achieve fitter staring candidate solutions under conditions in which a priori knowledge about initial members does not exist; Secondly, during MODE implementation in order to force the current population to jump into new candidate solutions that are fitter than current ones. These two stages are named as opposition-based population initialization and opposition-based generation jumping, respectively (Ahandani et al. 2010; Ahandani and Alavi-Rad 2012; Rahnamayan et al. 2008). In this way the proposed algorithm can converge faster while maintaining good diversity. Figure 1 shows the overall operational architecture of the proposed algorithm.

Opposition-based population initialization

This study considers the TCQT problem, in which project cost, project duration, and project quality are optimized simultaneously. The model requires project information inputs including activity relationship, activity duration, and construction methods for each activity. In addition, the user also must provide parameter settings for the search engine (OMODE) such

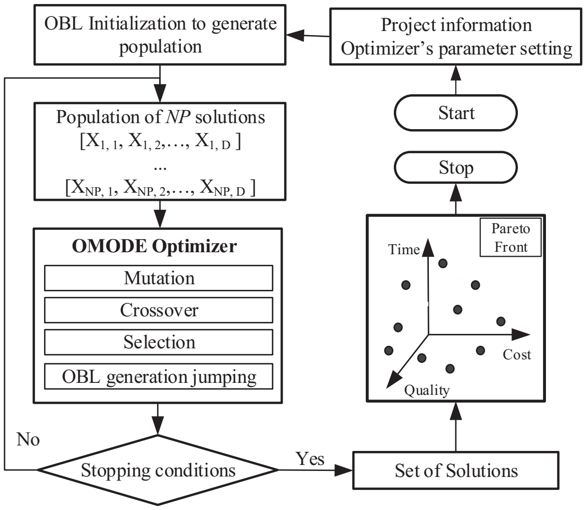

Figure 1. OMODE flowchart for the TCQT problem.

as the value of population size *NP*, number of decision variables *D*, number of objective functions *M*, value of the mutant constant *F*, value of the crossover probability constant *CR*, maximum number of generations G_{max}, the lower bound (*LB*) of decision variables, and the upper bound (*UB*) of decision variables. With these inputs, the optimizer conducts calculations to obtain an optimal set of construction methods for all construction project activities. With all the necessary information provided, the model is capable of operating automatically without any human intervention.

Population initialization is the first and the primary task in any evolutionary algorithm. The population in the DE may be guided toward more promising areas if the initial population can be spread as much as possible over the objective function surface. Hence, incorporating OBL into DE is worth investigating. In OMODE, a combined initial population of 2NP solutions is generated using uniform random distribution and the OBL technique.

The above process generates a point in D-dimensional space $X = \{x_1, x_2, \ldots, x_D\}$ in which $x_1, x_2, \ldots, x_D \in \Re$ and $x_j \in [0, 1]$ have uniform random distributions and its corresponding opposite point $X^o = \{x_1^o, x_2^o, \ldots, x_D^o\}$ is created as $x_j^o = 1 - x_j$. The first NP individuals may be easily generated as follows:

$$X_{i,j}^{G=0} = LB_j + x_{i,j} * (UB_j - LB_j) \tag{7}$$

An opposition-based learning approach then generates other NP solutions according to the following equation.

$$X_{i,j}^o = LB_j + x_{i,j}^o * (UB_j - LB_j)\ (j = 1, \ldots, D; i = 1, \ldots, NP) \tag{8}$$

A candidate solution to the TCQT problems may be represented as a vector with D elements as follows:

$$X = [X_{i,1}, X_{i,2}, \ldots, X_{i,j}, \ldots, X_{i,D}] \tag{9}$$

where D is the number of decision variables in the problem at hand. It is obvious that D is also the number of activities in the project network. Index *i* denotes the i^{th} individual in the population. Vector $X_{i,j}$ represents one construction method for activity j. Construction method $X_{i,j}$ is an integer number in the range [1,M_j] ($j = 1$ to D), meaning one position from M_j construction methods. Because the original DE operates with real-value variables, a function is employed to convert the construction method options of those activities from real values to integer values within the feasible domain.

$$X_{i,j} = Ceil\big(rand[0, 1] \times UB(j)\big) \tag{10}$$

where $X_{i,j}$ is the option of activity j at the individual i^{th}. $rand[0, 1]$ denotes a number between 0 and 1 generated by uniformly distributed random and opposition learning techniques. UB(j) = M_j represents the number of construction methods for each activity. Ceil is a function to round a real number to the nearest integer greater than or equal to it.

Population selection procedure

During the optimization process, size of population remains NP (Storn and Price 1997). NP best (elite) solutions are selected from the combined population. While the "highest fitness value" solution is the best solution in the single objective solution scenario, a two-solution dominance approach is used in multi-objective scenarios. Thus, the NP solutions in this research are selected using fast non-dominated sorting (Deb et al. 2002) and the entropy crowding technique (Wang et al. 2010). Figure 2 provides an overview of this procedure.

The solutions belonging to the best non-dominated set (Set F_1) are selected first. If size of F_1 is smaller than *NP*, the remaining members of the population are chosen from subsequent non-dominated fronts in rank order (F_2, F_3 …). This procedure is continued until additional sets cannot be accommodated. Assume that F_k is the last non-dominated set able to be accommodated. In general, number of solutions in all sets F_1 through F_k will be greater than NP. In order to use crowding entropy sorting to choose the *NP* population members needed to select the best solutions, it is necessary to first fill all population slots in descending order of distance.

Mutation operation

Once initialized, DE mutates the population to produce a set of mutant vectors. A mutated vector $V_{i,G+1}$ is created using Eq. (4) that corresponds to the target vector $X_{i,G}$.

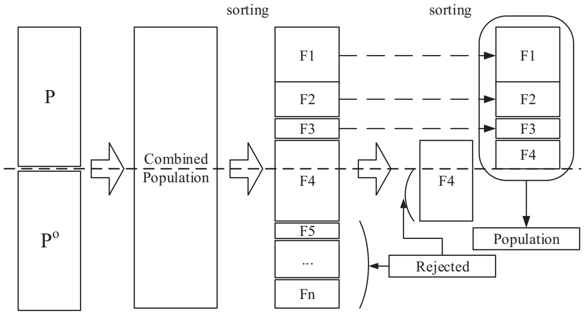

Figure 2. Population selection procedure.

Crossover operation

The crossover operation exchanges components of the target vector and the mutant vector to diversify the current population. In this stage, a new vector called the trial vector, is created using Eq. (5).

Selection operation

Modification of the selection mechanism is the most important task of multi-objective optimization because the careful selection of candidate solutions facilitates the generation of a good Pareto front.

This research utilized a selection mechanism proposed by Ali et al. (2012). This mechanism first evaluates the trial vector $U_{i,G+1}$ and then compares it with the target vector $X_{i,G}$. If the trial vector dominates the target solution, it replaces the target solution immediately in the current population and the target solution is added to another (advanced) solution. Otherwise, the new solution (trial solution) is added to the advanced population. The two populations (current and advanced) are combined after each generation. Note that the total size of the combined population is 2NP. However, population size during the optimization process remains NP. Thus, NP solutions are selected based on the same technique presented in the Population selection procedure section.

Opposition-based generation jumping

In this stage, if jumping condition $\mathrm{J_r}$ is satisfied $Rand() \leq -(\frac{g}{G_{\max}})^2 + 2(\frac{g}{G_{\max}})$ where g and $G_{\max}$ are the current and maximum generations, respectively, then the corresponding opposition population is computed and the N_{pop} fittest individuals are chosen from the combined current and opposition population after generating new populations through DE evolution. Unlike the process in the opposition-based initialization phase, generation jumping calculates the opposition population dynamically. Instead of using variables' predefined interval boundaries ($[\mathrm{LB_j}, \mathrm{UB_j}]$), generation jumping calculates the opposite of each variable based on minimum (Min_j^p) and maximum (Max_j^p) values for that variable in the current population.

$$X_{i,j}^{o,current} = Min_j^p + Max_j^p - x_{i,j}^{current} \qquad (j = 1, \ldots, D; i = 1, \ldots, NP) \tag{11}$$

By staying within variables' interval static boundaries, we jump outside of the already shrunken search space and lose the knowledge of the current reduced space (converged population). Hence, we calculate opposite points by using the current interval of variables in the population which is, as the search progresses, increasingly smaller than the corresponding initial range (Rahnamayan et al. 2008).

Stopping conditions

The optimization process terminates when the stopping conditions are met. The user can set the type of these conditions. Maximum generation G_{max} or maximum number of functions evaluations (NFE) may be used as the stopping criterion. When the optimization process terminates, the final set of optimal solutions, called the Pareto front, is presented to the user. Obtaining the entire Pareto front is of great importance because it assists planners to evaluate the pros and cons of each potential solution based on qualitative and experience-driven considerations.

Case study

This study analyzed a numerical case to demonstrate the effectiveness of the proposed OMODE for the TCQT problem, with obtained results compared against three approaches frequently employed to handle the TCQT problem, including NSGA-II, MOPSO, and MODE. The NSGA-II algorithm has been demonstrated to be one of the efficient algorithms for solving benchmark multi-objective optimization problems (Ramesh et al. 2012). The NSGA-II algorithm is successfully applied in various construction problems such as resource utilization (Zahraie and Tavakolan 2009), scheduling (Shahriari 2016; Zeng et al. 2018). The particle swarm optimization is an adaptive evolution algorithm based on population search, the thought comes to the prey behavior of birds, and its advantages is simple and fast convergence. The MOPSO is a very competitive swarm intelligence algorithm for multi-objective optimization problems including construction management (Aminbakhsh and Sonmez 2017; Elbeltagi et al. 2016). The MODE is a powerful and effective population based evolutionary algorithm for fast optimization. The MODE has been widely used in various fields including construction problems (Cheng and Tran 2014). The case project was adopted from previous study of an actual high-way construction project (El-Rayes and Kandil 2005). The case study is used for applying the proposed model that collected in planning phase. The project comprised 18 construction activities, each of which has a number of possible construction alternatives. For instance, in a construction phase of the highway project, there are many resource selection options for the soil excavation and compaction, concrete pavement, rebar installation, formwork installation activities, etc including: (1) different construction material; (2) crew formation with different number of workers and different machine; (3) working shift with different number of overtime hours. Each of selected option results in different time, cost, quality for an activity. Figure 3 shows the precedence relationships of the network and Table 1 shows the associated time, cost and quality for the options of activity. From this table, there is an average of 3.4 construction methods for each of 18 activities, which produces billions possible combination for delivering the entire project. Each possible combination has a unique impact on project performance, which means that project managers must search in a large number of potential solutions to find those that establish an optimal tradeoff/balance among construction duration, cost, and quality. This study used the developed newly developed multi-objective optimization model to search the many potential solutions.

Optimization result of OMODE-TCE

Table 2 shows parameter settings for the proposed OMODE-TCQT (Ali et al. 2012; Cheng and Tran 2015; Storn and Price 1997; Wu et al. 2010). Thirty independent optimization runs were conducted to avoid randomness. Table 3 lists the first 9 non-dominated solutions in descending order of time/duration, cost and quality, respectively, along with optimal construction method combinations. Project managers may select the optimal solution for a specific project scenario based on experience, preferences, and specific situation conditions. Figure 4 shows a typical Pareto optimal front obtained using OMODE. This front shows clearly the relationships among project duration, cost, and quality. This three-dimensional visualization of the tradeoffs may help decision makers evaluate the impact on project performance of the various potential resource-utilization plans.

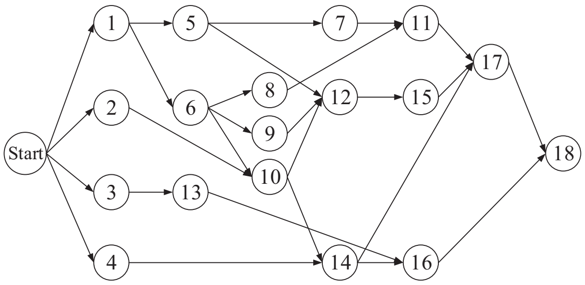

Figure 3. Network of project.

The non-dominated solutions may also be used to optimize tradeoffs between any two objectives on a two-dimensional plane. Figures 4–6 show the relationship between time and cost, cost and quality, and time and quality, respectively. As shown in the time-cost curve example (Figure 5), lower project funding correlates with longer project completion duration and vice versa. However, Figures 5–7 may not be good representatives of the entire tradeoff surface in the three-dimensional space. In fact, the two-dimensional tradeoff surface, when projected from three to two dimensions, may lose some non-dominated points because it is the hidden dimension that makes these points non-dominated.

Statistical comparison and analysis

We compared OMODE performance against NSGA-II (Deb et al. 2002), MOPSO (Yang 2007), and original MODE to assess comparative effectiveness. The latter three algorithms have all been previously applied to TCQT problem with impressive success. For comparison purposes, all four algorithms used an equal number of function evaluations, had population sizes of 100, and used a maximum of 500 generations. In NSGA-II, the constant mutant and crossover probability factors were set at 0.5 and 0.9, respectively. In MOPSO, the two learning factors c_1, c_2 were both chosen at 2, and the inertia factor w is set in range of 0.7 to 0.3. MODE and OMODE control parameters remained the same, as stated previously in Table 2. Thirty independent runs were carried out for all experiments.

Table 4 shows the performance results obtained by proposed algorithm and benchmarked algorithms. Observing from this table, all of five solutions found by OMODE model are superior to previous findings and benchmarked model. In terms of time, all models can obtain the best time, but OMODE yields the higher quality and lower cost (case 1). The proposed algorithm can acquire lowest cost and highest quality.

Multi-objective optimization problem performance measures are more complex than those of single-objective optimization problems. Three major issues are normally taken into consideration: (1) convergence to the Pareto optimal set; (2) maintenance of diversity in solutions of the Pareto optimal set and (3) the maximal distribution bound of the Pareto optimal set (Wu et al. 2010; Wang and Singh 2009). Numerous quality indicators have been suggested in the literature (Coello and Coello 2006; Deb et al. 2002; Fonseca and Fleming 1995). These indicators may be classified into four categories based on whether they evaluate: 1) closeness to the Pareto

Table 1. Options of resource utilization.

					QI k = 1		QI k = 2		QI k = 3		
Activity (i)	Resource option (n)	Duration (days)	Cost (dollars)	Act. weight (%) (wt_i)	IW ($wt_{i,k}$)	QP ($Q^n_{i,k}$)	IW ($wt_{i,k}$)	QP ($Q^n_{i,k}$)	IW ($wt_{i,k}$)	QP ($Q^n_{i,k}$)	Quality
1	1	14	2400	3	50	100	30	96	20	98	2.9520
	2	15	2150			90		89		89	2.6850
	3	16	1900			86		77		84	2.4870
	4	21	1500			75		72		73	2.2110
	5	24	1200			63		60		65	1.8750
2	1	15	3000	5	40	98	40	94	20	99	4.8300
	2	18	2400			87		94		95	4.5700
	3	20	1800			81		92		85	4.3100
	4	23	1500			77		72		70	3.6800
	5	25	1000			60		66		59	3.1100
3	1	15	4500	8	70	100	15	97	15	95	7.9040
	2	22	4000			80		82		81	6.4360
	3	33	3200			62		60		63	4.9480
4	1	12	45000	11	50	99	35	95	15	94	10.6535
	2	16	35000			74		71		76	8.0575
	3	20	30000			59		63		64	6.7265
5	1	22	20000	10	60	100	20	97	20	99	9.9200
	2	24	17500			93		89		89	9.1400
	3	28	15000			77		71		72	7.4800
	4	30	10000			61		64		61	6.1600
6	1	14	40000	11	50	95	25	95	25	100	10.5875
	2	18	32000			76		74		79	8.3875
	3	24	18000			59		62		68	6.8200
7	1	9	30000	10	30	97	30	99	40	93	9.6000
	2	15	24000			70		73		71	7.1300
	3	18	22000			61		62		67	6.3700
8	1	14	220	1	100	95	0	NA	0	NA	0.9500
	2	15	215			83					0.8300
	3	16	200			75					0.7500
	4	21	208			68					0.6800
	5	24	120			61					0.6100
9	1	15	300	1	50	100	50	99	0	NA	0.9950
	2	18	240			97		92			0.9450
	3	20	180			81		88			0.8450
	4	23	150			71		75			0.7300
	5	25	100			63		64			0.6350
10	1	15	450	1	60	94	40	97	0	NA	0.9520
	2	22	400			79		83			0.8060
	3	33	320			63		69			0.6540
11	1	12	450	2	70	96	30	95	0	NA	1.9140
	2	16	350			72		75			1.4580
	3	20	300			61		66			1.2500
12	1	22	2000	3	50	99	35	98	15	95	2.9415
	2	24	1750			89		85		87	2.6190
	3	28	1500			70		71		79	2.1510
	4	30	1000			62		61		63	1.8540
13	1	14	4000	7	40	99	40	96	20	97	6.8180
	2	18	3200			73		71		76	5.0960
	3	24	1800			60		62		63	4.2980
14	1	9	3000	6	80	100	10	95	10	98	5.9580
	2	15	2400			79		82		81	4.7700
	3	18	2200			63		67		66	3.8220
15	1	16	3500	7	70	100	30	98	0	NA	6.9580
16	1	20	3000	3	30	97	30	96	40	98	2.9130
	2	22	2000			89		85		87	2.6100
	3	24	1750			81		79		78	2.3760
	4	28	1500			72		73		74	2.1930
	5	30	1000			67		60		62	1.8870
17	1	14	4000	6	70	98	20	97	10	99	5.8740
	2	18	3200			73		75		72	4.3980
	3	24	1800			62		65		61	3.7500
18	1	9	3000	5	30	98	45	99	25	94	4.8725
	2	15	2400			75		77		71	3.7450
	3	18	2200			63		66		67	3.2675

Notation
QI: Quality indicator
IW: Indicator weight
QP: Quality performance
NA Not available

front; 2) the diversity in obtained solutions; 3) both 1 and 2. The following describes the three quality indicators used in this research to evaluate, respectively, each of the four issues:

1. C-metric (C): C-metric is often used to assess the quality of the true Pareto front of optimized problems (Zitzler and Thiele 1999). Let $S_1, S_2 \subseteq S$ be two sets of decision solutions. C-metric is defined as the mapping between the ordered pair (S_1, S_2) and the interval [0,1]:

$$C(S_1, S_2) = \frac{|\{a_2 \in S_2; \exists a_1 \in S_1 : a_1 \leq a_2\}|}{|S_2|} \qquad (12)$$

The numerator in Eq. (12) denotes the number of solutions in S_2 are dominated by at least one solution in S_1, the denominator is the total solutions in S_2. Provided that $C(S_1,S_2) = 1$, all solutions in S_2 are dominated by or equal to solutions in S_1. If $C(S_1,S_2) = 0$, then S_1 covers none of the solutions in S_2. Both $C(S_1,S_2)$ and $C(S_2,S_1)$ should be checked in the comparison because C-metric is not symmetrical in its arguments (Wang and Singh 2009). Table 5 illustrates comparison results among four algorithms in terms of C-metric, where A_1, A_2, A_3, and A_4 indicate OMODE, MODE, MOPSO, and NSGA-II, respectively. Results show that OMODE dominates more than 43% of MODE solutions, 75% of MOPSO solutions, and 73% of NSGA-II solutions on average.

2. Spread (SP): This indicator (Wang et al. 2010) measures the extent of spread achieved among the non-dominated solutions. The mathematical definition of SP may be given as:

Table 2. OMODE-TCQT parameter settings.

Input parameters	Notation	Setting
Number of decision variables	*D*	25
Population size	*NP*	100
Crossover probability	*CR*	0.5⊠0.9
Scaling factor	*F*	0.5
Maximum generation	G_{max}	500

$$SP = \frac{\sum_{i=1}^{k} d(E_i, \Omega) + \sum_{X \in \Omega} |d(X, \Omega) - \bar{d}|}{\sum_{i=1}^{k} d(E_i, \Omega) + (|\Omega| - k)\bar{d}} \qquad (13)$$

where Ω is a set of solutions, $(E_1, \ldots, E_k)$ are k extreme solutions in the set of true Pareto-front PF, k is the number of objectives and $d(X, \Omega) = \min_{Y \in \Omega, Y \neq X} ||F(X) - F(Y)||$ is the minimum Euclidean distance between solution X and its neighboring solutions in the obtained non-dominated Ω set; $\bar{d} = \frac{1}{|\Omega|} \sum_{X \in \Omega} d(X, \Omega)$ is the mean value of all $d(X, \Omega)$, $|\Omega|$ is the total solutions in Ω set. A value of zero for this metric indicates all members of the

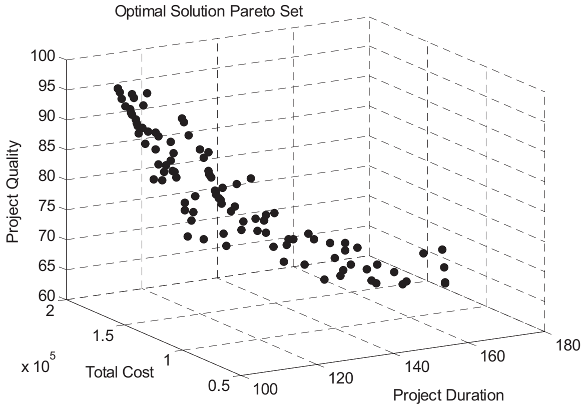

Figure 4. Time-Cost-Quality impact tradeoff Pareto front using OMODE.

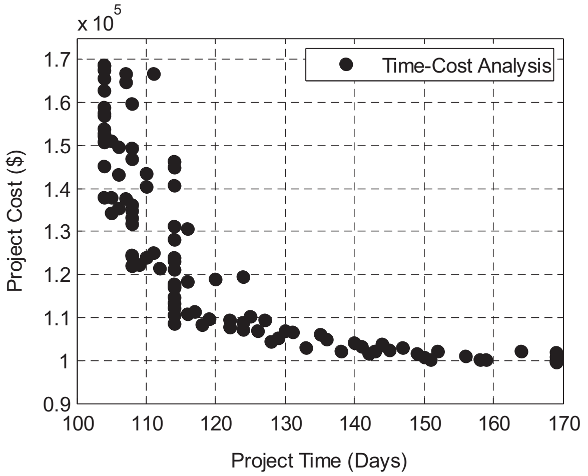

Figure 5. Time-Cost tradeoff analysis

Table 3. Best non-dominated solutions obtained by OMODE-TCQT.

			Project performance		
Solutions	Partial set	Optimal resource utilization options	Time (days)	Cost (dollars)	Quality (%)
1	Sorted by time	[1.2.1.1.2.1.1.2.1.1.1.1.1.1.1.1.2.1.1]	104	164715	96.17
2		[1.1.1.2.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1]	104	158820	95.03
3		[1.3.1.1.2.1.1.3.1.1.1.1.1.1.1.1.5.1.1]	104	163100	95.01
4	Sorted by tost	[5.5.3.3.4.3.3.5.4.2.3.4.3.3.1.5.3.3]	159	99870	65.24
5		[3.5.2.3.4.3.3.1.1.1.3.1.3.2.1.4.2.1]	120	105570	72.69
6		[1.5.3.3.4.3.3.5.1.1.3.1.3.3.1.5.1.1]	114	105270	71.55
7	Sorted by quality	[1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1]	104	168820	97.63
8		[1.1.1.1.1.1.1.2.3.1.1.1.1.1.1.1.2.1.1]	109	167695	97.06
9		[1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.2.1.1]	104	167820	97.33

Pareto optimal set are spaced equidistantly. A smaller value of SP indicates a better distribution and diversity of non-dominated solutions. Table 6 shows a comparison of the spread metric for different algorithms. This supports that the average performance of the OMODE is superior to that of the three other algorithms.

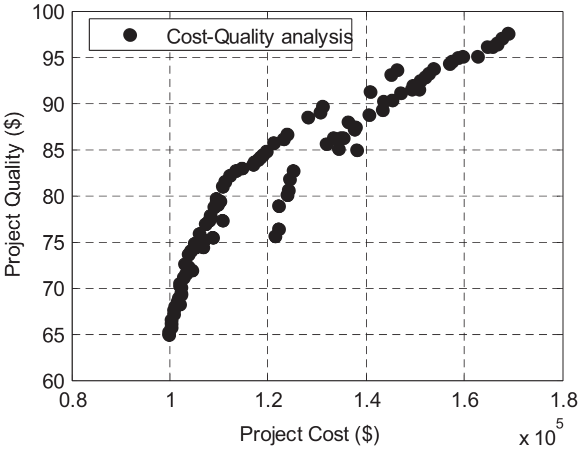

Figure 6. Cost-Quality impact tradeoff analysis.

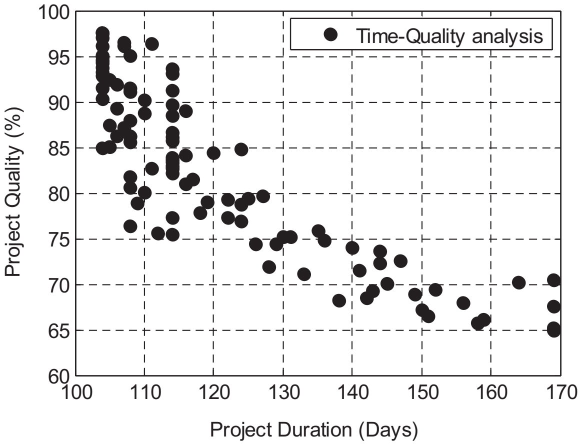

Figure 7. Time-Quality impact tradeoff analysis.

3. Hyper-volume (HV): This indicator calculates the volume (in the objective space) covered by members of a non-dominated set of solutions Ω for a problem that works to minimize all objectives (Wu et al. 2010; Zitzler et al. 2003). Mathematically, a hypercube v_i is constructed for each solution $X_i \in \Omega$ with reference point W and the solution $\mathrm{X_i}$ as the diagonal corners of the hypercube. The reference point can be found simply by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found and its HV is calculated as product of allv_i:

$$HV = \bigcup_{i=1}^{|\Omega|} v_i \tag{14}$$

Algorithms with larger HV values are desirable. The HV value of a set of solutions is normalized using a reference set of Pareto optimal solutions with the same reference point. After normalization, the HV values are confined to range [0,1]. Table 7 lists the results for each of the four compared algorithms in terms of HV. From Table 7, we see that the proposed model obtains the largest HV values, which means that OMODE has better convergence and diversity performance than the other three algorithms.

As shown in the statistical results, opposition-based learning operator helped the proposed algorithm in achieving the better results by having the diversified search space during optimization process. The computation time is related to the distance of these initial guesses from the optimal solution. The opposition-

Table 4. Result comparison between OMODE-TCQT and benchmarked algorithms.

			Project performance		
Model	Case	Optimal resource utilization options	Time (days)	Cost (dollars)	Quality (%)
Previous findings	1	[1.1.1.1.2.1.1.1.1.1.1.1.1.1.1.1.1.1]	104	166320	95.00
	2	[1.5.3.3.4.3.3.5.1.1.3.1.3.2.1.5.1.1]	114	105470	71.00
	3	[2.3.1.1.2.3.1.1.1.1.1.1.1.1.1.3.1.1]	115	141620	90.00
	4	[2.5.1.3.4.2.3.3.1.1.1.1.3.1.1.5.1.1]	109	121350	77.00
	5	[1.5.1.3.4.3.3.5.1.1.2.1.3.2.1.5.3.1]	124	104620	72.00
NSGA-II-TCQT	1	[1.1.2.3.1.1.1.1.1.1.1.1.1.1.1.1.1.1]	104	153320	92.20
	2	[1.2.2.3.1.1.2.1.1.1.2.1.2.1.1.1.1.1]	104	145820	87.29
	3	[5.3.1.3.4.3.3.2.4.2.1.3.3.2.1.5.3.3]	157	102915	71.56
	4	[5.4.1.3.4.3.3.1.3.1.1.2.3.1.1.4.3.1]	141	104850	74.88
	5	[1.1.2.2.1.1.1.1.1.1.1.1.1.1.1.1.1.1]	104	158320	93.53
MODE-TCQT	1	[1.3.1.1.2.1.1.3.1.1.1.1.1.1.1.5.1.1]	104	163100	95.10
	2	[1.5.3.3.4.3.3.5.1.1.3.1.3.3.1.5.1.1]	114	105270	71.55
	3	[1.1.1.2.2.3.1.2.1.1.1.1.1.1.1.2.1.1]	114	133315	90.06
	4	[2.5.1.3.4.2.3.2.1.1.3.1.3.2.1.5.1.1]	109	120615	77.01
	5	[5.5.3.3.4.3.3.5.1.1.1.1.3.2.1.5.1.1]	124	104420	72.08
OMODE-TCQT	1	[1.2.1.1.2.1.1.2.1.1.1.1.1.1.1.2.1.1]	104	164715	96.17
	2	[1.1.1.2.1.1.1.1.1.1.1.1.1.1.1.1.1.1]	104	158820	95.03
	3	[1.5.3.3.4.3.3.5.1.1.3.1.3.3.1.5.1.1]	114	105270	71.55
	4	[5.5.3.3.4.3.3.5.4.2.3.4.3.3.1.5.3.3]	159	99870	65.24
	5	[1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.2.1.1]	104	167820	97.33

Table 5. Comparison of C-metric for different algorithms.

C-metric	C(A1,A2)	C(A2,A1)	C(A1,A3)	C(A3,A1)	C(A1,A4)	C(A4,A1)
Best	0.71	0.15	0.88	0.14	0.93	0.11
Worst	0.28	0.01	0.58	0.00	0.39	0.00
Average	0.43	0.08	0.75	0.04	0.73	0.03
Std.	0.15	0.05	0.09	0.04	0.16	0.04

Table 6. Comparison of SP-metric for different algorithms.

Spread	OMODE	MODE	MOPSO	NSGA-II
Best	0.6931	0.7308	0.7043	0.9761
Worst	0.8461	1.0670	0.9641	1.2699
Average	0.7671	0.8688	0.7848	1.1005
Std.	0.0428	0.1010	0.0957	0.0876

Table 7. Comparison of HV-metric for different algorithms.

HV	OMODE	MODE	MOPSO	NSGA-II
Best	0.9998	0.9775	0.9832	0.9533
Worst	0.9165	0.9158	0.9000	0.9000
Average	0.9579	0.9481	0.9325	0.9262
std.	0.0291	0.0242	0.0287	0.0194

based learning operator has improved the chance of starting with a closer (fitter) solution by simultaneously checking the opposite solution. By doing this the fitter one (guess or opposition guess) may be chosen as an initial solution. In fact, according to probability theory, a guess will be further from the solution than its opposite guess 50% of the time. Therefore, starting with the closer of the two guesses (as judged by its fitness) has accelerated the convergence.

Conclusions

The research provides an alternative DE-based methodology to solve the construction TCQT problems, that is, determining an optimal set of construction methods for all activities which corresponds to the minimum project time, cost and the maximum project quality. The novel multi-objective optimization algorithm OMODE utilizes an opposition-based learning technique to enhance significantly the diversity of the initial population and generate potential candidates at the beginning of the algorithm. OBL is employed through the entire evolution process to balance OMODE diversity and convergence. Experimental results demonstrate that the proposed OMODE approach efficiently solves multi-objective TCQT problems and finds multi-objective Pareto optimal solutions in one simulation run. OMODE displayed better diversity characteristics, yielded better compromise solutions, and attained a higher degree of satisfaction than the widely used multi-objective evolutionary algorithms MODE, MOPSO, and NSGA-II. The Pareto front generated by OMODE provides information that helps construction-project decision makers determine the optimal tradeoff among the three important project considerations of project duration, project cost, and project quality.

The proposed OMODE is expected to assist researchers or practitioners in effectively analyzing and planning construction projects. However, the scheduling tool in this study only considered deterministic and static input data and assumed that the logical relationship amongst activities is ‘finish-to-start’. Despite the model’s positive application on the aforementioned case study, it is still worth to point out some limitations and suggestions for improvement. The proposed model need to be tested with large scale repetitive projects, i.e. a high rise building or even a skyscraper project with thousands of activities involved. The computational efficiency of the proposed model in such large scale situation is of practical concern. Integrating advanced techniques such as parallel computing into current model represents an interesting direction for future research. Therefore, further study is required to build an optimization model to stochastic data. Other possible dependencies in activity relationship including ‘finish-to-finish’, ‘start-to-start’, ‘start -to- finish’ could be involved in the proposed model.

Acknowledgments

This research is funded by Ho Chi Minh City University of Technology - VNU-HCM under grant number T-KTXD-2018-47.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Afshar A, Kaveh A, Shoghli O. 2007. Multi-objective optimization of time–cost–quality using multi-colony ant algorithm. Asian J. Civil Eng. (Building and Housing). 8(2):113–124.

Afshar A, Ziaraty A, Kaveh A, Sharifi F. 2009. Nondominated archiving multicolony ant algorithm in time–cost trade-off optimization. J Constr Eng Manage. 135(7):668–674.

Ahandani M, Alavi-Rad H. 2012. Opposition-based learning in the shuffled differential evolution algorithm. Soft Comput. 16(8):1303–1337.

Ahandani M, Shirjoposh N, Banimahd R. 2010. Three modified versions of differential evolution algorithm for continuous optimization. Soft Comput. 15(4):803–830.

Ali M, Siarry P, Pant M. 2012. An efficient differential evolution based algorithm for solving multi-objective optimization problems. Eur J Oper Res. 217(2):404–416.

Aminbakhsh S, Sonmez R. 2017. Pareto front particle swarm optimizer for discrete time-cost trade-off problem. J Comput Civ Eng. 31(1):04016040.

Babu AJG, Suresh N. 1996. Project management with time, cost, and quality considerations. Eur J Oper Res. 88(2): 320–327. 1/20/

Burns SA, Liu L, Feng C-W. 1996. The LP/IP hybrid method for construction time-cost trade-off analysis. Constr Manage Econom. 14(3):265–276.

Cheng M-Y, Tran D-H. 2014. Two-phase differential evolution for the multiobjective optimization of time-cost tradeoffs in resource-constrained construction projects. IEEE Trans Eng Manage. 61(3):450–461.

Cheng M-Y, Tran D-H. 2015. Opposition-based multiple objective differential evolution (OMODE) for optimizing work shift schedules. Autom Constr. 55(0):1–14.

Coello Coello CA. 2006. Evolutionary multi-objective optimization: a historical view of the field. IEEE Comput Intell Magange. 1(1):28–36.

Das S, Suganthan PN. 2011. Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Computat. 15(1): 4–31.

De P, James Dunne E, Ghosh JB, Wells CE. 1995. The discrete time-cost tradeoff problem revisited. Eur J Oper Res. 81(2):225–238.

Deb K, Pratap A, Agarwal S, Meyarivan T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Computat. 6(2):182–197.

Elbeltagi E, Ammar M, Sanad H, Kassab M. 2016. Overall multiobjective optimization of construction projects scheduling using particle swarm. Eng Const Arch Man. 23(3):265–282.

El-Rayes K, Kandil A. 2005. Time-cost-quality trade-off analysis for highway construction. J Constr Eng Manage. 131(4):477–486.

El-Rayes K, Moselhi O. 2001. Optimizing resource utilization for repetitive construction projects. J Const Eng Manage. 127(1):18–27.

Eshtehardian E, Afshar A, Abbasnia R. 2008. Time–cost optimization: using GA and fuzzy sets theory for uncertainties in cost. Constr Manage Econom. 26(7):679–691.

Fonseca CM, Fleming PJ. 1995. An overview of evolutionary algorithms in multiobjective optimization. Evol Comput. 3(1):1–16.

Fu F, Zhang C. A new approach to time-cost-quality trade-off problem for construction project. In *2014 International Conference on Management Science & Engineering 21th Annual Conference Proceedings*, 2014, pp. 1836–1842.

Hyari KH, El-Rayes K, El-Mashaleh M. 2009. Automated trade-off between time and cost in planning repetitive construction projects. Constr Manage Econom. 27(8): 749–761.

Jiang A, Zhu Y. 2010. A multi-stage approach to time-cost trade-off analysis using mathematical programming. Int J Constr Manage. 10(3):13–27.

Kalhor E, Khanzadi M, Eshtehardian E, Afshar A. 2011. Stochastic time–cost optimization using non-dominated archiving ant colony approach. Autom Constr. 20(8): 1193–1203.

Kazaz A, Ulubeyli S, Er B, Acikara T. 2016. Construction materials-based methodology for time-cost-quality trade-off problems. Procedia Eng. 164:35–41.

Khalili-Damghani K, Tavana M, Abtahi A-R, Santos Arteaga FJ. 2015. Solving multi-mode time–cost–quality trade-off problems under generalized precedence relations. Optimiz Meth Softw. 30(5):965–1001.

Khang DB, Myint YM. 1999. Time, cost and quality trade-off in project management: a case study. Int J Pro Manage. 17(4):249–256.

Kim J, Kang C, Hwang I. 2012. A practical approach to project scheduling: considering the potential quality loss cost in the time–cost tradeoff problem. Int J Proj Manage. 30(2):264–272.

Li H, Cao J, Love P. 1999. Using machine learning and GA to solve time-cost trade-off problems. J Constr Eng Manage. 125(5):347–353.

Nabipoor AE, Roghanian E, Najafi AA, Mazinani M. 2013. A multi-mode resource-constrained discrete time–cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm. Scientia Iranica. 20(3):931–944.

Ng S, Zhang Y. 2008. Optimizing construction time and cost using ant colony optimization approach. J Constr Eng Manage. 134(9):721–728.

Ö. Bettemir H, Talat Birgönül M. 2017. Network analysis algorithm for the solution of discrete time-cost trade-off problem. KSCE J Civ Eng . 21(4):1047–1058.

Ogunsemi DR, Jagboro GO. 2006. Time-cost model for building projects in Nigeria. Constr Manage Econom. 24(3):253–258.

Price KV, Storn RM, Lampinen JA. 2005. Differential evolution a practical approach to global optimization. Springer-Verlag; Berlin, Heidelberg.

Rahnamayan S, Tizhoosh HR, Salama MMA. 2008. Opposition-based differential evolution. IEEE Trans Evol Computat. 12(1):64–79.

Ramesh S, Kannan S, Baskar S. 2012. Application of modified NSGA-II algorithm to multi-objective reactive power planning. Appl Soft Comput. 12(2):741–753.

Reddy M, Kumar D. 2007. Multiobjective differential evolution with application to reservoir system optimization. J Comput Civ Eng. 21(2):136–146.

Shahriari M. 2016. Multi-objective optimization of discrete time–cost tradeoff problem in project networks using non-dominated sorting genetic algorithm. J Ind Eng Int. 12(2):159–169.

Storn R ,Price K. 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J Global Optimiz. 11(4):341–359.

Tareghian HR, Taheri SH. 2006. On the discrete time, cost and quality trade-off problem. Appl Math Comput. 181(2):1305–1312.

Tareghian HR, Taheri SH. 2007. A solution procedure for the discrete time, cost and quality tradeoff problem using electromagnetic scatter search. Appl Math Comput. 190(2):1136–1145.

Tizhoosh HR. 2005. Opposition-based learning: a new scheme for machine intelligence. In Comput Intell Modell Contr Autom. 1:695–701.

Tran D-H, Cheng M-Y, Cao M-T. 2015. Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem. Knowledge-Based Syst. 74:176–186.

Tran D-H, Luong D-L, Duong M-T, Le T-N, Pham A-D. 2017. Opposition multiple objective symbiotic organisms search (OMOSOS) for time, cost, quality and work continuity tradeoff in repetitive projects. J Comput Design Eng. 5(2):160–172.

Wang L, Singh C. 2009. Reserve-constrained multiarea environmental/economic dispatch based on particle swarm optimization with local search. Eng Appl Artif Intell. 22(2):298–307.

Wang Y-N, Wu L-H, Yuan X-F. 2010. Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. Soft Comput. 14(3):193–209.

Wood DA. 2017. Gas and oil project time-cost-quality tradeoff: Integrated stochastic and fuzzy multi-objective optimization applying a memetic, nondominated, sorting algorithm. J Nat Gas Sci Eng. 45:143–164.

Wu LH, Wang YN, Yuan XF, Zhou SW. 2010. Environmental/economic power dispatch problem using multi-objective differential evolution algorithm. Electric Power Syst Res. 80(9):1171–1181.

Yang I-T. 2007. Using elitist particle swarm optimization to facilitate Bicriterion time-cost trade-off analysis. J Constr Eng Manage. 133(7):498–505.

Zahraie B, Tavakolan M. 2009. Stochastic time-cost-resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets. J Constr Eng Manage. 135(11):1162–1171.

Zeng Z, Hong M, Man Y, Li J, Zhang Y, Liu H. 2018. Multi-object optimization of flexible flow shop scheduling with batch process – consideration total electricity consumption and material wastage. J Clean Product. 183: 925–939.

Zhang H, Li H. 2010. Multi-objective particle swarm optimization for construction time-cost tradeoff problems. Constr Manage Econom. 28(1):75–88.

Zhang H, Xing F. 2010. Fuzzy-multi-objective particle swarm optimization for time–cost–quality tradeoff in construction. Automation Constr. 19(8):1067–1075.

Zheng F, Zecchin A, Simpson A. 2013. Self-adaptive differential evolution algorithm applied to water distribution system optimization. J Comput Civ Eng. 27(2):148–158.

Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q. 2011. Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evolut Comput. 1(1):32–49.

Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VGD. 2003. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Computat. 7(2):117–132.

Zitzler E, Thiele L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Computat. 3(4):257–271.