Knowledge-Based Systems 74 (2015) 176–186

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem

Duc-Hoc Tran *, Min-Yuan Cheng [1], Minh-Tu Cao [2]

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan

A R T I C L E I N F O

Article history:
Received 23 May 2014
Received in revised form 5 November 2014
Accepted 15 November 2014
Available online 21 November 2014

Keywords:
Multi-objective analysis
Artificial bee colony
Differential evolution
Time–cost–quality tradeoff
Construction management

A B S T R A C T

Time, cost, and quality are three important but often conflicting factors that must be optimally balanced during the planning and management of construction projects. Tradeoff optimization among these three factors within the project scope is necessary to maximize overall project success. In this paper, the MOABCDE-TCQT, a new hybrid multiple objective evolutionary algorithm that is based on hybridization of artificial bee colony and differential evolution, is proposed to solve time–cost–quality tradeoff problems. The proposed algorithm integrates crossover operations from differential evolution (DE) with the original artificial bee colony (ABC) in order to balance the exploration and exploitation phases of the optimization process. A numerical construction project case study demonstrates the ability of MOABCDE-generated, non-dominated solutions to assist project managers to select an appropriate plan to optimize TCQT, which is an operation that is typically difficult and time-consuming. Comparisons between the MOABCDE and four currently used algorithms, including the non-dominated sorting genetic algorithm (NSGA-II), the multiple objective particle swarm optimization (MOPSO), the multiple objective differential evolution (MODE), and the multiple objective artificial bee colony (MOABC), verify the efficiency and effectiveness of the developed algorithm.

1. Introduction

Planning a schedule that achieves both project deadline and project budget requirements is one of the most difficult tasks in construction management. Moreover, minimizing the time and the cost of construction projects is typically a key determinant of construction company success [1]. Researchers have established numerous models and applied many methods to optimize the time–cost tradeoff (TCQT) problem in the construction industry. Of these methods, evolutionary algorithms have proven relatively more efficient at avoiding local optimization [2–6]. The costs of resources and technologies used in a project are typically associated with work productivity and, thus, project duration. Conversely, higher resource and technology costs are typically associated with shorter project durations. However, shorter project durations and/or costs may decrease the quality of project work, resulting in earlier-than-projected aging/deterioration and higher-than-anticipated maintenance and rehabilitation costs. Therefore, determining the optimal combination of execution methods, construction technologies, and resource utilization plans that minimize time and costs while simultaneously maximizing quality, i.e., the TCQT problem, is crucial for construction planners [7].

Quality is a crucial factor that correlates highly with time and cost factors [8]. Scholars have established various models and methods that attempt to optimize the time–cost–quality tradeoff. Babu and Suresh [9] proposed an initial methodology for studying the TCQT problem using three inter-related linear programming models. Khang and Myint [10] established the validity of this methodology in a trial application on a cement factory construction project. El-Rayes and Kandil [11] used multiple objective evolutionary algorithms in a multi-objective model that they used to search for an optimal resource utilization plan to minimize construction cost and time while maximizing quality. They applied a genetic algorithm (GA) to quantify and consider quality and used the example of a highway construction project to visualize the optimal tradeoffs among the three variables. Zhang and Xing [7] indicated that time, cost, and, especially, quality are difficult to describe precisely. Therefore, they proposed a Pareto-based multi-objective Particle Swarm Optimization (PSO) for the fuzzy TCQT problem. More recently, hybridized evolutionary algorithms have been used successfully to solve the TCQT problem [12,13].

* Corresponding author. Tel.: +886 92636949.
E-mail addresses: duchoc87@gmail.com (D.-H. Tran), myc@mail.ntust.edu.tw (M.-Y. Cheng), caominhtu2201@gmail.com (M.-T. Cao).
[1] Tel.: +886 2 2733 6596; fax: +886 2 2730 1074.
[2] Tel.: +886 921634420.

http://dx.doi.org/10.1016/j.knosys.2014.11.018
0950-7051/

The Artificial Bee Colony (ABC) algorithm is one of the most recently introduced evolutionary methods [14,15]. The ABC is a swarm intelligence-based optimization algorithm inspired by honeybee foraging behavior. With its few parameters, ABC is simple to implement and relatively efficient and robust in comparison to other algorithms. It has been applied successful to solve complex multi-model optimization problems [16]. The ABC seems particularly suited for multi-objective optimization problems mainly because of solution quality and the high speed of convergence that the algorithm presents for single-objective optimization [17]. The ABC algorithm distributes information on food location throughout the entire population of bees. This characteristic makes ABC good at exploration but poor at exploitation and thus inadequate for problems that must apply existing information to find a better solution. Additionally, ABC converges relatively slowly for certain complex issues [18]. 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 [19]. The ability of DE to provide efficient solutions for complex problems with relatively simple operations has encouraged many researchers to develop MODE-based techniques [20,21].

Although meta-heuristic methods have been proven to have superior features than other traditional methods, they also suffer some limitations. For example, the convergence speed of the ABC algorithm is typically slower than the convergence speeds of representative population-based algorithms such as DE when handling unimodal problems because the ABC cannot utilize the information adequately to determine the most promising search direction [22]. The performance of the original DE algorithm is highly dependent on the mutation and crossover operator. In some cases, it may become easily trapped in the local minimum or converge too slowly [23]. In addition, numerous researchers have found that a skilled combination of two meta-heuristics may be beneficial and perform significantly better than single pure meta-heuristic algorithm in handling real-world and large-scale problems [24,25]. Therefore, hybridization with other algorithms offers the potential to further improve the performances of ABC and DE. The superior performance of hybridized ABC and DE over other algorithms in single and multiple-objective problems have been widely reported and verified [18,26–28]. Despite many reports of impressive hybridized ABC-DE performance in benchmark functions and practical applications, this algorithm has yet to be applied to solving the TCQT problem. Therefore, this paper applies the hybridized ABC-DE algorithm in a model that is designed to solve the TCQT problem.

Based on the above, this study develops the novel, hybrid MOABCDE model to conduct TCQT analysis. This new algorithm follows the same scheme as ABC while incorporating DE operators, which are used to generate new candidate solutions by combining the parent individual and several other individuals of the same population according to a simple formulation. The hybridization balances the exploration, exploitation, and convergence capabilities of the optimization process. Additionally, this new algorithm offers a strategy to balance the tradeoff between local search and global search capabilities and enhances MOABC performance. The objective is to demonstrate that the proposed algorithm attains fast convergence without losing solution diversity on the Pareto front.

The remainder of this paper is organized as follows. In Section 2, the TCQT problem is mathematically formulated. In Section 3, we review briefly the literature related to the establishment of the new optimization model. In Section 4, we present detailed descriptions of the proposed optimization model for the TCQT problem. Section 5 uses a numerical case study and result comparisons to demonstrate the performance of the newly developed model. Finally, some conclusions and suggested directions for future work are presented in Section 6.

2. Time–cost–quality tradeoff problem formulation

In the TCQT problem, a project is represented by the diagram $G = (A)$, which is an activity-on-node network with N activities. These activities are numbered from 0 to $N + 1$ in the project network, where activities 0 and $N + 1$ are "dummy" activities denoting, respectively, the start and finish of the project. P is the set of all paths in the activity-on-node network, starting from activity (0) and ending at activity ($N + 1$). P_l is the set of activities contained in path $l \in P$. Each activity $i \in A$ is associated with several execution methods. Each method has its time T_i, cost C_i, and quality Q_i. The TCQT problem concentrates mainly on selecting an optimal combination of execution methods for all activities in order to arrive at an optimal compromise among time, cost, and quality for the project. The project time, cost, and quality are quantified as follows:

2.1. Calculating overall time in project

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

$$\begin{aligned} &\textit{Minimize project time } T = \sum_{n=1}^{l} T_n^{S_n} = Max_{\forall n}(ES_n + d_n) \\ &ES_n = \underset{\text{all predecessors } m \text{ of } n}{Maximum}(ES_m + d_m) \end{aligned} \tag{1}$$

where $T_n^{S_n}$ is duration of the activity n $\{n = 1, 2, \ldots, l\}$ on the critical path for a specific option of resources (S_n); l is the total number of critical activities on a specific critical path. ES_i is the earliest start of activity n, d_n is the duration of activity n. 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.

2.2. Calculating overall cost

The total cost of a project includes direct costs, indirect costs and tardiness cost. The direct cost (DC) of the project is the sum of the direct costs of all its activities. Indirect costs (IC) are costs proportional to the duration of the project as a whole. In accordance with contract requirements, contractors are often subject to tardiness cost (TC) because of delays in project completion.

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

$$\textit{Minimize project } \mathrm{cost} = \sum_{i=1}^{N} \left(DC_i^{S_i} + IC_i^{S_i} + TC_i^{S_i} \right) \tag{2}$$

where $DC_i^{S_i}, IC_i^{S_i}, TC_i^{S_i}$ is the direct, indirect and tardiness cost of activity i, respectively, for a specific option of execution methods (S_n) and N is the total number of activities.

2.3. Calculating overall quality

Since a project comprises various resources such as materials, machines, labor, the overall project quality is determined by each activity's quality. On the basic of the previous researches [8,12] and the collected data, we define Quality Performance Index (QPI) as the achieved level of the quality goal under the contract in a given period.

We define QPI_i as the quality of a single activity as follows:

$$QPI_i = a_i T_i^2 + b_i T_i + c_i \tag{3}$$

where $QPI_i \in [0,1]$, $i = 1, 2, \ldots, N$, T_i is duration of activity i, with $T_i > 0$, a_i, b_i, c_i are the coefficients decided by the quadratic function (Fig. 1). Fig. 1 shows that SD_i, BD_i, LD_i are the shortest duration, best duration, longest duration of activity i.

The third and final objective, is the maximization of project's overall quality.

$$\text{Maximize project quality} = \sum_{i=1}^{N}(QPI_i)/N \tag{4}$$

The scope of this study is limited to a deterministic environment in which decisions are made with deterministic input data from the case study. The assumptions of the case study are that all the objective functions are quantified data and the mean values of the resources are used.

3. Literature review

3.1. Review of multiple objective optimization

A multiple objective optimization (MOO) problem involves several conflicting objectives and has a set of Pareto optimal solutions. An MOO model considers a vector of decision variables, objective functions, and constraints. Decision makers attempt to minimize (or maximize) the objective functions [29].

A MOO problem may be mathematically formulated as:

$$\min_{X \in D} f(X) = [f_1(X), f_2(X), \ldots, f_k(X)] \tag{5}$$

$$\text{s.t. } g_i(X) \geqslant 0; \quad i = 1, \ldots, m \tag{6}$$

$$h_j(X) = 0; \quad j = 1, \ldots, p \tag{7}$$

$$D = \{X | g(X) \geqslant 0, h(X) = 0\} \tag{8}$$

where $f(X)$ is the objective vector, k is the number of objective functions. $g_i(X)$ is the set of inequality constraints, and $h_j(X)$ is the set of inequality equality constraints. The m and p are the number of inequality and equality constraints, respectively. The solution $X(x_1, x_2, \ldots, x_n)^T$ is a vector of n decision variables in feasible region D. The multi-objective optimization problem works to determine those vectors X that yield the optimum values for all the objective functions from the set D of all vectors which satisfy (6) and (7).

Because this problem rarely presents a unique solution, decision makers are expected to choose a solution from among a set of efficient solutions, known collectively as the Pareto. The Pareto dominance is formally defined as follows [30]:

Solution $X_1(x_{1.1}, x_{1.2}, \ldots, x_{1.n})^T$ dominates $X_2(x_{2.1}, x_{2.2}, \ldots, x_{2.n})^T$ if both the conditions are satisfied:

1. $\forall\, i \in (1, 2, \ldots, k)$: $f_i(X_1) \leqslant f_i(X_2)$. The solution X_1 is no worse than X_2 in attaining all objectives.
2. $\exists\, i \in (1, 2, \ldots, k)$: $f_i(X_1) < f_i(X_2)$. The solution X_1 is strictly better than X_2 in at least one objective.

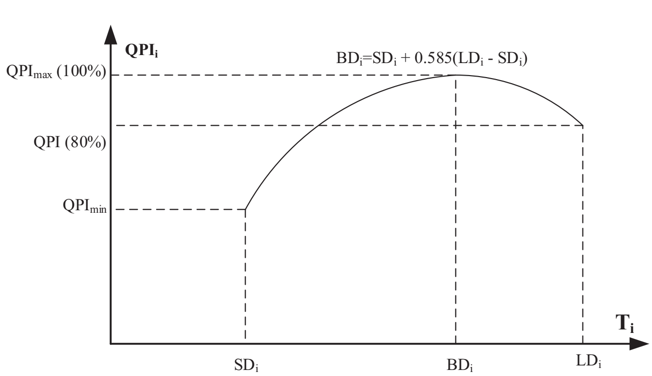

Fig. 1. Graphic QPI for activity.

So, while comparing two different solutions X_1 and X_2, there are three possibilities of dominance relation between them.

- X_1 dominates X_2.
- X_1 is dominated by X_2.
- X_1 and X_2 are non-dominated to each other.

A non-dominated solution means that no other solution has been found that dominates it. The set of non-dominated solutions is called the Pareto front.

Using Multiple Objective Evolutionary Algorithms (MOEAs) to analyze MOO problems has attracted increasing attention in recent years. Numerous researchers from several domains of science and engineering have applied MOEAs to solve optimization problems that arise in their own fields [21], including engineering optimization [31], construction management [32,33], scheduling problems [34,35], economic and finance problems [36], and group decision making [29]. New multiple objective optimization algorithms will continue to emerge as MOO problems become more complex.

3.2. Basic artificial bee colony

The ABC algorithm is a recently developed population-based optimization algorithm now widely used to solve multidimensional optimization problems [14]. This swarm-intelligence-based optimization algorithm was inspired by honeybee foraging behavior. ABC incorporates three kinds of honeybees: employed bees, onlooker bees, and scout bees. Food source position indicates a possible optimization problem solution and food source profitability corresponds to the quality of the associated solution (fitness).

The ABC algorithm is an iterative algorithm. It commences the search process by generating NF number of food source positions randomly in D dimensions. NF does not change during the optimization process in the standard ABC algorithm. Eq. (9) was used as our initial population generator. Amount of nectar at each food source position was calculated after all food source positions were generated.

$$X_{i,j} = LB_j + rand[0,1] \times (UB_j - LB_j); \quad (j = 1, \ldots, D; i = 1, \ldots, NP) \tag{9}$$

where LB_j and UB_j indicates the lower and upper bound of the ith decision variable; $rand[0,1]$ denotes a uniformly distributed random number between 0 and 1; and $X_{i,j}$ is the ith decision variable (food source position) in the initial population.

Each employed bee chooses a new candidate food source position to update feasible solutions based on the neighborhood of the previously selected food source. A candidate solution $U_{i,j}$ may thus be generated from the old solution $X_{i,j}$ as in Eq. (10). Employed bees are created from the strength of previously discovered sources.

$$U_{i,j} = X_{i,j} + \phi_{i,j}(X_{i,j} - X_{k,j}) \tag{10}$$

where $\{k, i\} \in \{1, 2, \ldots, NF\}$ and $k \neq i$; $j \in \{1, 2, \ldots, D\}$ are randomly chosen indices; $\phi_{i,j}$ is a random number in the range $[-1, 1]$.

Amount of nectar is used to compare the candidate solution (candidate food source position) with the old solution. The candidate solution will replace the old solution if its food source quality is equal to or better than the latter; otherwise, the old solution is retained. Food source information is shared with onlooker bees when employed bees return to their hive. The information sharing stage of the ABC algorithm generates collective intelligence. Probability value influences the behavior of onlooker bees, which select food sources based on probability. Probability value is calculated as:

$$p_i = fit_i \Big/ \sum_{j=1}^{NF} fit_j \tag{11}$$

where fit_i is the fitness value of the ith food source.

An onlooker bee chooses a food source depending on the probability value p_i associated with that food source. ABC produces an onlooker bee using Eq. (11). After evaluating the source, greedy selection is applied and the onlooker bee either updates the new position by removing or retaining the old solution. The ABC algorithm updates the best food source position after termination of the onlooker bee phase. A new best food source position will replace the old if the former provides an equal or better amount of nectar. Otherwise, the old food source position remains valid.

If food source X_i (solution X_i) shows no further improvement through a continuous pre-determined number of cycles, then Eq. (9) is used to replace that food source with a new food source discovered by the scout bee. The optimization process terminates when the user-determined stop criterion is reached. The final optimal solution is available to the user once the optimization process terminates.

3.3. Basic differential evolution

DE is a simple population-based, direct-search used to solve global optimization problems [19,37]. 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 $X_i^G = \left\{x_{i,1}^G, x_{i,2}^G, \ldots, x_{i,D}^G\right\}$, $i = 1, 2, \ldots, NP$ 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 the mutation operator and the crossover (recombination) operator to produce 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.

$$V_i^{G+1} = X_{r1}^G + F\left(X_{r2}^G - X_{r3}^G\right) \tag{12}$$

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 $U_i^{G+1} = \{u_{i,1}^G, u_{i,2}^{G+1}, \ldots, u_{i,D}^{G+1}\}$ is generated using the following scheme.

$$u_{i,j}^{G+1} = \begin{cases} v_{i,j}^{G+1} & \text{if } (rand_j[0,1) \leqslant CR \text{ or } j = j_{rand} \\ x_{i,j}^G & \text{otherwise} \end{cases} \quad j = 1, 2, \ldots, D \tag{13}$$

$CR \subset [0,1]$ 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:

$$X_i^{G+1} = \begin{cases} U_i^{G+1} & \text{if } f\left(U_i^{G+1}\right) < f\left(X_i^G\right) \\ X_i^G & \text{otherwise} \end{cases} \tag{14}$$

Once the memberships of the next generation are selected, the evolutionary cycle of the DE iterates until the specified termination criterion is reached.

3.4. Related works on modified ABC and DE

Storn and Price [19] introduced the DE in 1997 and Karaboga [38] proposed the ABC in 2005. Many researchers worldwide have used these algorithms in numerous variants of the basic algorithm formats in order to achieve improved performance [20,39]. Some of these variants have been designed to tackle specific applications while others are generalized for numerical optimization. Talbi and Batouche [40] proposed a hybrid approach based on DE and PSO to solve multimodal image registration. Zhang et al. [41] studied the hybridization of the DE and PSO for unconstrained optimization problems. Omkar et al. [42] presented a generic multi-objective approach named the Vector Evaluated Artificial Bee Colony for composite design optimization. Kang et al. [43] combined Rosenbrock's rotational direction method with the ABC to improve the ABC search capabilities for accurate global optimization. Kiran and Gündüz [44] incorporated the GA crossover operation-based neighbor selection technique for information sharing in the hive of the ABC process to improve local search and exploration of the ABC. In addition, a number of studies [18,45] have examined the hybridization of ABC and DE.

An extensive review of the studies conducted on ABC and DE in the literature indicates that the ABC has powerful global search ability but poor local search ability [46], while the DE has powerful local search ability but poor global search ability [47,48]. To provide powerful global search capability, a diverse population should be maintained during iterations while the population is saturated (exploited) during the iteration that is designed to provide powerful local search capability. Thus, exploration (diversification) and exploitation (intensification) in the algorithms should be balanced to obtain better-quality results for the optimization problems [49]. In this study, combining the best features of the two algorithms ABC and DE balances the local search capabilities and the global search capabilities of the proposed algorithm ABCDE in order to obtain better-quality results.

4. Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem (MOABCDE-TCQ)

This section describes the hybrid multiple objective artificial bee colony differential evolution (MOABCDE) algorithm that was designed in this study based on the original ABC [14] and DE algorithms [19] in order to solve the TCQT problem. MOABCDE first creates the population (solutions) that will be exploited in the employed bee phase. Then, the DE crossover-mutation operators are applied to improve the population. Subsequently, a probability vector is calculated according to the qualities of the solutions to select those solutions that will enter the onlooker bee phase. These solutions (bees) analyze the neighborhood of the selected solution and determine the better one. The DE crossover-mutation operators are also applied in this phase. Afterward, the algorithm examines which solutions do not improve the solution and then assigns scout bees to find new solutions to replace these suboptimal solutions. Finally, the best solutions will be kept and exploited by the employed bee phase in the next generation. Fig. 2 shows the overall operational architecture of the proposed algorithm. The steps of MOABCDE-TCQT are described as follows:

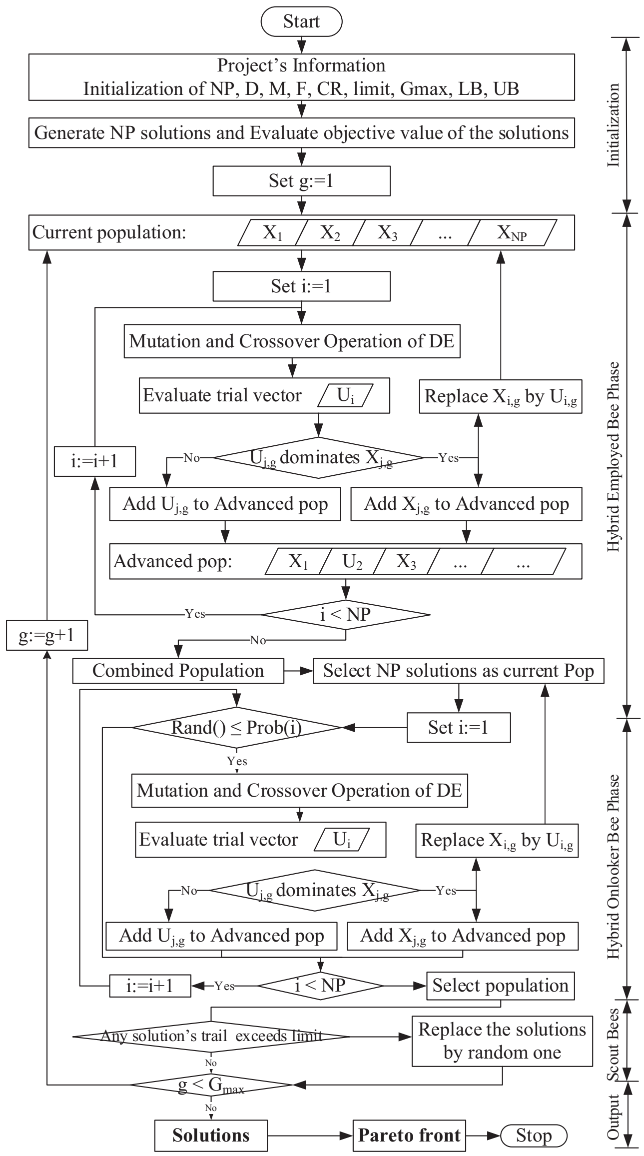

Fig. 2. MOABCDE flowchart for the TCQT problem.

4.1. 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 (D_i), activity cost (C_i), activity quality (QPI_i), and execution methods (S_n) for each activity. In addition, the user also must provide parameter settings for the search engine (MOABCDE) such as the value of population size *NP*, number of decision variables *D*, number of objective functions *M*, value of "*limit*", value of the mutant constant *F*, value of the crossover probability constant *CR*, maximum number of generations G_{max}, the lower bound (*LB*) and the upper bound (*UB*) of decision variables. With these inputs, the optimizer conducts calculations to obtain an optimal set of execution 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 MOABCDE may be guided toward more promising areas if the initial population is spread as much as possible over the objective function surface. Hence, the *NP* individuals of the population may be easily generated as Eq. (9).

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

$$X = [X_{i,1}, X_{i,2}, \ldots, X_{i,j}, \ldots, X_{i,D}] \tag{15}$$

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 execution method for activity *j*. execution method $X_{i,j}$ is an integer number in the range $[1, M_j]$ ($j = 1$ to *D*), meaning one position from M_j execution methods. Because the original ABC operates with real-value variables, a function is employed to convert the execution methods of those activities from real values to integer values within the feasible domain.

$$X_{i,j} = Ceil(rand[0,1] \times UB(j)) \tag{16}$$

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 execution methods for each activity. Ceil is a function to round a real number to the nearest integer greater than or equal to it.

The search engine (MOABCDE) takes into account the results obtained from the scheduling module and the search for an optimal combination of execution methods for each activity. This research used three contradicting objectives. Section 2 describes the formulae for each objective function.

4.2. Population selection procedure

During the optimization process, size of population remains *NP*. *NP* best (elite) solutions are selected from the combined population, which mixed of the current and advanced population together. While the "highest fitness value" solution is the best solution in the single objective solution scenario, a two-solutions dominance approach is used in multi-objective scenarios. Thus, the *NP* solutions in this research are selected using fast non-dominated sorting [30] and the entropy crowding technique [50]. Fig. 3 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

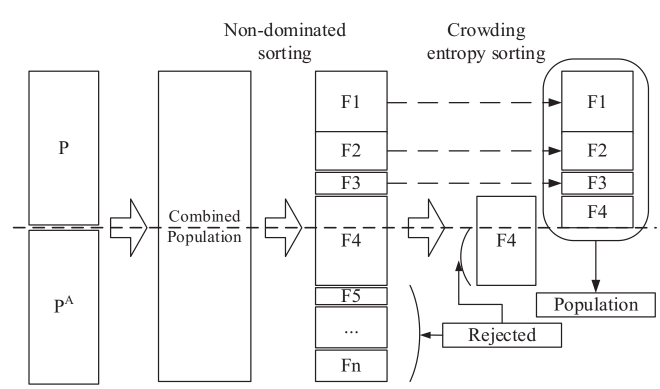

Fig. 3. Population selection procedure.

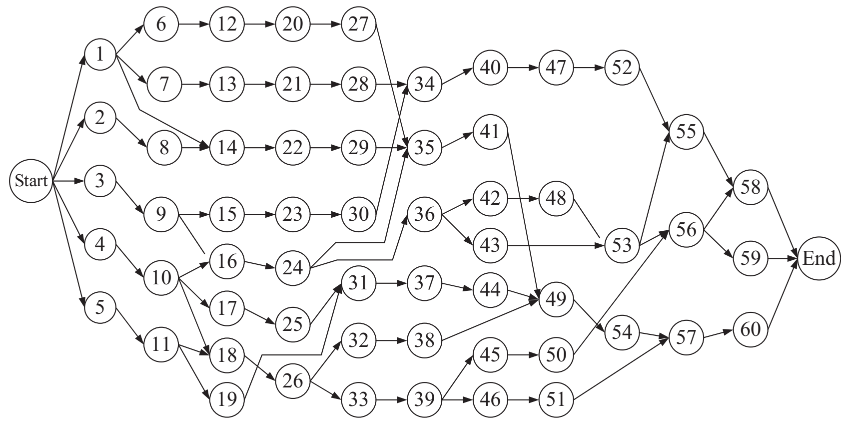

Fig. 4. Network of project.

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.

4.3. Hybrid DE operators

Once initialized, the DE crossover-mutation operators mutate the population to produce a set of mutant vectors. A mutated vector V_i^{G+1} is created using Eq. (12) that corresponds to the target vector X_i^G. 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. (13).

4.4. 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 study used a new selection mechanism proposed by Ali, et al. [51]. 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 moved to another (advanced) population. 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 2*NP*. 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.

4.5. Probability calculation

Employed bees return to their hive and share food-source information with onlooker bees. The information sharing stage of the ABC algorithm generates collective intelligence. The probability value influences the behavior of onlooker bees, which select food sources based on probability. The probability value is calculated as:

$$P_i = 0.9^* Fit(X_i) / \max(Fit(X_i)) + 0.1 \tag{17}$$

where $Fit(x_i)$ is the fitness value of the ith solution (food source). To calculate the fitness value of solution x_i in multiple objective context, we adopted a new metric fitness calculation [27].

$$Fit(X_i) = \left(2^{X_i.rank} + \frac{1}{1 + X_i.ce}\right)^{-1} \tag{18}$$

To evaluate the fitness of a given solution x_i, we need to know the Pareto front to which it belongs (for instance, if it belongs to the first Pareto front, then $X_i.rank = 1$). Next, we calculate its crowding entropy value ($X_i.ce$) with respect to the solutions of the same Pareto front, i.e., all solution in with *rank* = 1. The Pareto front class of each solution and its crowding entropy value in the population is determined by the fast non-dominated sorting [30] and the entropy crowding technique [50], respectively.

4.6. Scout bee phase

If solution X_i (food source X_i) shows no improvement further through a continuous pre-determined number of cycles (limit), then the food source abandoned by its bee is replaced with a new food source discovered by the scout bee, Eq. (9).

4.7. Stopping conditions

The optimization process terminates when the stopping conditions are met. The user sets the type of these conditions. Maximum generation G_{max} or maximum number of functions evaluations (NFE) may be used as the stopping criterion. This study used the maximum number of generation as stopping condition for the proposed algorithm. 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 decision makers to evaluate the pros and cons of each potential solution based on qualitative and experience-driven considerations.

5. Case study

This study analyzed a numerical construction project to demonstrate the effectiveness of the proposed MOABCDE in application to the TCQT problem. The obtained results were compared to four approaches used previously in the literature to handle the TCQT problem, including NSGA-II, MOPSO, MODE, and MOABC. The project comprised 60 construction activities, each of which has a number of possible execution methods. Fig. 4 shows the precedence relationships of the network. Table 1 shows the time, cost, and quality impact associated with each execution method (mode), with an average of 4.68 execution methods for each of the 60 activities, generating multiple billions (4.68^{60}) of possible combinations for completing the entire project. Each possible combination has a unique impact on project performance, which means that decision makers must search a large number of potential solutions to find those that establish an optimal tradeoff/balance among construction project duration, cost, and quality. We used the newly developed multi-objective optimization model to search the many potential solutions.

Table 1
Time–cost-quality options for 60-activity project.

Act	Mode 1			Mode 2			Mode 3			Mode 4			Mode 5		
	Time	Cost	Quality	Time	Cost	Quality	Time	Cost	Quality	Time	Cost	Quality	Time	Cost	Quality
1	14	3750	85.00	12	4250	99.98	10	5400	87.09	9	6250	70.19	–	–	–
2	21	11,250	85.00	18	14,800	99.37	17	16,200	94.48	15	19,650	70.19	–	–	–
3	24	22,450	85.00	22	24,900	98.54	19	27,950	92.20	17	31,650	70.19	–	–	–
4	19	17,800	85.00	17	19,400	99.37	15	21,600	70.19	–	–	–	–	–	–
5	28	31,180	85.00	26	34,200	98.54	23	38,250	92.20	21	41,400	70.19	–	–	–
6	44	54,260	85.00	42	58,450	96.76	38	63,225	94.48	35	68,150	70.19	–	–	–
7	39	47,600	85.00	36	50,750	99.42	33	54,800	94.48	30	59,750	70.19	–	–	–
8	52	62,140	85.00	47	69,700	99.92	44	72,600	96.50	39	81,750	70.19	–	–	–
9	63	72,750	85.00	59	79,450	98.54	55	86,250	97.87	51	91,500	82.97	49	99,500	70.19
10	57	66,500	85.00	53	70,250	97.63	50	75,800	99.96	46	80,750	93.53	41	86,450	70.19
11	63	83,100	85.00	59	89,450	96.76	55	97,800	99.92	50	104,250	91.78	45	112,400	70.19
12	68	75,500	85.00	62	82,000	99.14	58	87,500	98.92	53	91,800	87.79	49	96,550	70.19
13	40	34,250	85.00	37	38,500	99.42	33	43,950	88.54	31	48,750	70.19	–	–	–
14	33	52,750	85.00	30	58,450	99.86	27	63,400	90.23	25	66,250	70.19	–	–	–
15	47	38,140	85.00	40	41,500	99.77	35	47,650	87.09	32	54,100	70.19	–	–	–
16	75	94,600	85.00	70	101,250	98.36	66	112,750	99.37	61	124,500	88.54	57	132,850	70.19
17	60	78,450	85.00	55	84,500	99.92	49	91,250	83.81	47	94,640	70.19	–	–	–
18	81	127,150	85.00	73	143,250	99.98	66	154,600	90.23	61	161,900	70.19	–	–	–
19	36	82,500	85.00	34	94,800	99.42	30	101,700	70.19	–	–	–	–	–	–
20	41	48,350	85.00	37	53,250	99.92	34	59,450	88.54	32	66,800	70.19	–	–	–
21	64	85,250	85.00	60	92,600	98.08	57	99,800	99.77	53	107,500	91.17	49	113,750	70.19
22	58	74,250	85.00	53	79,100	99.09	50	86,700	99.37	47	91,500	93.53	42	97,400	70.19
23	43	66,450	85.00	41	69,800	94.06	37	75,800	99.81	33	81,400	89.07	30	88,450	70.19
24	66	72,500	85.00	62	78,500	97.19	58	83,700	99.73	53	89,350	89.35	49	96,400	70.19
25	54	66,650	85.00	50	70,100	98.54	47	74,800	99.37	43	79,500	88.03	40	86,800	70.19
26	84	93,500	85.00	79	102,500	96.93	73	111,250	99.37	68	119,750	91.51	62	128,500	70.19
27	67	78,500	85.00	60	86,450	99.37	57	89,100	92.20	56	91,500	88.03	53	94,750	70.19
28	66	85,000	85.00	63	89,750	97.63	60	92,500	99.37	58	96,800	94.48	54	100,500	70.19
29	76	92,700	85.00	71	98,500	99.09	67	104,600	98.11	64	109,900	90.23	60	115,600	70.19
30	34	27,500	85.00	32	29,800	97.63	29	31,750	96.16	27	33,800	81.57	26	36,200	70.19
31	96	145,000	85.00	89	154,800	98.68	83	168,650	98.60	77	179,500	87.64	72	189,100	70.19
32	43	43,150	85.00	40	48,300	98.85	37	51,450	97.02	35	54,600	87.09	33	61,450	70.19
33	52	61,250	85.00	49	64,350	96.49	44	68,750	97.87	41	74,500	88.03	38	79,500	70.19
34	74	89,250	85.00	71	93,800	95.04	66	99,750	99.73	62	105,100	92.63	57	114,250	70.19
35	138	183,000	85.00	126	201,500	98.85	115	238,000	97.77	103	283,750	81.57	98	297,500	70.19
36	54	47,500	85.00	49	50,750	97.27	42	56,800	97.87	38	62,750	89.52	33	68,250	70.19
37	34	22,500	85.00	32	24,100	95.97	29	26,750	99.37	27	29,800	92.93	24	31,600	70.19
38	51	61,250	85.00	47	65,800	99.00	44	71,250	98.67	41	76,500	89.07	38	80,400	70.19
39	67	81,150	85.00	61	87,600	99.42	57	92,100	98.28	52	97,450	84.76	49	102,800	70.19
40	41	45,250	85.00	39	48,400	95.97	36	51,200	99.37	33	54,700	87.09	31	58,200	70.19
41	37	17,500	85.00	31	21,200	99.98	27	26,850	92.20	23	32,300	70.19	–	–	–
42	44	36,400	85.00	41	39,750	96.49	38	42,800	99.98	32	48,300	82.97	30	50,250	70.19
43	75	66,800	85.00	69	71,200	98.54	63	76,400	97.87	59	81,300	89.52	54	86,200	70.19
44	82	102,750	85.00	76	109,500	99.14	70	127,000	95.91	66	136,800	84.11	63	146,000	70.19
45	59	84,750	85.00	55	91,400	97.63	51	101,300	99.37	47	126,500	90.23	43	142,750	70.19
46	66	94,250	85.00	63	99,500	95.49	59	108,250	99.96	55	118,500	93.53	50	136,000	70.19
47	54	73,500	85.00	51	78,500	97.04	47	83,600	98.67	44	88,700	89.07	41	93,400	70.19
48	41	36,750	85.00	39	39,800	95.97	37	43,800	99.98	34	48,500	92.93	31	53,950	70.19
49	173	267,500	85.00	159	289,700	98.15	147	312,000	99.37	138	352,500	94.20	121	397,750	70.19
50	101	47,800	85.00	74	61,300	99.05	63	76,800	91.32	49	91,500	70.19	–	–	–
51	83	84,600	85.00	77	93,650	98.24	72	98,500	99.37	65	104,600	85.84	61	113,200	70.19
52	31	23,150	85.00	28	27,600	98.85	26	29,800	99.37	24	32,750	92.93	21	35,200	70.19
53	39	31,500	85.00	36	34,250	97.04	33	37,800	99.81	29	41,250	89.07	26	44,600	70.19
54	23	16,500	85.00	22	17,800	95.97	21	19,750	99.98	20	21,200	97.02	18	24,300	70.19
55	29	23,400	85.00	27	25,250	98.54	26	26,900	99.98	24	29,400	92.20	22	32,500	70.19
56	38	41,250	85.00	35	44,650	99.42	33	47,800	98.28	31	51,400	88.54	29	55,450	70.19
57	41	37,800	85.00	38	41,250	98.24	35	45,600	98.52	32	49,750	85.84	30	53,400	70.19
58	24	12,500	85.00	22	13,600	97.63	20	15,250	99.37	18	16,800	90.23	16	19,450	70.19
59	27	34,600	85.00	24	37,500	98.85	22	41,250	99.37	19	46,750	87.09	17	50,750	70.19
60	31	28,500	85.00	29	30,500	95.97	27	33,250	99.98	25	38,000	97.02	21	43,800	70.19

Table 2
MOABCDE-TCQT parameter settings.

Input parameters	Notation	Setting
Number of decision variables	D	60
Population size	NP	300
limit	l	D
Crossover probability	CR	$0.5 \sim 0.9$
Scaling factor	F	0.5
Maximum generation	G_{max}	500

5.1. Optimization result of MOABCDE-TCQT

Table 2 shows parameter settings for the proposed MOABCDE-TCQT [18,19,51,52]. Thirty independent optimization runs were conducted to avoid randomness. Table 3 lists the first 12 non-dominated solutions in descending order of time, cost, quality, and compromised, respectively, along with optimal execution method combinations. Solutions 1, 2, and 3 generated the smallest project duration value, solution 4 generated the smallest values for cost,

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

No.	Partial Set	Alternatives	Project performance		
			Time (days)	Cost ($)	Quality (%)
1	Sorted by time	[2.4.4.3.4.4.1.4.5.5.1.5.1.4.1.3.4.4.1.4.1.5.1.5.5.5.5.1.5.1. 5.1.5.1.5.1.5.1.5.1.4.1.2.5.1.2.1.1.5. 1.1.1.1.5.1.1.5.1.1.5]	492	4,607,880	78.487
2		[3.4.3.3.2.4.2.4.4.5.5.5.3.4.2.5.4.3.2.3.3.5.3.4.5.3.5.3.5.2. 5.2.5.3.5.2.5.2.2.2.4.3.2.5.2.3.2.2.5. 2.3.1.3.5.2.2.5.3.3.5]	492	4,752,390	85.933
3		[2.4.3.3.3.4.1.4.3.5.5.5.1.4.1.5.4.1.1.4.1.5.1.5.5.5.5.2.5.1. 5.2.2.1.5.1.5.1.5.1.4.1.1.5.1.1.1.1.5. 1.4.1.2.5.1.1.5.2.1.5]	492	4,611,380	80.099
4	Sorted by cost	[3.4.2.3.1.4.2.4.5.5.3.5.2.4.1.5.4.4.1.4.2.5.2.4.5.4.5.1.5.1. 5.2.3.1.5.1.5.2.5.1.4.1.2.5.1.2.1.1.5. 1.3.1.1.5.3.1.5.3.1.5]	683	3,787,870	85.000
5		[2.4.3.3.3.4.1.4.3.5.5.5.1.4.1.5.4.1.1.4.1.5.1.5.5.5.5.2.5.1. 5.2.2.1.5.1.5.1.5.1.4.1.1.5.1.1.1.1.5. 1.4.1.2.5.1.1.5.2.1.5]	681	3,798,020	85.833
6		[3.4.4.3.2.4.2.4.5.5.5.5.2.4.1.3.4.2.2.3.2.5.1.5.5.5.5.3.5.1. 5.1.3.3.5.2.5.2.1.2.4.1.2.5.2.2.2.1.5. 1.5.1.2.5.2.1.5.3.1.5]	673	3,801,470	85.360
7	Sorted by quality	[3.4.2.3.1.4.2.4.5.5.3.5.2.4.1.5.4.4.1.4.2.5.2.4.5.4.5.1.5.1. 5.2.3.1.5.1.5.2.5.1.4.1.2.5.1.2.1.1.5. 1.3.1.1.5.3.1.5.3.1.5]	603	4,286,750	99.304
8		[2.4.4.3.4.4.1.4.5.5.1.5.1.4.1.3.4.4.1.4.1.5.1.5.5.5.5.1.5.1. 5.1.5.1.5.1.5.1.5.1.4.1.2.5.1.2.1.1.5. 1.1.1.1.5.1.1.5.1.1.5]	616	4,260,300	98.975
9		[2.4.3.3.3.4.1.4.3.5.5.5.1.4.1.5.4.1.1.4.1.5.1.5.5.5.5.2.5.1. 5.2.2.1.5.1.5.1.5.1.4.1.1.5.1.1.1.1.5. 1.4.1.2.5.1.1.5.2.1.5]	586	4,326,050	98.866
10	Compromised	[2.4.4.3.3.4.2.4.5.5.3.5.2.4.2.3.4.2.2.4.3.5.2.5.4.5.5.3.5.2. 5.2.2.3.5.3.5.2.4.3.4.2.2.5.3.3.2.3.5. 2.4.3.3.5.3.2.5.2.2.5]	495	4,757,240	86.177
11		[3.1.2.3.2.3.2.4.2.4.2.4.2.1.2.1.3.1.1.2.2.4.2.4.5.4.1.2.4.2. 1.2.1.1.4.1.5.1.2.3.3.2.2.3.2.2.2.3.5. 1.4.2.2.4.2.1.5.4.2.5]	532	4,409,675	90.081
12		[2.2.2.2.2.2.2.2.3.4.1.3.1.2.2.1.2.2.2.2.1.4.2.1.3.3.2.2.2.1. 2.2.3.2.3.2.3.2.2.2.3.3.2.2.2.2.1.3.4. 1.1.2.1.3.1.1.5.3.1.2]	578	4,229,600	94.822

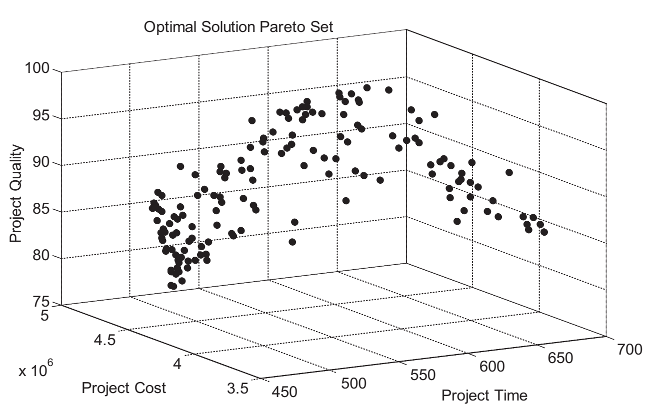

Fig. 5. Time–cost–quality tradeoff Pareto front using MOABCDE.

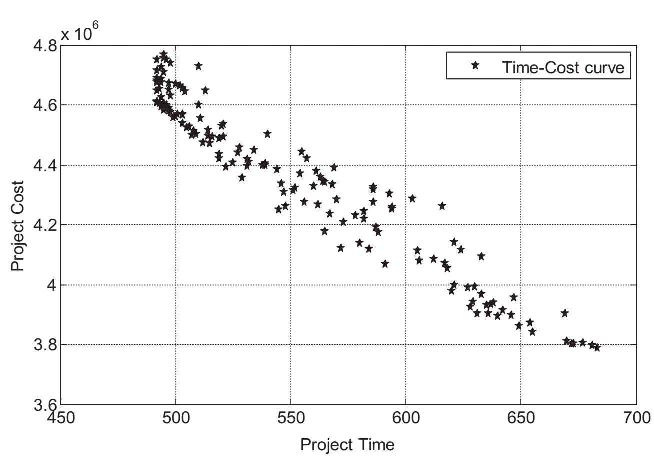

Fig. 6. Time–cost tradeoff analysis.

solution 7 generated the highest values for project quality, and the other solutions struck a balance among the three objectives. Project managers may select the optimal solution for a specific project scenario based on experience, preferences, and specific situation conditions. For instance, if a manager prioritizes time, then solutions 1, 2, and 3 are optimal. If a manager prioritizes budget, then solution 4 is optimal. Solution 7 is optimal if a manager prioritizes project quality. However, if a manager wants to strike a measured balance between the three objectives, then solution 11, for example, provides a centrist solution marked by acceptable project duration (532 days), moderate cost (US$ 4,409,675), and moderate project quality (90.081). Fig. 5 shows the typical Pareto optimal fronts obtained using the MOABCDE for this case study. These fronts show 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.

The non-dominated solutions may also be used to optimize tradeoffs between any two objectives on a two-dimensional plane. Figs. 6–8 show the relationship between time and cost, cost and quality, and time and quality, respectively. As shown in the time–cost curve example (Fig. 6), the lower project funding we spend on project correlates with the longer project duration we need to complete project and vice versa. However, Figs. 6–8 may not adequately represent 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 there is a hidden dimension that makes these points non-dominated.

5.2. Statistical comparison and analysis

We compared MOABCDE performance against NSGA-II [30], MOPSO [4], MOABC and MODE [51] to assess comparative effectiveness. For comparison purposes, all five algorithms used an equal number of function evaluations, had a population size of

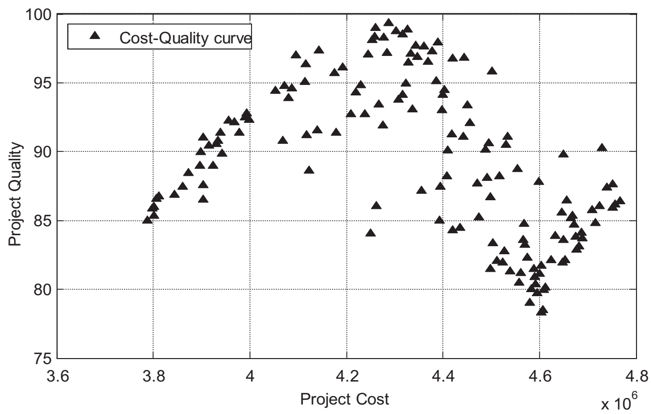

Fig. 7. Cost–quality tradeoff analysis.

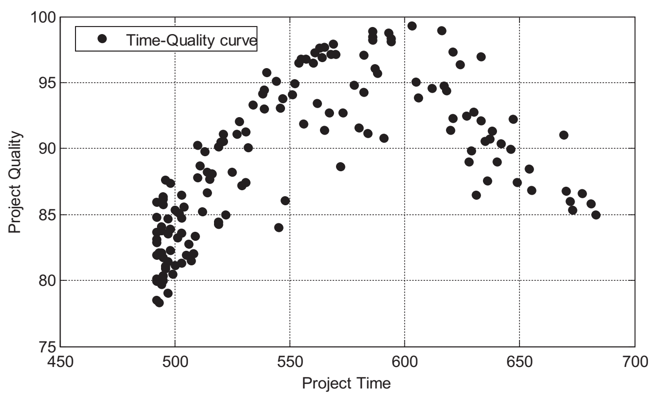

Fig. 8. Time–quality tradeoff analysis.

300 and 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 0.3–0.7. MOABC, MODE and MOABCDE control parameters remained the same, as stated previously in Table 2. Thirty independent runs were carried out for all experiments.

Multi-objective optimization problem performance measures are more complex than those of single-objective optimization problems. Three 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 [52]. In the literature, the researchers have suggested numerous quality indicators [30,53–55]. These indicators may be classified into three categories based on whether they evaluate: (1) closeness to the Pareto front; (2) the diversity in obtained solutions; or (3) both (1) and (2) [54]. The following describes the three quality indicators used in this research to evaluate, respectively, each of the three issues:

1. C-metric (C): C-metric is often used to assess the quality of the true Pareto front of optimized problems [56]. 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]:

$$C(S_1, S_2) = \frac{|\{a_2 \in S_2;\ \exists a_1 \in S_1 : a_1 \leqslant a_2\}|}{|S_2|} \tag{19}$$

The numerator in Eq. (19) denotes that the number of solutions in S_2 is dominated by at least one solution in S_1, and the denominator equals 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 the C-metric is not symmetrical in its arguments [57]. Table 4 illustrates the comparative results among five algorithms in terms of the C-metric, where A_1, A_2, A_3, A_4 and A_5 indicate MOABCDE, MODE, MOABC, MOPSO, and NSGA-II, respectively. Results show that MOABCDE dominates more than 16.1% of the MODE solutions, 10.5% of the MOABC solutions, 71.1% of the MOPSO solutions, and 59.4% of the NSGA-II solutions on average.

2. Spread (SP): This indicator [50] measures the extent of spread achieved among the non-dominated solutions. The mathematical definition of SP may be given as:

$$SP = \frac{\sum_{i=1}^{k} d(E_i, \Omega) + \sum_{X\in\Omega} |d(X,\Omega) - \bar{d}|}{\sum_{i=1}^{k} d(E_i, \Omega) + (|\Omega| - k)\bar{d}} \tag{20}$$

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 that all members of the Pareto optimal set are spaced equidistantly. A smaller value of SP indicates a better distribution and diversity of non-dominated solutions. Table 5 shows a comparison of the spread metric for

Table 5
Comparison of SP-metric for different algorithms.

	MOABCDE	MODE	MOABC	MOPSO	NSGA-II
Best	0.4577	0.4555	0.4176	0.4424	0.4798
Worst	0.5727	0.6368	0.7032	0.9604	1.0583
Average	0.5075	0.5374	0.5341	0.7103	0.6915
Std.	0.0325	0.0479	0.0638	0.1600	0.1601

Table 6
Comparison of HV-metric for different algorithms.

	MOABCDE	MODE	MOABC	MOPSO	NSGA-II
Best	0.9593	0.8988	0.8988	0.9782	0.9782
Worst	0.7203	0.7071	0.3460	0.4000	0.2953
Average	0.8122	0.7460	0.5715	0.7560	0.5511
Std.	0.0509	0.0520	0.1947	0.1434	0.2000

Table 4
Comparison of C-metric for different algorithms.

	C(A1,A2)	C(A2,A1)	C(A1,A3)	C(A3,A1)	C(A1,A4)	C(A4,A1)	C(A1,A5)	C(A5,A1)
Best	0.268	0.192	0.320	0.152	1.000	0.192	1.000	0.179
Worst	0.050	0.000	0.000	0.000	0.091	0.000	0.020	0.000
Average	0.161	0.065	0.105	0.033	0.711	0.036	0.594	0.034
Std.	0.055	0.052	0.094	0.041	0.388	0.056	0.420	0.055

Table 7
Hypothesis test results between MOABCDE and other approaches.

Indicators	$\bar{x}_1$	$\bar{x}_2$	s_1	s_2	t	v	$t_{\alpha;v} = t_{0.05;v}$
C-metric (C)	0.1050	0.0334	0.0938	0.0406	3.837	40	1.684
Spread (SP)	0.5075	0.5341	0.0325	0.0638	−2.035	43	−1.681
Hyper-volume (HV)	0.8122	0.7560	0.0509	0.1434	2.022	36	1.688

different algorithms. This supports that the average performance of the MOABCDE is superior to that of the four other algorithms.

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 [52,54]. A hypercube v_i is constructed for each solution $X_i \in \Omega$, with reference point W and the solution X_i as the diagonal corners of the hypercube. The reference point may be found simply by constructing a vector of worst objective function values. Thereafter, a union of all hypercubes is found, with the HV of this union calculated as:

$$HV = \bigcup_{i=1}^{|\Omega|} v_i \tag{21}$$

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 6 lists the results for each of the four compared algorithms in terms of HV. From Table 6, we see that the proposed model obtains the largest HV values, which means that the MOABCDE has better convergence and diversity performance than the other four algorithms.

5.3. Hypothesis test

A hypothesis test was performed to further demonstrate the superiority of the MOABCDE over the other approaches. In all indicators, the hypothesis tests only considered the MOABCDE and the best of other approaches. A one-tailed t-test with equal sample sizes and unequal and unknown variances analyzed the following hypothesis tests:

Hypothesis: MOABCDE versus standard MOABC in term of C-metric (Table 4).

H_0: There is no difference in the C-metric of the MOABCDE algorithm and that of the MOABC algorithm.

H_1: The MOABCDE algorithm is significantly better than the MOABC algorithm.

MOABCDE $s_1 = 0.094$; MOABC: $s_2 = 0.041$; $n_1 = n_2 = n = 30$;

$$v = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{\frac{(s_1^2/n_1)^2}{n_1-1} + \frac{(s_2^2/n_2)^2}{n_2-1}} = \frac{\left(0.094^2/30 + 0.041^2/30\right)^2}{\frac{(0.094^2/30)^2}{30-1} + \frac{(0.041^2/30)^2}{30-1}}$$
$$= 39.5 \text{ (closest to 40)}$$

Critical value: with significant level of t-test $\alpha = 0.05$; $v = 40$; we have $t_{\alpha;v} = t_{0.05;40} = 1.684$

$$\text{Statistical test}: \ t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} = \frac{(0.1050 - 0.0334)}{\sqrt{0.094^2/30 + 0.041^2/30}}$$
$$= 3.837 > 1.684 = t_{0.05;40}$$

where n is the sample size (number of experimental runs), v is the degrees of freedom used in the test, s_1^2 and s_2^2 are the unbiased estimators of the variances of the two samples (MOABCDE and MOABC). The denominator of t is the standard error of the difference between two means $\bar{x}_1, \bar{x}_2$ (average).

H_0 is rejected because the statistical test value noted above is greater than the critical value, which demonstrates the proposed MOABCDE as statistically superior to the standard MOABC in terms of the C-metric. In the same manner, Table 7 shows the results of the hypothesis test between MOABCDE and the best of other approaches in terms of the C-metric (C), Spread (SP) and Hyper-volume (HV):

As shown in Table 7, the proposed algorithm MOABCDE yielded results that were significantly better than other approaches in terms of the C-metric, spread, and hyper-volume ($t = 3.837 > 1.684 = t_{\alpha;v}$; $t = -2.0348 < -1681 = -t_{\alpha;v}$ and $t = 2.022 > 1.688 = t_{\alpha;\,v}$).

6. Conclusions and further study

Time, cost, and quality are important and interdependent variables in construction projects. Integrating all goals into the optimization process and pursuing the tradeoffs among these goals represent one approach to improving the overall efficiency and effectiveness of construction projects. This paper developed a hybrid algorithm MOABCDE to solve the TCQT problem for construction projects. This hybrid algorithm makes two important contributions: First, it significantly improves MOABCDE process-convergence speed and solution accuracy by using DE operators in the MOABC scheme. Second, the MOABCDE is more effective and efficient than current, widely used multi-objective evolutionary algorithms, as demonstrated in a numerical construction case study. MOABCDE outperformed the non-dominated sorting genetic algorithm, the multiple objective particle swarm optimization, the multiple objective differential evolution, and the multiple objective artificial bee colony in terms of diversity of characteristics, compromise solutions, and degree of satisfaction.

Results show that the proposed model MOABCDE generates a better Pareto front than current, widely used approaches. The Pareto front generated by MOABCDE provides information that helps construction-project decision makers determine the optimal tradeoff among the three important project considerations of project duration, cost, and quality.

The proposed hybrid multiple objective artificial bee colony with differential evolution is simple, robust, and efficient. It does not impose any limitation on the number of objectives and may be extended to include additional objectives. Further minor modifications of the proposed MOABCDE algorithm hold interesting potential to resolve other multi-objective optimization problems in the field of construction management such as the tradeoffs among performance, cost, and reliability in engineering design work and resource-constrained and resource-leveling in project scheduling activities.

The case study considered quantifiable project performance variables, including time, cost, and, particularly, quality. However, in practice, experts, contractors, engineers, and managers often evaluate performance using linguistic and other imprecise terms due to uncertainties in the environment and subjectivity. Therefore, further study is required to build an optimization model to solve the time–cost–quality tradeoff, which considers aspects of performance that are vague, uncertain, and imprecise. Integrating the current model with other techniques such as fuzzy and stochastic simulation are interesting directions for future research.

References

[1] B. Zahraie, M. Tavakolan, Stochastic time–cost–resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets, J. Construct. Eng. Manage. 135 (2009) 1162–1171.
[2] L. Feng, Burns Liu, Using genetic algorithms to solve construction time–cost trade-off problems, J. Comput. Civil Eng. 11 (1997) 184–189.
[3] T. Hegazy, Optimization of construction time–cost trade-off analysis using genetic algorithms, Can. J. Civil Eng. 26 (1999) 685–697.
[4] I. Yang, Using elitist particle swarm optimization to facilitate bicriterion time–cost trade-off analysis, J. Construct. Eng. Manage. 133 (2007) 498–505.
[5] Y. Xiong, Y. Kuang, Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off, J. Construct. Eng. Manage. 134 (2008) 153–156.
[6] Z. Geem, Multiobjective optimization of time–cost trade-off using harmony search, J. Construct. Eng. Manage. 136 (2010) 711–716.
[7] H. Zhang, F. Xing, Fuzzy-multi-objective particle swarm optimization for time–cost–quality tradeoff in construction, Autom. Construct. 19 (2010) 1067–1075.
[8] W. Hu, X. He, An Innovative time–cost–quality tradeoff modeling of building construction project based on resource allocation, Scient. World J. 2014 (2014) 10.
[9] A.J.G. Babu, N. Suresh, Project management with time, cost, and quality considerations, Euro. J. Operat. Res. 88 (1996) 320–327.
[10] D.B. Khang, Y.M. Myint, Time, cost and quality trade-off in project management: a case study, Int. J. Project Manage. 17 (1999) 249–256.
[11] K. El-Rayes, A. Kandil, Time–cost–quality trade-off analysis for highway construction, J. Construct. Eng. Manage. 131 (2005) 477–486.
[12] L. Zhang, J. Du, S. Zhang, Solution to the time–cost–quality trade-off problem in construction projects based on immune genetic particle swarm optimization, J. Manage. Eng. 30 (2014) 163–172.
[13] S. Mungle, L. Benyoucef, Y.-J. Son, M.K. Tiwari, A fuzzy clustering-based genetic algorithm approach for time–cost–quality trade-off problems: a case study of highway construction project, Eng. Appl. Artif. Intell. 26 (2013) 1953–1966.
[14] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (2008) 687–697.
[15] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optimiz. 39 (2007) 459–471.
[16] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[17] R. Akbari, R. Hedayatzadeh, K. Ziarati, B. Hassanizadeh, A multi-objective artificial bee colony algorithm, Swarm Evolution. Comput. 2 (2012) 39–52.
[18] Y. Li, Y. Wang, B. Li, A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow, Int. J. Electr. Power Energy Syst. 52 (2013) 25–33.
[19] R.M. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optimiz. 11 (1997) 341–359.
[20] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evolution. Comput. 15 (2011) 4–31.
[21] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evolution. Comput. 1 (2011) 32–49.
[22] W. Gao, S. Liu, Improved artificial bee colony algorithm for global optimization, Inform. Process. Lett. 111 (2011) 871–882.
[23] X. Li, M. Yin, Hybrid differential evolution with biogeography-based optimization for design of a reconfigurable antenna array with discrete phase shifters, Int. J. Anten. Propag. 2011 (2011) 12.
[24] X. Li, M. Yin, Hybrid differential evolution with artificial bee colony and its application for design of a reconfigurable antenna array with discrete phase shifters, IET Microwav., Anten. Propag. 6 (2012) 1573–1582.
[25] V. Plevris, M. Papadrakakis, A hybrid particle swarm—gradient algorithm for global structural optimization, Comp.-Aided Civil Infrastruct. Eng. 26 (2011) 48–68.
[26] S. Kumar, V.K. Sharma, R. Kumari, A novel hybrid crossover based artificial bee colony algorithm for optimization problem, Int. J. Comp. Appl. 82 (2013) 18–25.
[27] D. González-Álvarez, M. Vega-Rodríguez, Hybrid multiobjective artificial bee colony with differential evolution applied to Motif finding, in: L. Vanneschi, W. Bush, M. Giacobini (Eds.), Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer, Berlin Heidelberg, 2013, pp. 68–79.
[28] A. Rajasekhar, A. Abraham, M. Pant, A hybrid differential artificial bee colony algorithm based tuning of fractional order controller for permanent magnet synchronous motor drive, Int. J. Mach. Learn. Cybernet. (2012) 1–11.
[29] J. Lu, G. Zhang, D. Ruan, Multi-objective Group Decision Making: Methods, Software and Applications With Fuzzy Set Techniques, Imperial College Press, 2007.
[30] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolution. Comput. 6 (2002) 182–197.
[31] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidisc. Optimiz. 26 (2004) 369–395.
[32] C. Min-Yuan, T. Duc-Hoc, Two-phase differential evolution for the multiobjective optimization of time–cost tradeoffs in resource-constrained construction projects, IEEE Trans. Eng. Manage. 61 (2014) 450–461.
[33] M. Cheng, D. Tran, Opposition-based multiple-objective differential evolution to solve the time–cost–environment impact trade-off problem in construction projects, J. Comput. Civil Eng. 0 (2014) 04014074.
[34] D. Lei, Multi-objective production scheduling: a survey, Int. J. Adv. Manuf. Technol. 43 (2009) 926–938.
[35] F. Ramezani, J. Lu, F. Hussain, Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization, in: S. Basu, C. Pautasso, L. Zhang, X. Fu (Eds.), Service-Oriented Computing, Springer, Berlin Heidelberg, 2013, pp. 237–251.
[36] M.G. Castillo Tapia, C.A. Coello Coello, Applications of multi-objective evolutionary algorithms in economics and finance: a survey, in: IEEE Congress on Evolutionary Computation, 2007, CEC 2007, 2007, pp. 532–539.
[37] K.V. Price, R.M. Storn, J.A. Lampinen, Differential Evolution A Practical Approach to Global Optimization, Springer-Verlag, Berlin Heidelberg, 2005.
[38] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Repport-TR06, Erciyes University, Kayseri/Turkey, 2005.
[39] A.L.a. Bolaji, A.T. Khader, M.A. Al-Betar, M.A. Awadallah, Artificial bee colony algorithm, its variants and applications: a survey, J. Theoret. Appl. Inform. Technol. 47 (2013) 434–459.
[40] H. Talbi, M. Batouche, Hybrid particle swarm with differential evolution for multimodal image registration, in: IEEE International Conference on Industrial Technology, 2004, IEEE ICIT '04, 2004, vol. 1563, 2004, pp. 1567–1572.
[41] C. Zhang, J. Ning, S. Lu, D. Ouyang, T. Ding, A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization, Operat. Res. Lett. 37 (2009) 117–122.
[42] S.N. Omkar, J. Senthilnath, R. Khandelwal, G. Narayana Naik, S. Gopalakrishnan, Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures, Appl. Soft Comput. 11 (2011) 489–499.
[43] F. Kang, J. Li, Z. Ma, Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions, Inform. Sci. 181 (2011) 3508–3531.
[44] M.S. Kiran, M. Gündüz, A novel artificial bee colony-based algorithm for solving the numerical optimization problems, Int. J. Innov. Comput., Inform. Control 8 (2012) 6107–6122.
[45] A. Rajasekhar, M. Pant, A. Abraham, A hybrid differential artificial bee algorithm based tuning of fractional order controller for PMSM drive, in: Third World Congress on Nature and Biologically Inspired Computing (NaBIC), 2011, 2011, pp. 1–6.
[46] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (2010) 3166–3173.
[47] M.-Y. Cheng, D.-H. Tran, Y.-W. Wu, Using a fuzzy clustering chaotic-based differential evolution with serial method to solve resource-constrained project scheduling problems, Autom. Construct. 37 (2014) 88–97.
[48] N. Noman, H. Iba, Accelerating differential evolution using an adaptive local search, IEEE Trans. Evolution. Comput. 12 (2008) 107–125.
[49] M.S. Kıran, M. Gündüz, A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems, Appl. Soft Comput. 13 (2013) 2188–2203.
[50] Y.-N. Wang, L.-H. Wu, X.-F. Yuan, Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure, Soft Comput. – Fus. Found., Methodol. Appl. 14 (2010) 193–209.
[51] M. Ali, P. Siarry, M. Pant, An efficient differential evolution based algorithm for solving multi-objective optimization problems, Euro. J. Operat. Res. 217 (2012) 404–416.
[52] L.H. Wu, Y.N. Wang, X.F. Yuan, S.W. Zhou, Environmental/economic power dispatch problem using multi-objective differential evolution algorithm, Electr. Power Syst. Res. 80 (2010) 1171–1181.
[53] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput. 3 (1995) 1–16.
[54] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G.d. Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, Trans. Evol. Comp 7 (2003) 117–132.
[55] C.A. Coello Coello, Evolutionary multi-objective optimization: a historical view of the field, IEEE Comput. Intell. Magaz. 1 (2006) 28–36.
[56] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evolution. Comput. 3 (1999) 257–271.
[57] L. Wang, C. Singh, Reserve-constrained multiarea environmental/economic dispatch based on particle swarm optimization with local search, Eng. Appl. Artif. Intell. 22 (2009) 298–307.