Journal of Civil Engineering and Management
ISSN 1392-3730 / eISSN 1822-3605
2019 Volume 25 Issue 4: 322–339
https://doi.org/10.3846/jcem.2019.9681

MULTI-OBJECTIVE SYMBIOTIC ORGANISMS OPTIMIZATION FOR MAKING TIME-COST TRADEOFFS IN REPETITIVE PROJECT SCHEDULING PROBLEM

Duc-Hoc TRAN[1*], Jui-Sheng CHOU[2], Duc-Long LUONG[1]

[1]Department of Construction Engineering and Management, Ho Chi Minh City University of Technology, Vietnam National University Ho Chi Minh, City (VNU-HCM), Ho Chi Minh City, Viet Nam
[2]Department of Civil and Construction Engineering National Taiwan University of Science and Technology, Taipei, Taiwan

Received 13 October 2018; accepted 25 February 2019

Abstract. Time-cost problems that arise in repetitive construction projects are commonly encountered in project scheduling. Numerous time-cost trade-off approaches, such as mathematical, metaheuristic, and evolutionary methods, have been extensively studied in the construction community. Currently, the scheduling of a repetitive project is conducted using the traditional precedence diagramming method (PDM), which has two fundamental limitations: (1) progress is assumed to be linear from start to finish; and (2) activities in the schedule are connected each other only at the end points. This paper proposes a scheduling method that allows the use of continuous precedence relationships and piece-wise linear and nonlinear activity-time-production functions that are described by the use of singularity functions. This work further develops an adaptive multiple objective symbiotic organisms search (AMOSOS) algorithm that modifies benefit factors in the basic SOS to balance exploration and exploitation processes. Two case studies of its application are analyzed to validate the scheduling method, as well as to demonstrate the capabilities of AMOSOS in generating solutions that optimally trade-off minimizing project time with minimizing the cost of non-unit repetitive projects. The results thus obtained indicate that the proposed model is feasible and effective relative to the basic SOS algorithm and other state-of-the-art algorithms.

Keywords: scheduling, singularity functions, time-cost trade-offs, repetitive project, multiple objective, optimization.

Introduction

Construction scheduling is one of the most important aspects of construction management. The time-cost trade-off (TCT) optimization problem is a cornerstone of construction scheduling and planning efforts because it seeks to identify construction method options that balance these two conflicting considerations (Koo, Hong, & Kim, 2015; Lim, Jang, Choi, & Lee, 2015). Typically, projects of a shorter duration have higher direct cost. A construction company may have a competitive advantage over its rivals if it can minimize both project time and cost simultaneously (Tran & Cheng, 2014; Cheng, Tran, & Cao, 2013).

In the original TCT optimization problems, the project duration can often be compressed by accelerating some of its activities at an additional expense (Hegazy, 1999; Kelley & Walker, 1959). From the perspective of practical applicability, the original TCT still faces some challenges when applying to real case study. Over the past decades, the vast majority of its extensions deal with the possible shortenings and their effects on the project duration while activity stretching has the same importance in decreasing the project cost as well, especially in the case of the TCT problems associated with repetitive projects.

Specifically, the aforementioned optimization involves: (1) minimizing the direct cost of a project that is completed before a specified deadline (deadline problem) (Khalied & Khaled, 2006; Long & Ohsato, 2009); (2) finding the shortest project duration without exceeding a given budget (budget problem) (Hegazy & Nagib, 2001); (3) combining multiple objectives in one objective and thereby providing a single solution to the optimization problem (Ipsilandis, 2007); or (4) generating a complete and efficient set of optimal time cost trade-off solutions called Pareto front (time-cost curve problem) (Feng, Liu, & Burns, 1997). The multiple objective algorithms are used to optimize both

*Corresponding author. E-mail: *tdhoc@hcmut.edu.vn*

objectives simultaneously in a single run (Hyari, El-Rayes, & El-Mashaleh, 2009; Liberatore & Pollack-Johnson, 2013; Huang, Zou, & Zhang, 2016).

Many methodologies have been proposed in the extensive subject of project scheduling since the development of the critical path method (CPM). The original solution procedures for the time cost trade-off were proposed by Kelley and Walker (1959) using parametric linear programming. Techniques are categorized into heuristic methods, and mathematical programming models (Feng et al., 1997). At the beginning, the mathematical approaches were used to solve the time-cost tradeoff problems because they can directly yield an optimum value; otherwise all solutions need to be enumerated. In general, methods using linear and/or integer programming lead to optimal solutions whereas such approach can be of great computational effort (Chassiakos & Sakellaropoulos, 2005).

Other researchers attempted to utilize heuristic methods in solving the time-cost tradeoff problems (Hazır, Erel, & Günalay, 2011; Li & Love, 1997). Heuristic approaches, however, operate on rules of thumb and lack rigor (Zheng, Ng, & Kumaraswamy, 2005). They can only deliver good feasible solutions and by no means guarantee an optimum solution (Shannon & Lucko, 2012).

Owing to the limitations of mathematical and heuristic approaches, such as their inability to deal with non-linear activities or handle more than one objective (Zheng et al., 2005), metaheuristic methods were designed. Evolutionary Algorithms (EAs), known as a type of metaheuristic optimization for solving time cost tradeoff problems, have drawn more attention in recent years. The advantage of EAs is its ability to find optimal solutions to complex problems in a relatively short time (Agdas, Warne, Osio-Norgaard, & Masters, 2018).

Noticeably, the traditional precedence diagramming method (PDM)-based scheduling optimization focuses only on time without consideration of productivity issue (Haidu, 1996; Photios & Yang, 2016). An extensive review of the literature done in this study also found that the existing algorithms do not handle recent developments of traditional PDM well, e.g., continuous relationships and piecewise linear and non-linear activities, shedding a light of an urgent research need.

Repetitive projects (RPs) are common in the construction sector. Repetition arises from geometry and location layouts or the multiplication of units (Cho, Hong, & Hyun, 2013; Srisuwanrat & Ioannou, 2007; Zhang, 2015). RPs often require resources (such as crews) to perform the same task in various units (locations, segments) by moving from one unit to the next (Vanhoucke, 2006).

Several repetitive scheduling methods (RSMs) have been proposed for the planning and scheduling of repetitive construction projects (Huang & Sun, 2005; Jeeno, Brijesh, Dileeplal, & Tinjumol, 2016; Khalied & Khaled, 2006; Photios & Yang, 2016). The general consensus is that RSMs are simple and easily applied scheduling methodology that follows naturally from the concepts and relationships found in the PDM (Harris & Ioannou, 1998). RSMs address the need for work continuity and uninterrupted resource deployment in the construction of a repetitive project. Consequently, RSMs are preferred for the scheduling and resource planning of repetitive construction projects (Huang & Sun, 2006).

The currently used RSMs on the basis of the PDM have two fundamental limitations on precedence relationships: (1) it assumes that progress is linear from start to finish; (2) activities in the schedule are connected each other only at the end points. The first basic assumption on precedence relationships of the PDM is that activities are linear, progressing at a fixed production speed without planned breaks. In practice, this restrictive assumption is almost always inaccurate (Hajdu, Lucko, & Yi, 2017). For example, in a trench construction project, the production speed if it is measured along the street accelerates if the depth of the trench is decreasing and slows down if the depth of the trench is increasing.

Hence, the relationship between duration of activity and quantity of work is non-linear (Figure 1a). With respect to the second basic assumption of the PDM, Figure 1b shows a relationship between activity A (linear) and activity B (non-linear), violating end-point relations. Generating a more competitive schedule requires consideration of continuous precedence relationships; piece-wise linear and nonlinear activity-time-production functions that are described by the use of singularity functions (Su & Lucko, 2016), presenting a serious challenge to the planner in facilitating the time-cost tradeoff.

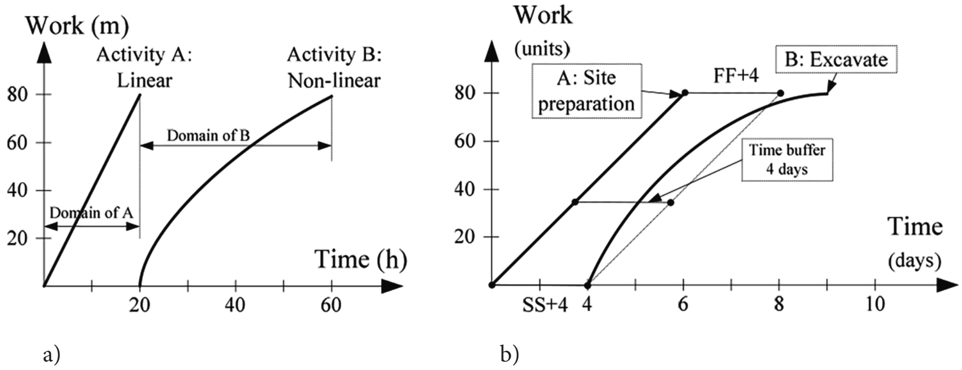


Figure 1. Activity function in trench project

Cheng and Prayogo (2014) presented a powerful optimization algorithm, called symbiotic organisms search (SOS) for solving structural engineering problems. SOS has only two common controlling parameters (population size and the maximum number of function evaluations), making it robust and generalizable. Many researchers have proved that SOS outperforms such well-known algorithms (Abdullahi & Ngadi, 2016; Cheng, Prayogo, & Tran, 2016; Tejani, Savsani, & Patel, 2016; Verma, Saha, & Mukherjee, 2017) as artificial bee colony (ABC), differential evolution (DE), particle swarm optimization (PSO), and genetic algorithms (GAs) in dealing with both benchmarked functions and engineering problems such as scheduling (Abdullahi, Ngadi, & Abdulhamid, 2016; Cheng et al., 2016; Ezugwu & Adewumi, 2017), structural design (Tejani et al., 2016; Tejani, Savsani, Bureerat, & Patel, 2018), and power flow problems (Duman, 2017).

Motivated by these advantages of SOS, several researchers have extended the SOS algorithm to multiple objectives. The fact that multiple objective symbiotic organisms search (MOSOS) is more powerful than other widely used multiple objective algorithms has been demonstrated (Panda & Pani, 2016; Tran, Cheng, & Prayogo, 2016). Therefore, this study develops a new model that is based on the SOS algorithm for facilitating the time-cost tradeoff for repetitive projects by considering continuous precedence relationships; piece-wise linear and nonlinear activity-time-production functions via singularity functions.

The contributions of this study are as follows: (1) developing a new numerical method for scheduling repetitive projects by allowing continuous precedence relationships and piece-wise linear and nonlinear activity-time-production functions; (2) introducing an adaptive multiple objective symbiotic organisms search algorithm to handle multi-objective problems simultaneously in a single run; and (3) facilitating the solving of time-cost tradeoff problems. This paper is organized as follows. First, literature that is related to the proposed optimization model is reviewed. Then, a model for solving TCT problems with a singularity is presented. This new model is demonstrated using a numerical example and relevant statistical results. The final section draws conclusions and provides suggestions for future research.

1. Literature review

1.1. Methods of project scheduling

The most widely used network technique for establishing construction project schedules, the precedence diagram method (PDM), has existed for almost six decades (Fondahl, 1961; Roy, 1959). PDM has become the prevalent technique of our time due to the flexibility provided by its different precedence relations and easy drawing by using the activity-on-node notation. In the PDM, the start or end of an activity can be connected to other activity in a schedule, allowing for the use of three precedence relations between project activities – the minimal start-to-start (SS), finish-to-finish (FF), and start-to-finish (SF) – in addition to the traditional finish-to-start relationship (Fondahl, 1987; Kelley & Walker, 1959).

Several researchers have criticized these strict endpoint relations and several methods that explicitly allow point-wise links to emerge from anywhere on a predecessor to anywhere on a successor have been developed (Hajdu & Isaac, 2016). Francis and Miresco (2002) proposed the chronographic method, which allows the predecessor and successor to have several new relationships in order to allow a superior monitoring of the project execution. Kim (2012) developed the beeline diagramming method as a new networking technique that captures all overlaps among activities. de Leon (2008) connected internal points, called embedded nodes, to each other, and called his method the "graphical diagramming method".

All of these methods are based on the same concept; only the terminology and definitions differ. Hajdu (2015b) studied descriptions of point-to-point relations using standardized nomenclature, formulas to expand the CPM, and an algorithm for generating both minimal and maximal relationships. Point-to-point relations are practically acceptable, but they cannot support a comprehensive model because they do not allow for the control of all points that are associated with the connected activities.

Drawing on the literature, Hajdu (2015a) was the first to describe continuous precedence relations that perfectly represent the overlapping of activities in network schedules. This technique applies a non-linear definition of activities to overcome the limitation in the original PDM, which assumes continuous linear activities only.

Research on the temporal logical relationships among activities in time-work diagrams has led to mathematical formulations thereof using singularity functions to minimize the duration of linear schedules and to determine criticality (Lucko, 2014). Lucko (2009) was the first to introduce the productivity scheduling method (PSM) using singularity functions as a powerful and flexible mathematical method for overcoming the inherent limitation of the linear scheduling method, which is difficult to computerize. The PSM formulates singularity function equations of activities and buffers. The difference between the extended PDM and extended PSM methods is that the PDM allows non-linear activities and continuous relationships. Moreover, the PDM uses stacking activities and their buffers to consolidate them in the final configuration with minimum differences of critical activities (Lucko, 2009). Lucko (2011) derived one flexible equation for solving the resource leveling problem by transposing and differentiating activity equations. Su and Lucko (2015b) used singularity functions to derive a new mathematical method within activities to model unbalanced bidding projects. Su and Lucko (2015a) optimized present value scheduling using a synthetic cash flow model with singularity functions.

More recently, Hajdu et al. (2017) derived an algorithm for identifying pairs of activities that are connected by a continuous relation and that may be nonlinear. Their work used singularity functions, temporarily stacked activities, the first derivative of singularity functions and the time gap between successor and predecessor and buffer techniques for dealing with continuous precedence relations and nonlinear activity-time-production functions in a schedule.

The method of Hajdu et al. (2017) used the first derivative of the singularity functions of the relationships among activities to identify the minimum time gap between the successor and predecessor buffers. Solving the first differential equation for complicated activities-based singularity functions is difficult. Singularity functions have been applied to linear schedules but never explicitly to multiple objective optimization problems in the field of construction.

This work develops a novel numerical method to obtain the relationship between predecessor and successor activities using a time or work buffer in the project network. The method will be applied to the time-cost tradeoff problem. It enables optimization of project scheduling with a wide range of production functions of construction activities (linear/ nonlinear singularity functions) and general relations (point-to-point or near continuous) in complex project environments.

1.2. Singularity functions

Singularity functions, which were first used in the structural analyses of beams under various loads, are being newly applied to construction scheduling (Lucko, 2009). Singularity functions are a generalization of traditional polynomial functions that involve a right-continuous operator. The general and continuous features of singularity functions satisfy the requirements of current research and make singularity functions effective for solving scheduling problems with nonlinear activities and continuous relations (Lucko, 2011; Isaac, Su, Lucko, & Dagan, 2017).

Lucko (2009) explicated the following advantages of singularity functions; they describe the phenomenon of interest geometrically; they separate the components of the phenomenon of interest; they capture any changes in progress across activity time and amount of work. This study uses a non-linear singularity, which (1) incorporates non-linear changes in progress with respect to time and amount (such as in the excavation of a trench that gradually deepens from begin to end) in repetitive projects; and (2) considers general relationships among activities in schedule for the better modeling of overlapping activities.

Equation (1) defines a single term that is used in a singularity function; a complete function includes one or more of such terms:

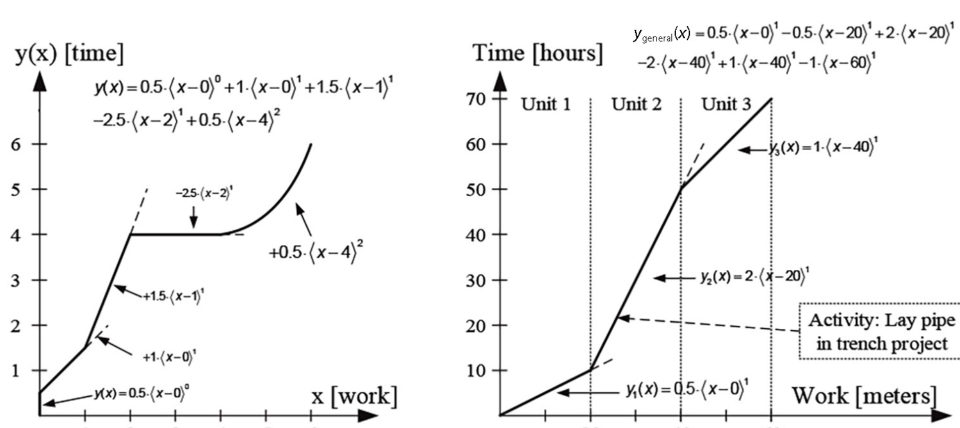
$$t(w) = k.\langle w-a\rangle^m = \begin{cases} 0 & if\ w < a \\ k.(w-a)^m & if\ w \ge a \end{cases}, \tag{1}$$

where w is the variable that is represented on the horizontal w-axis; a is the cutoff value on the w-axis at which the function becomes valid; m is the order of the exponent, and k is a scaling factor. Eqn (1) is zero for all $w < a$, and is evaluated normally for all $w \ge a$. The singularity function is cumulative: as w increases, more terms become active. If a term is active only from a_1 to a_2, then it must be added at a_1 and subtracted at a_2. By applying singularity function to linear schedules of construction projects, the traditional definition the activity production function is formulated as Eqn (2) (Lucko, 2009):

$$y(\mathrm{x}) = y_0.\langle x-0\rangle^0 + \frac{y_1-y_0}{x_1-x_0}.\langle x-0\rangle^1 + \sum_{k=1}^{m-1}\left[\left(\frac{y_{k+1}-y_k}{x_{k+1}-x_k} - \frac{y_k-y_{k-1}}{x_k-x_{k-1}}\right).\langle x-x_k\rangle^1\right], \tag{2}$$

where y is time variable of an activity with m units; x is amount variable; the pair of coordinates (x_k, y_k) corresponds to index k.

Figure 2a displays how multiple basic terms produce a shape that can be completely customized. Figure 2b presents the scheduling of an activity in a repetitive project with singularity functions.



a) Singularity function with multiple basic terms

b) Example of singularity function for an actual activity in repetitive project

Figure 2. Example of singularity functions

2. Time-cost in repetitive projects

A project consists of M activities, which may be repeated in U units. Each unit is modeled on an activity-on-node network, in which a set of M nodes represents M activities and their precedence relationships. This network is repeated in U units. Resources are required to execute each activity (i), which is repetitively executed in U units from unit 1 to unit U. The TCT problem for RPs requires that project planners determine the execution options of all activities (i) in U units for optimal scheduling while satisfying all project constraints. The problem of scheduling RPs must tradeoff two conflicting objectives, which are minimizing project duration and minimizing cost.

Objective 1: Minimize project duration $T_p \Rightarrow T_p^{Min}$, as defined in Eqn (3):

$$T_p = \min\left(\underset{\substack{i=1,\dots,M. \\ j=1,\dots,U.}}{Max} (FT_{i,j}) \right) = \min\left(\underset{\substack{i=1,\dots,M. \\ j=1,\dots,U.}}{Max} (ST_{i,j} + D_{i,j}) \right), \quad (3)$$

where $FT_{i,j}$ is the finish time of activity (i) in unit (j). The duration of each activity i at unit j is denoted by $D_{i,j}$. Notably, duration of activities can be changed between their normal and crash durations by selecting execution option.

Objective 2: Minimize total project cost $TC_p \Rightarrow TC_p^{Min}$, given by Eqn (4):

$$TC_p = C_D + C_I = \sum_{i=1}^{M}\sum_{j=1}^{U} c_{i,j} + C_o + b.T_P. \quad (4)$$

Indirect cost $C_I = C_o + b.T_P$ where T_P is the project duration and is determined by Eqn (3). The term b is the indirect cost per unit of project time. C_o is the total initial cost (which may include mobilization cost, the cost of temporary facilities, and other initial costs). Direct cost $C_D = \sum_{i=1}^{M}\sum_{j=1}^{U} c_{i,j}$, where $c_{i,j}$ is the direct cost of completing activity (i) in unit (j).

The scope of this study is limited to continuous use of resources between units. In repetitive construction projects, most activities require several resources to be employed together. However, only the most significant resource is assumed to be associated with an activity, and the same resource will be used for this activity in successive repeating units, so each resource must be consistent from a repetitive-unit to other repetitive-unit (Harris & Ioannou, 1998). This assumption is still valid for the development of the proposed method. Moreover, interruptions are not allowed between the start time ($ST_{i,j}$) and the finish time ($FT_{i,j}$ ($j = 1, \dots, U$).

3. Proposed model for time-cost trade-off

This section describes the adaptive multiple objective symbiotic organisms search for simultaneously optimizing project time and cost in repetitive projects with singularity functions. The proposed model is developed based on the original MOSOS algorithm (Tran et al., 2016). Figure 3 schematically depicts the adaptive MOSOS algorithm, which was written in the MATLAB programming language. The figure shows various stages of the adaptive algorithm, including initialization, the adaptive mutualism phase, the commensalism phase, the parasitism phase, and termination. The following subsections describe these phases in detail.

3.1. Initialization

This study considers the scheduling of repetitive projects with singularity functions, in which project duration and cost are simultaneously optimized. The inputs to the model include such information about the project as relationships among activities, the singularity function of the duration of each activity in each unit, and the activity cost in each unit $c_{i,j}$. The user must set the parameters of AMOSOS, including the size of the ecosystem *ecosize*, the number of decision variables D, the number of objective functions O, the maximum number of generations $G_{\max}$, and the lower bounds (LB) and upper bounds (UB) on the decision variables. Given these inputs, the optimizer automatically calculates an optimal set of execution options for all project activities.

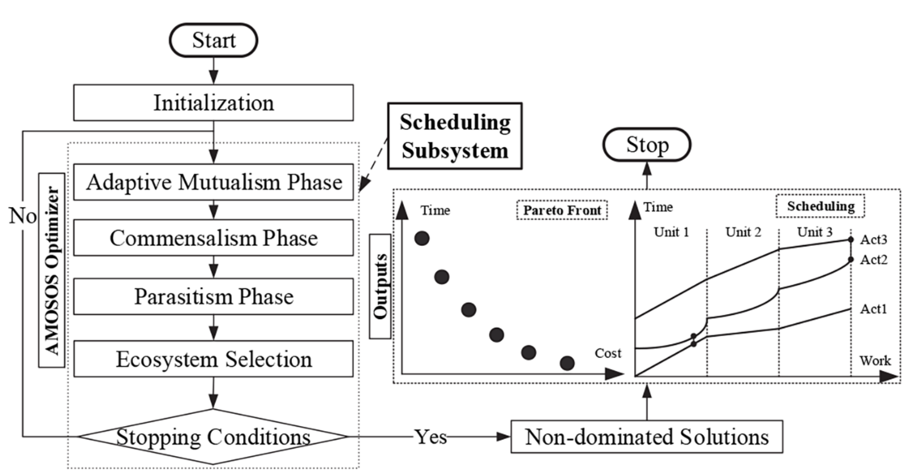


Figure 3. Proposed model for TCT

AMOSOS uses Eqn (5) to generate uniform-randomly the first *ecosize* organisms of the population with $x_{i,j} \in [0,1]$:

$$X_{i,j}^{G=0} = LB_i + x_{i,j} * (UB_i - LB_i);$$
$$i = 1,2,...,D;\ j = 1,2,...,\text{ecosize}. \quad (5)$$

3.2. Decision variables

The vector with D elements in Eqn (6) is a candidate time-cost tradeoff in a project scheduling problem:

$$X = [X_{1,j}, X_{2,j}, ..., X_{i,j}, ..., X_{D,j}], \quad (6)$$

where D is both the number of decision variables and the number of activities. Index j specifies the j^{th} individual in the ecosystem. The element $X_{i,j}$ is an integer in the range $[1, M_i]$ (where M_i is the number of methods for executing each activity), which represents one method for executing activity i. A *ceil* function in Eqn (7) is employed to convert the real-valued variables of the SOS to their nearest integers to determine the methods for executing the activities:

$$X_{i,j} = Ceil(rand[0,1] \times UB(i)). \quad (7)$$

Along with crew information and the precedence relationships among activities, the $X_{i,j}$ values determine the times that are required to perform project activities and yield the project duration and costs, based on the scheduling subsystem that will be described below.

3.3. Scheduling subsystem

Once the AMOSOS organism has been created, the scheduling subsystem (S1) determines the project objectives. Figure 4a displays the generation of project objectives by the scheduling subsystem. The schedule inputs are obtained from the project information; they include the number of activities, their durations, sequence, task quantities, and buffer types and values, which are similar to those used by Hajdu et al. (2017). All singularity functions must be transformed into time over work $t(w)$ using their inverse functions $w(t)$. The time and work buffer – or, more precisely, the function that defines the edges – must also be transformed.

A numerical method is used to determine the relationship between predecessors and successors in a project network. Eqn (8) defines the singularity functions for all activities in repetitive projects.

$$t_{act}(w) = k_1.\langle w - a_1 \rangle^m + \sum_{i=2}^{U} k_i.\langle w - a_i \rangle^m - k_{i-1}.\langle w - a_{i-1} \rangle^m;$$
$$i = \overline{1,U};\ w = 0 : \Delta w : L, \quad (8)$$

where w is the variable on the work axis; a_i is the cutoff value on the w-axis at which the function becomes valid; and U is the number of repetitive units in the project. The values of w fall in range $[0; L]$. L is the total number of work units. The number of elements in array w depends

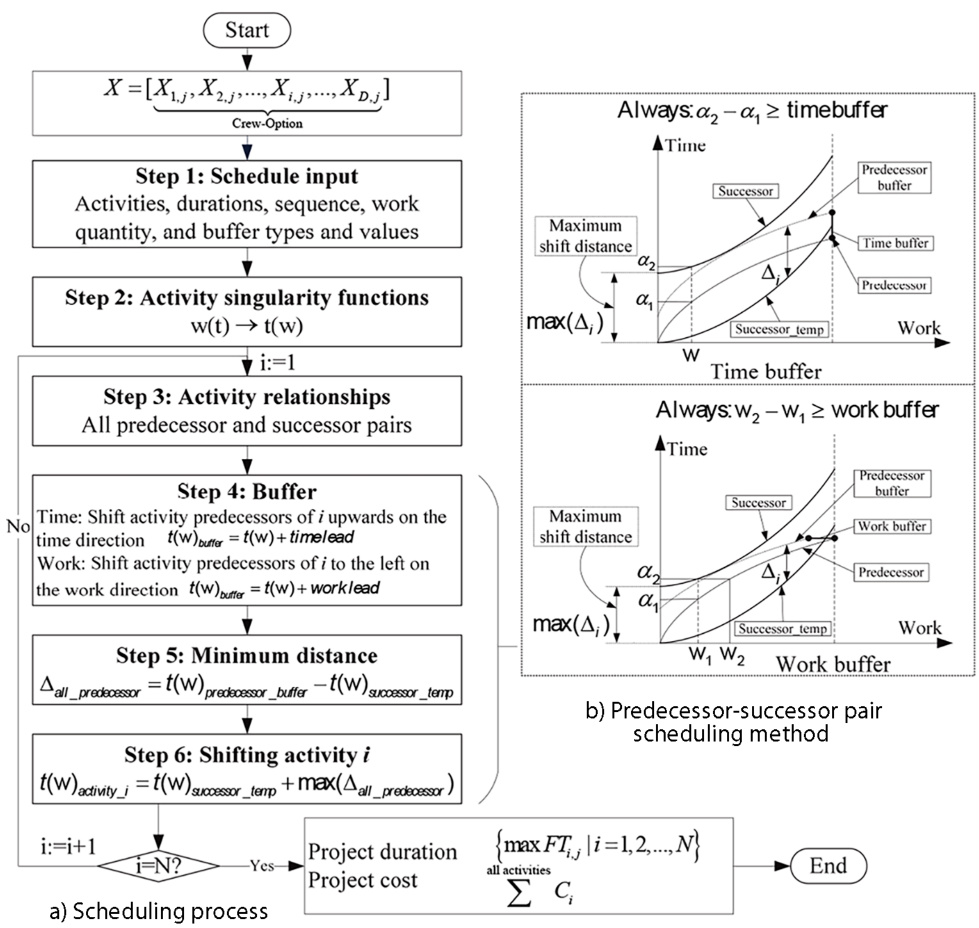


Figure 4. Scheduling subsystem

on the value of the work-step (Δw) (time or duration). A smaller Δw corresponds to more precise activity scheduling. In repetitive construction projects, the project manager sets the value of Δw based on the project's specifications. For example, in the excavation of a trench, Δw is set to one meter. The project information yields all predecessor and successor pairs for each activity in the project network, as mentioned with regard to step 3.

This work considers two types of continuous relation: one involves time lead (time buffer) and the other involves work lead (work buffer). The time buffer requires that the successor of *i* stays above its predecessor on the time axis (vertical axis). The work buffer ensures that the successor of *i* is always at least a minimum distance to the left of *i* along the work axis (horizontal axis). Figure 4b presents the fourth to sixth steps of the scheduling subsystem.

Figure 5 presents a sample trench project with two activities – A (linear singularity function) and B (non-linear singularity function). In this example, the trench is 24 meters long and consists of three units. The singularity function for each activity can be changed among units. For $\Delta w = 1(m)$, the maximum shift time is 4.8 hours at $w = 2$. Therefore, activity B will be shifted forward by 4.8 hours and the start time of activity B in the first unit is 4.8 hours from the starting time of *i*. The total project duration is calculated to be 42.2 hours.

The scheduling subsystem determines the two conflicting objective values of the project, which were described in the preceding subsection. Then, the AMOSOS algorithm uses these values to obtain the optimal combination of execution options for each activity.

3.4. Adaptive mutualism phase

In this phase, the design vector X_i of the i^{th} organism interacts with another design vector X_j that is randomly selected from the ecosystem (where $i \neq j$). Eqns (9), (10) and (11) are mathematical formulations of a mutualistic relationship between organism X_i and X_j:

$$X_{i\text{ new}} = X_i + rand(0,1) * (X_{best} - Mutual\_Vector * BF_1); \tag{9}$$

$$X_{j\text{ new}} = X_j + rand(0,1) * (X_{best} - Mutual\_Vector * BF_2); \tag{10}$$

$$Mutual\_Vector = \frac{X_i + X_j}{2}, \tag{11}$$

where X_{best} is the solution in the first non-dominated rank of the ecosystem. Organism X_i may gain significantly from interacting with organism X_j while, organism X_j may benefit only slightly from interacting with organism X_i. The benefit factors (BF_1) and (BF_2) are obtained randomly as either 1 or 2 with equal probability; these values specify two conditions under which an organism benefits fully or partially from the interaction, respectively. A lower benefit factor corresponds to a finer search using smaller steps, but with slower convergence (Tejani et al., 2016).

To balance exploration and exploitation effectively, the benefit factors are modified using Eqn (12):

$$MBF_1 = \frac{\sum F(X_i)}{\sum F(X_{best})}; MBF_2 = \frac{\sum F(X_j)}{\sum F(X_{best})}. \tag{12}$$

Organisms evolve to become fitter only if their post-interaction fitness dominates their pre-interaction fitness. In such a case, the old X_i and X_j are replaced immediately by X_{i_new} and X_{j_new}, respectively. The old X_i and X_j will be moved into the advanced population. Otherwise, X_{i_new} and X_{j_new} are added to the advanced population for the purpose of yielding the next-generation ecosystem. Accordingly, the convergence rate of the proposed algorithm is improved while favorable diversity is maintained, as the important information of population may be input into the algorithm after selection population phase.

3.5. Commensalism phase

After the mutualism phase is complete, organism X_i again randomly selects organism X_j as a new partner from the

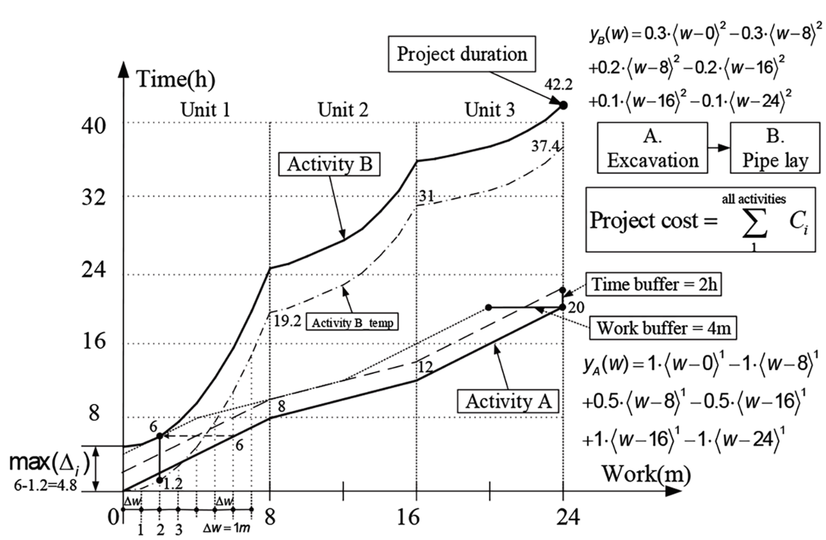


Figure 5. Example schedule with singularity functions

ecosystem. In this situation, organism X_i attempts to benefit from the interaction while organism X_j neither benefits nor suffers from it. Eqn (13) models commensal symbiosis between organism X_i and X_j.

$$X_{i\ \text{new}} = X_i + rand(-1,1) * (X_{best} - X_j), \tag{13}$$

where X_{best} is the organism in the first non-dominated set of population. The selection mechanism in the commensalism phase is analogous to those in the mutualism phase. If organism $X_{i\ new}$ dominates organism X_i, then $X_{i\ new}$ immediately replaces X_i, and X_i will be moved into the advanced population; otherwise, $X_{i\ new}$ will be moved into the advanced population.

3.6. Parasitism phase

A *Parasite_Vector* is initialized by duplicating organism X_i in the search space. To differentiate the *Parasite_Vector* from original organism X_i, some randomly selected design elements of the parasitism vector are modified using a random generated number within the lower and upper bounds. The total number of modified elements is an integer between one and the number of decision variables. The location of the modified elements is determined stochastically.

Then, the *Parasite_Vector* again randomly selects a new organism X_j ($i \neq j$) from the ecosystem for acting in the parasitism phase. The fitness of *Parasite_Vector* is compared with that of organism X_j. The selection mechanism in the parasitism phase is similar to those in the two phases above. If the fitness of *Parasite_Vector* exceeds or equals that of X_j, then *Parasite_Vector* will take the position of organism X_j in the current ecosystem and X_j will be moved into the advanced population. Otherwise, *Parasite_Vector* will enter the advanced population.

3.7. Ecosystem selection

The size of the ecosystem remains *ecosize* throughout the optimization process. Therefore, the *ecosize* best (elite) organisms for the next generation are selected from the combined ecosystem (current and advanced populations) by fast non-dominated sorting technique (Deb, Pratap, Agarwal, & Meyarivan, 2002) and crowding entropy method (Wang, Wu, & Yuan, 2010a).

At first, the combined ecosystem will be divided into the non-dominated sets from F_1 to F_n using the fast non-dominated sorting technique. The solutions in the best non-dominated set (Set F_1) are chosen as the first members of the main ecosystem. If F_1 is smaller than *ecosize*, then the subsequent non-dominated fronts in rank order (F_2, F_3 …) are chosen as the remaining members of the ecosystem. This procedure terminates when all positions in the ecosystem are filled. Assume that F_k is the last non-dominated set to be selected. Normally, the number of solutions in all sets F_1 to F_k exceeds *ecosize*. Therefore, the *ecosize* members are identified using the crowding entropy sorting technique.

3.8. Stopping conditions

When a user-specified stopping threshold, such as the maximum number of generations $G_{\max}$ or the maximum number of function evaluations (NFE), is reached, the optimization terminates. The stopping condition for the proposed algorithm is specified as the reaching of a maximum number of generations. The termination of the optimization process yields a set of optimal solutions, called the Pareto front. The project planners evaluate the pros and cons of each potential solution to determine the best schedule.

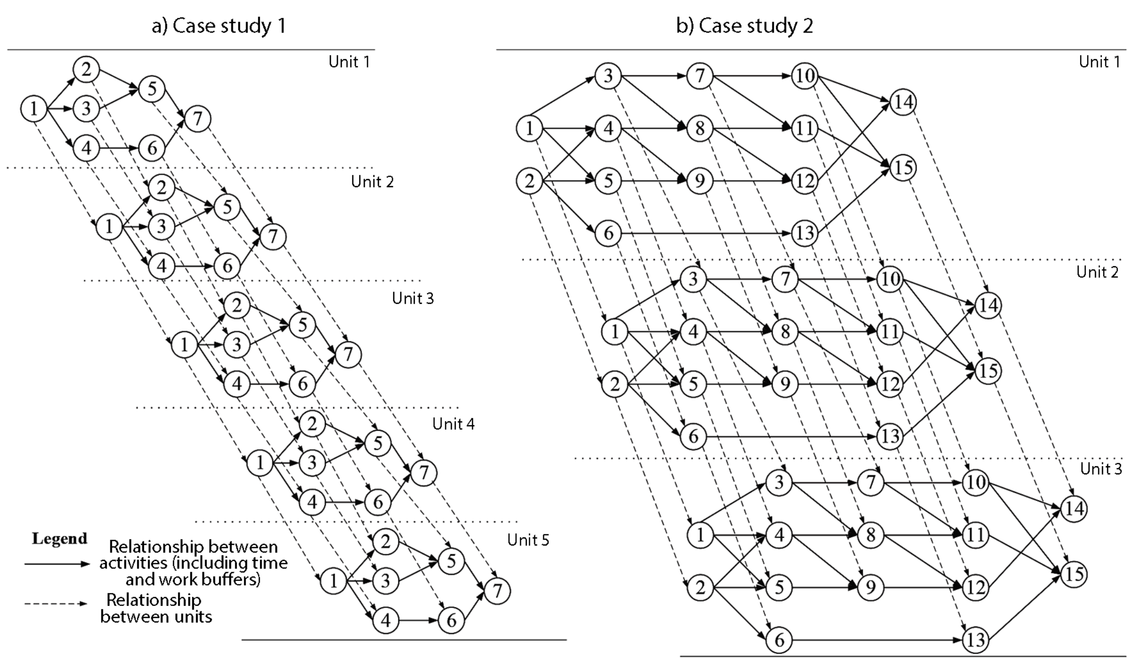


Figure 6. Network of projects

4. Case studies

This investigation analyzed two case studies to demonstrate the effectiveness of the proposed model for solving the TCT problem. Figure 6 shows the precedence relationships of the network projects. In these networks, the relationships between the same activities of different units are FS0 (finish to start without lag time to allow the continuous of crew). The first case is considered as a bridge project which involves seven activities and five segments (units) with a total length of 250 meters (because each unit is 50 meters). The second case is medium sized as a 15-activity project (450 meters with 150 meters in each segments). The second project consists of three similar sections (units), and each includes the repetitive activities in sequence from (1) to (15). Tables 1a and 1b provide project information, including precedence relationships among activities, activity durations, and the costs of the crew in the projects. The execution selections (options) are decision variables that are chosen by the optimizer.

For instance, each option of the activity named "*site preparation*" in the first case has different value of time and cost. The first option is about crash time and normal cost. The last option is about crash cost and normal time. The difference between the original CPM model and the proposed model is that the relationship between two activities in each unit of a project has the time and work buffer to allow continuous precedence relationships.

In Tables 1a and 1b, m is the order of the exponent of the modeled curve segment; k is the scaling factor, which can be obtained from field observations, and c is the cost of the activity. The coefficients m and k denote the non-linear change in progress with respect to working time and amount of work done, such as when piles that become gradually deeper from beginning to the end are drilled, height at which work is done changes, and limited space has an increasing effect.

In the first case, for example, activity 1, site preparation, has five possible execution methods; each method has different values of m and k for each of the five segments. If the project manager selects method 1 for the first activity, site preparation, then the project's cost is \$90,000 and the time of the activity is determined as $m = \{1;1;1;1;1\}$ and $k = \{0.5;1;1;2;0.5\}$. Then, Eqn (8) yields the singularity function of time (t), depending on the amount of work (w) associated with the first activity in each segment or unit, given by Eqn (14). The two first terms are the order and the scaling factor as a pair $\{m;k\} = \{1;0.5\}$, specifying the schedule for activity 1 in the first segment (from 0 to 50 m in length).

$$t_{act\_1}(w) = 0.5.\langle w-0\rangle^1 - 0.5.\langle w-50\rangle^1 + 1.\langle w-50\rangle^1 - 1.\langle w-100\rangle^1 + 1.\langle w-100\rangle^1 - 1.\langle w-150\rangle^1 + 2.\langle w-150\rangle^1 - 2.\langle w-200\rangle^1 + 0.5.\langle w-200\rangle^1 - 0.5.\langle w-250\rangle^1. \quad (14)$$

Tables 2a and 2b provide the time and work buffers for each "*predecessor-successor*" pair. In the case 1, for instance, activities 1 and 2 constitute one pair, whose time buffer is 20 hours and work buffer is 10 meters. Buffers are frequently used to protect production processes from the negative impact of low productivity, leading to a smooth, safe and reliable construction workflow.

4.1. Optimization results obtained using proposed model

The proposed model is used to minimize simultaneously project duration and cost while both the logical relationship constraints and the time and work buffer constraints are satisfied. In the preliminary optimization process, a trial-and-error strategy (Tran et al., 2016) was used to obtain the following suitable parameters for the AMOSOS optimizer in the case study. The parameter *ecosize* was set to 100 and the maximum number of generations was set to 100. To obtain smooth curves of the non-linear singularity functions of activities, the work-step (Δw) was set to 0.01 meters. Thirty independent optimization runs were conducted to avoid randomness.

Figure 7 displays typical Pareto optimal fronts for the first case study, which are obtained using AMOSOS. The Pareto front represents clearly the relationship between project duration and cost. S1 has the shortest project duration; S3 has the smallest cost, and other solutions strike trade-off these two objectives. This two-dimensional visualization of the trade-offs may help decision-makers evaluate the impact of various potential resource-utilization plans on project performance.

Figure 8 shows the schedules (start time, number of crew) assigned to each activity of the three selected non-dominated solutions and the corresponding project times and costs for the bridge construction project (case 1). For example, Figure 8c presents the optimal schedule with respect to cost, obtained using the AMOSOS-TCT-based method for the bridge project, with total project duration of 1693.3 hours and a project cost of \$625,000. The schedule specifies, along with the sequence, start times and finish times of all activities, the profile of the assigned crew. The curves (linear or non-linear) in Figure 8 represent the

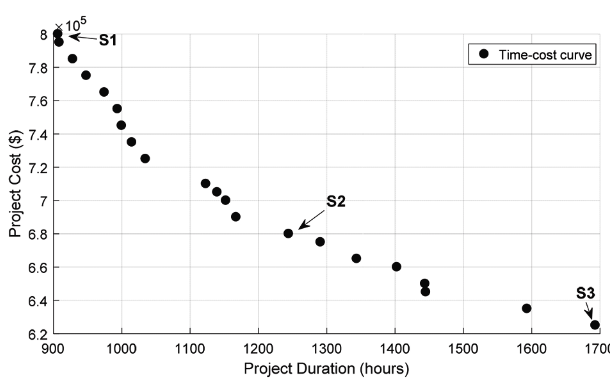


Figure 7. Typical time-cost-trade-off Pareto front obtained using proposed model

Table 1a. Bridge project data – case 1

No	Activity name	Prede-cessors	Option 1			Option 2		
			m	k	c	m	k	c
1	Site preparation	–	1;1;1;1;1	0.5;1;1;2;0.5	90	1;1;1;1;1	1;1;2;2;1	80
2	Forms and re-bars	1;	0.5;0.5;0.5;0.5;0.5	10;14;20;20;20	130	0.5;0.5;0.5;0.5;0.5	12;16;22;22;22	130
3	Excavation	1;	0.5;0.5;0.5;0.5;0.5	8;12;15;15;15	85	0.5;0.5;0.5;0.5;0.5	8;12;15;15;15	75
4	Pre-cast concrete girders	1;	1;1;1;1;1	1;1;2;2;3	100	1;1;1;1;1	2;2;3;3;4	90
5	Pour foundation and piers	2;3	2;2;2;2;2	0.1;0.05;0;0.09;0.08	170	2;2;2;2;2	0.15;0.07;0;0.1;0.09	150
6	Deliver pre-cast girders:	4;	2;2;2;2;2	0.1;0.05;0.06;0.04;0.06	140	2;2;2;2;2	0.11;0.06;0.07;0.05;0.07	120
7	Erect girders	5;6	0.5;0.5;0.5;0.5;0.5	10;12;15;15;15	150	0.5;0.5;0.5;0.5;0.5	12;14;16;16;16	140

Option 3			Option 4			Option 5		
m	k	c	m	k	c	m	k	c
1;1;1;1;1	2;2;2;2;2	70	1;1;1;1;1	2;2;3;3;2	60	1;1;1;1;1	2;2;4;4;2	50
0.5;0.5;0.5;0.5;0.5	13;17;24;24;24	110	0.5;0.5;0.5;0.5;0.5	14;18;25;25;25	100	–	–	–
0.5;0.5;0.5;0.5;0.5	9;13;16;16;16	70	–	–	–	–	–	–
1;1;1;1;1	3;3;4;4;5	85	1;1;1;1;1	4;4;5;5;6	80	–	–	–
2;2;2;2;2	0.16;0.08;0;0.11;0.1	140	2;2;2;2;2	0.17;0.1;0;0.12;0.12	130	2;2;2;2;2	0.18;0.11;0;0.13;0.13	120
2;2;2;2;2	0.12;0.07;0.08;0.06;0.08	100	2;2;2;2;2	0.13;0.08;0.09;0.07;0.09	95	2;2;2;2;2	0.14;0.09;0.1;0.08;0.1	90
0.5;0.5;0.5;0.5;0.5	13;15;17;17;17	130	0.5;0.5;0.5;0.5;0.5	14;16;18;18;18	120	0.5;0.5;0.5;0.5;0.5	15;17;20;20;20	115

Noted that: Cost values (in thousands)

Table 1b. Project data – case 2

No	Prede-cessors	Option 1			Option 2			Option 3		
		m	k	c	m	k	c	m	k	c
1	–	1;1;1	0.5;1;1	90	1;1;1	1;1;2	80			
2	–	0.5;0.5;0.5	1;1;2	130	0.5;0.5;0.5	2;1;2	120	1;1;1	2;2;2	100
3	1;	0.5;0.5;0.5	1;1;2	85	0.5;0.5;0.5	2;1;3	75	2;2;2	2;1;3	50
4	1;2	1;1;1	1;1;2	100	1;1;1	2;2;2	90	1;1;1	2;2;3	60
5	1;2	2;2;2	0.1;0.05;0	170	2;2;2	0.15;0.07;0	150	2;2;2	0.15;0.07;0	120
6	2;	2;2;2	0.1;0.05;0.06	140	2;2;2	0.11;0.06;0.07	120			
7	3;	0.5;0.5;0.5	1;1;2	150	0.5;0.5;0.5	2;1;1	140	0.5;0.5;0.5	1;2;2	100
8	3;4	2;2;2	0.15;0.07;0	80	2;1;2	2;1;1	60	2;2;2	2;1;2	50
9	4;5	0.5;0.5;0.5	0.1;0.05;0.2	150	0.5;0.5;0.5	0.15;0.7;0.8	120	2;2;2	0.5;0.5;1	90
10	7;	2;2;2	0.5;0.5;0.5	90	2;2;2	1;1;2	80			
11	7;8	0.5;0.5;0.5	0.1;0.05;0	200	1;1;1	1;1;1	180			
12	8;9	1;1;1	0.15;0.07;0	180	2;2;2	1;1;1	150	2;2;2	0.15;0.07;0	120
13	6	2;2;2	0.1;0.05;0.06	150	1;1;1	2;1;2	110	1;1;1	2;2;2	80
14	10;12	2;2;2	1;1;2	160	0.5;0.5;0.5	1;2;2	100	0.5;0.5;0.5	2;1;2	60
15	10;11;13	0.5;0.5;0.5	0.1;0.05;0	100	0.5;0.5;0.5	0.11;0.06;0.07	60	0.5;0.5;0.5	0.11;0.05;0.04	40

Table 2a. Time and work buffer data – case 1

Time buffer (hours); Work buffer (meters)							
From activity	To activity						
	1	2	3	4	5	6	7
1	–	20;10	30;15	25;20	–	–	–
2	20;10	–	–	–	30;12	–	–
3	30;15	–	–	–	20;14	–	–
4	25;20	–	–	–	–	25;18	–
5	–	30;12	20;14	–	–	–	35;16
6	–	–	–	25;18	–	–	30;15
7	–	–	–	–	35;16	30;15	–

Table 2b. Time and work buffer data – case 2

Time buffer (hours); Work buffer (meters)															
From Act.	To activity														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	–	–	20;15	15;8	10;5	–	–	–	–	–	–	–	–	–	–
2	–	–	–	15;12	20;10	10;5	–	–	–	–	–	–	–	–	–
3	20; 15	–	–	–	–	–	20;15	25;15	–	–	–	–	–	–	–
4	15;8	15;12	–	–	–	–	–	10;5	15;9	–	–	–	–	–	–
5	10;5	20;10	–	–	–	–	–	–	12;10	–	–	–	–	–	–
6	–	10;5	–	–	–	–	–	–	–	–	–	–	18;15	–	–
7	–	–	20;15	–	–	–	–	–	–	15;10	15;8	–	–	–	–
8	–	–	25;15	10;5	–	–	–	–	–	–	20;10	10;5	–	–	–
9	–	–	–	15;9	12;10	–	–	–	–	–	–	18;10	–	–	–
10	–	–	–	–	–	–	15;10	–	–	–	–	–	–	12;8	10;5
11	–	–	–	–	–	–	15;8	20;10	–	–	–	–	–	–	20;10
12	–	–	–	–	–	–	–	10;5	18;10	–	–	–	–	15;9	–
13	–	–	–	–	–	18;15	–	–	–	–	–	–	–	–	20;15
14	–	–	–	–	–	–	–	–	–	12;8	–	15;9	–	–	–
15	–	–	–	–	–	–	–	–	–	10;5	20;10	–	20;15	–	–

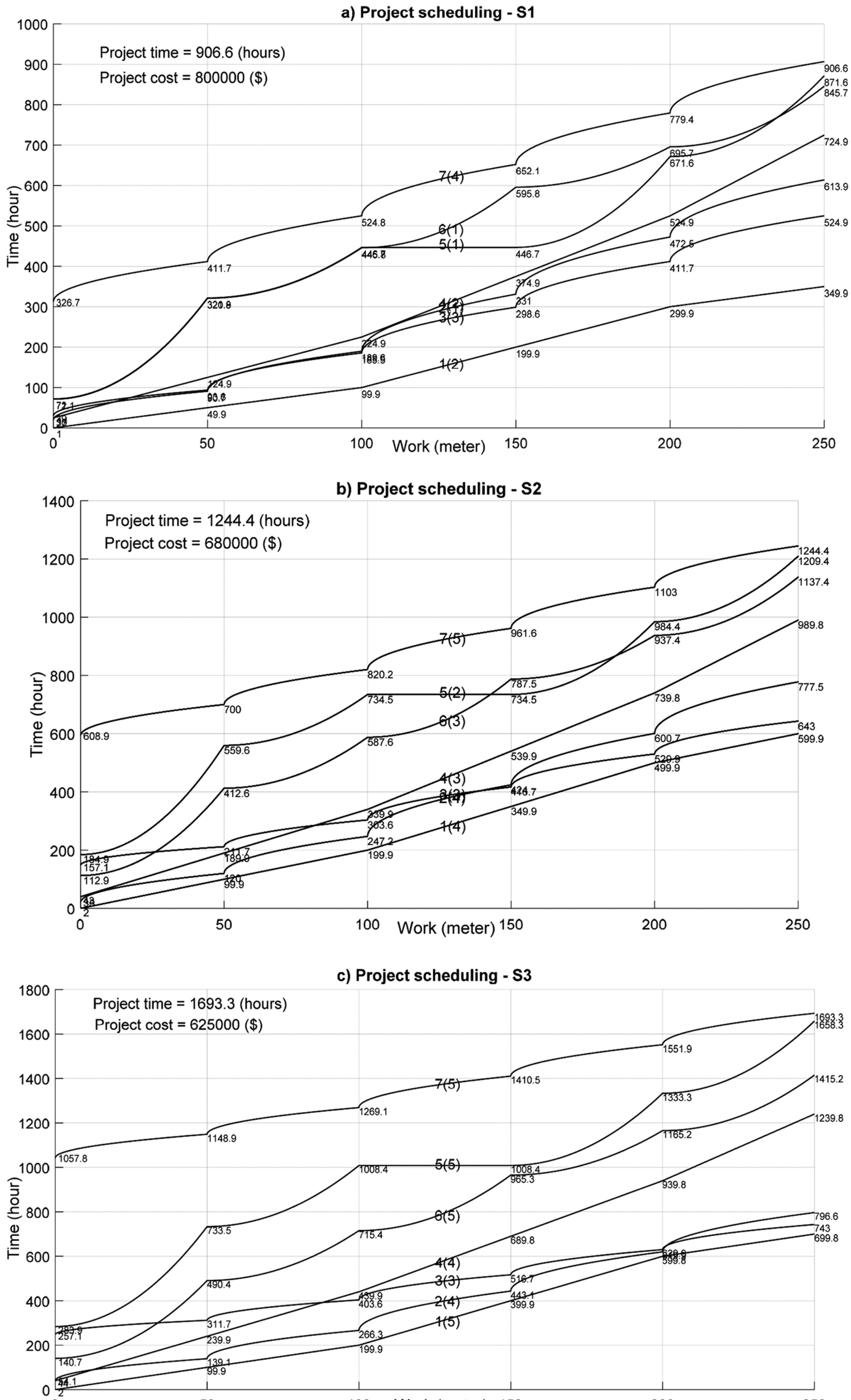


Figure 8. Schedules associated with three non-dominated solutions for case study

schedule of activities from the beginning to the end of the project.

As an example, in Figure 8c, the start times of activity 7 in the five segments are 1057.8, 1148.9, 1269.1, 1410.5 and 1551.9 hours. The numbers in the middle of each curve refer to the activity, and the corresponding execution option is indicated in parentheses. The numbers in the middle of the highest curve (activity 7) in Figure 8c are 7(5), revealing that the project manager will select option 5 for activity 7. As evident from the results that were obtained for the bridge project, the proposed model achieved the research goals and objectives. It generated a schedule for the project that satisfied all relevant constraints. The proposed model minimized the duration and cost of the project by optimizing the crew assignment (execution option). Therefore, the proposed AMOSOS-TCT is an efficient model for scheduling a non-unit repetitive project with singularity functions.

4.2. Analysis and comparison of results

The AMOSOS performance was compared with that of four widely used algorithms – the non-dominated sorting genetic algorithm (NSGA-II) (Deb et al., 2002), multi-objective particle swarm optimization (MOPSO) (Dai, Wang, & Ye, 2015), the multi-objective artificial bee colony (MOABC) algorithm (Akbari, Hedayatzadeh, Ziarati, & Hassanizadeh, 2012), and multi-objective differential evolution (MODE) (Ali, Siarry, & Pant, 2012). All five algorithms were executed for 100 generations with a population size of 100.

NSGA-II used a crossover probability of $p_c = 0.9$ and a mutation constant of $p_m = 0.5$. In MOPSO, the inertia weight w was in the range 0.3 to 0.7, and the two learning factors c_1 and c_2 were both set to 2. In MOABC, the limit factor l was set to 20. In MODE, the mutant constant was $F = 0.9$ and the crossover probability was $CR = 0.5$. Each algorithm was applied independently 30 times to the bridge project.

Figure 9 plots typical Pareto fronts that were obtained when the five algorithms were applied to the case study. AMOSOS achieved the best distribution of solutions along the Pareto front and provided many optimal solutions. AMOSOS also yielded a better spread than did the four competing algorithms; NSGA-II yielded the worst spread.

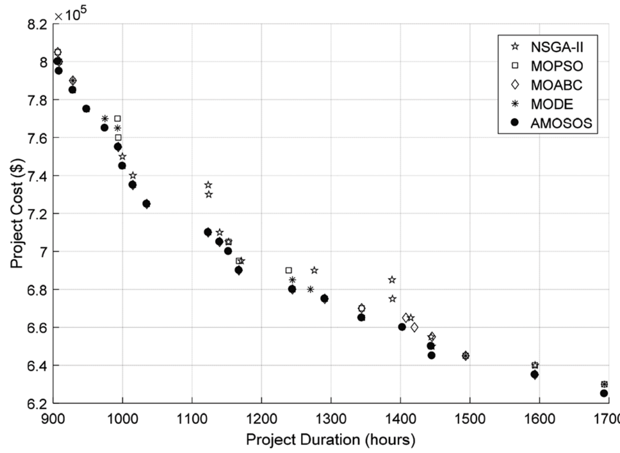


Figure 9. Pareto fronts obtained using five algorithms

The following quantitative assessments were performed to evaluate further the effectiveness of the proposed method.

The performance of an optimization algorithm can be evaluated using various criteria. In this work, the following criteria and corresponding indicators are used (Senouci & Mubarak, 2016; Zitzler, Thiele, Laumanns, Fonseca, & da Fonseca, 2003).

1. C-metric (C): this metric compares the quality of two non-dominated sets of two considered algorithm (Zitzler & Thiele, 1999). The C-metric is computed without considering the standard efficient frontier. Let S_1 and S_2 be two approximate sets of decision solutions. The C-metric is a binary index that is given by Eqn (15):

$$C(S_1,S_2)=\frac{\left|\{a_2 \in S_2;\ \exists a_1 \in S_1\ : a_1 \le a_2\}\right|}{\left|S_2\right|}. \quad (15)$$

The numerator in Eqn (15) is the number of solutions in S_2 that are dominated by at least one solution in S_1, and the denominator equals the total number of solutions in S_2. Therefore, $C(S_1, S_2) = 1$ means that all solutions in S_2 are dominated by at least one solution in S_1. If $C(S_1, S_2) = 0$, then no solution in S_2 is dominated by a solution in S_1. Owing to the asymmetry of the C-metric, the comparison requires checking both $C(S_1, S_2)$ and $C(S_2, S_1)$ (Wang & Singh, 2009).

Table 3 compares the values of the C-metric for the five algorithms, where A_1, A_2, A_3, A_4, and A_5 are AMOSOS,

Table 3. Comparison of C-metrics for various algorithms

Performance measurement		$C(A_1, A_2)$	$C(A_2, A_1)$	$C(A_1, A_3)$	$C(A_3, A_1)$	$C(A_1,A_4)$	$C(A_4, A_1)$	$C(A_1,A_5)$	$C(A_5, A_1)$
Case 1	Best	0.94	0.00	0.94	0.00	1.00	0.00	0.92	0.00
	Worst	0.57	0.00	0.52	0.00	0.42	0.00	0.75	0.00
	Average	0.77	0.00	0.74	0.00	0.72	0.00	0.86	0.00
	Std.	0.17	0.00	0.19	0.00	0.24	0.00	0.10	0.00
Case 2	Best	0.49	0.03	0.87	0.12	0.70	0.14	0.97	0.12
	Worst	0.00	0.00	0.81	0.05	0.63	0.07	0.89	0.01
	Average	0.28	0.06	0.83	0.09	0.69	0.11	0.95	0.07
	Std.	0.13	0.13	0.05	0.04	0.05	0.08	0.02	0.03

MODE, MOABC, MOPSO, and NSGA-II, respectively. These results reveal that AMOSOS solutions dominate more than 77% of the MODE solutions, 74% of the MOABC solutions, 72% of the MOPSO solutions, and 86% of the NSGA-II solutions on average for the first case. In the second case, on average the AMOSOS dominates more than 28% of MODE solutions, 83% of MOABC solutions, 69% of MOPSO solutions, and 95% of NSGA-II solutions, respectively.

2. Diversification Metric (DM): this metric is used to measure the diversity of the obtained non-dominated solutions (Maghsoudlou, Afshar-Nadjafi, & Niaki, 2016). The algorithm with higher value of DM will have better performance. The value of this metric is computed as below.

$$DM = \sqrt{\sum_{i=1}^{l}\left(Min\, f_i - Max\, f_i\right)^2}\ , \tag{16}$$

where $Min\, f_i$ and $Max\, f_i$ are the minimum and the maximum value of each fitness function among all non-dominated solutions resulted from the algorithms.

3. Mean Ideal Distance (MID): this index is defined to measure the distance of the solutions in the Pareto fronts from an ideal solution (Maghsoudlou, Afshar-Nadjafi, & Akhavan Niaki, 2017). The formula of this metric is defined as follows:

$$MID = \frac{\sum_{i=1}^{n}\sqrt{\left(\frac{f^1{}_i - f^1{}_{best}}{f^{1\max}_{total} - f^{1\min}_{total}}\right)^2 + \left(\frac{f^2{}_i - f^2{}_{best}}{f^{2\max}_{total} - f^{2\min}_{total}}\right)^2}}{n}\ , \tag{17}$$

where $f^1{}_i$ and $f^2{}_i$ are the value of the first and the second objective functions per non-dominated solutions obtained by an algorithm. n is considered as the number of non-dominated solutions. Algorithms with lower *MID* values are more desirable.

4. Spread (*SP*): this index (Wang, Wu, & Yuan, 2010a, 2010b) quantifies the spread of the obtained non-dominated solutions. Eqn (18) gives the value of *SP*:

$$SP = \frac{d_f + d_l + \sum_{i=1}^{N-1}\left|d_i - \bar{d}\right|}{d_f + d_l + (N-1)\bar{d}}\ , \tag{18}$$

where N is the number of non-dominated solutions that have already been found. The parameters d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions in the obtained non-dominated set.

The parameter d_i is the Euclidean distance between consecutive solutions in the obtained non-dominated set of solutions and $\bar{d}$ is the mean of all d_i. This metric is smaller for better distributions and has a value of zero for the most widely and uniformly spread-out set of non-dominated solutions.

5. Hyper-volume (*HV*): this metric is also known as the hyper-area. It is a unary metric of the size of the objective space that is covered by an approximation set Ω. A reference point W and the solution X_i must be used as the diagonal corners of a hypercube v_i for each solution $X_i \in \Omega$ (Wu, Wang, Yuan, & Zhou, 2010; Zitzler et al., 2003). Eqn (19) yields the union of all hypercubes, *HV*.

$$HV = \bigcup_{i=1}^{|\Omega|} v_i\ . \tag{19}$$

The algorithm with highest *HV* outperforms the other algorithms. The *HV* value is normalized using the reference point.

6. Computational time (CPU time). The central processing unit (CPU) time of obtaining the corresponding Pareto fronts with pre-specified maximum number of iteration $G_{\max}$ are compared under the exactly identical conditions including both hardware and software platforms. Table 4 displays the computational time required to achieve the Pareto fronts for various algorithms.

Table 4 provides the average experimental outputs of all compared algorithms, which support the claim that on average AMOSOS outperforms the other four algorithms.

4.3. Multi-attribute decision making

In order to prioritizing algorithms in terms of all performance metrics, a hybrid multi-attribute decision making method called AHP-TOPSIS is applied to rank the algorithms in terms of all the metrics simultaneously (Maghsoudlou et al., 2016). The approach consists of two main

Table 4. The average experimental outputs of five algorithms

Case study	Algorithms	*DM*	*MID*	*SP*	*HV*	CPU time (in seconds)
Case1	NSGA-II	109995.5	0.965	0.466	0.585	12.83
Case2		657.8	8.157	0.812	0.606	36.03
Case1	MOPSO	174997.7	0.723	0.494	0.462	13.45
Case2		757.8	7.208	0.745	0.742	36.93
Case1	MODE	169997.6	0.762	0.442	0.815	13.78
Case2		993.5	7.342	0.762	0.735	37.07
Case1	MOABC	169997.6	0.746	0.442	0.795	13.54
Case2		1435.6	7.215	0.723	0.776	37.23
Case1	AMOSOS	174997.7	0.714	0.366	0.855	12.34
Case2		1545.0	5.821	0.622	0.948	35.89

steps the weight identification of the metrics and final rank determination to decide the best algorithms with highest priority value. In the first step, an analytical hierarchy process (AHP) method is used to identify the weights of the metrics (Saaty, 1989). The performance metrics and the algorithms are considered as criteria and alternatives, respectively. The computational process for the weights of each metric followed these steps: (1) create a pairwise comparison matrix based on aggregating judgement from different experts; (2) sum up each column of the comparison matrix; (3) divide each corresponding element by its sum-up value; and (4) take the average of each row to obtain the weights of the criteria. Table 5 shows the paired comparisons matrix as input data of the AHP method, and the weights assigned to the metrics.

Table 5. The pairwise comparisons matrix of the metrics

Metric	*DM*	*MID*	SP	*HV*	Weights
DM	1	0.5	1	2	0.243
MID	2	1	0.5	1	0.246
SP	1	2	1	3	0.362
HV	0.5	1	0.333	1	0.148

In the second step, the TOPSIS method is used to determine the algorithm with the best performance in solving TCT problem in both case studies. The following steps are used to obtain the priorities of the algorithms: (1) create the decision matrix; (2) normalize decision matrix by dividing each corresponding element by the maximum value; (3) calculate the weighted normalized decision matrix by multiplying the weighted normalized matrix and corresponding weight of the criteria; (4) calculate Euclidean distances of the alternatives from the positive and the negative ideal solutions, which are defined as the biggest value of the positive criteria and the smallest value of the negative criteria; (5) Calculate the relative closeness of each alternative to the ideal solution.

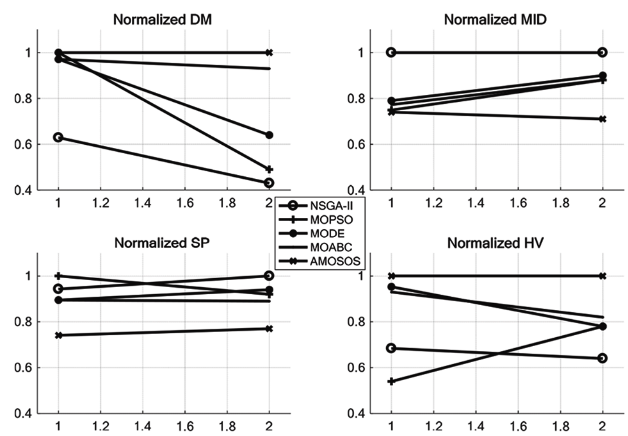


Figure 10. Graphs of the metrics over two case studies

Table 6 presents the decision matrix, the normalized decision matrix, the weighted normalized decision matrix, the Euclidean distances of the alternatives, and the relative closeness of the alternatives for both case studies. As shown in Table 6, the AMOSOS has the best performance in solving TCT problems. Furthermore, Figure 10 demonstrates the trend of all metrics over the case studies. Obviously, the proposed AMOSOS algorithm yielded results that were better than those obtained using the other approaches in terms of all metrics.

Conclusions and further work

This work developed a scheduling method-based multiple objective optimization to solve time-cost tradeoff problems for repetitive projects with singularity functions. A numerical method was used to determine the relationship between predecessor and successor activities, in which time and work buffers were used to calculate total project duration. The proposed scheduling method is easy to understand and convenient to implement, and generates accurate results quickly.

Table 6. The result of the TOPSIS method for case studies

No	Algorithms	Decision matrix				Normalize decision matrix				Weighted normalized decision matrix				d_{i+}	d_{i-}	CL	Rank
		DM	*MID*	*SP*	*HV*	*DM*	*MID*	*SP*	*HV*	*DM*	*MID*	*SP*	*HV*				
Case 1	NSGA-II	109995.5	0.965	0.466	0.585	0.63	1.00	0.94	0.68	0.15	0.25	0.34	0.10	0.020	0.001	0.04	5
	MOPSO	174997.7	0.723	0.494	0.462	1.00	0.75	1.00	0.54	0.24	0.18	0.36	0.08	0.013	0.012	0.47	4
	MODE	169997.6	0.762	0.442	0.815	0.97	0.79	0.89	0.95	0.24	0.19	0.32	0.14	0.003	0.015	0.82	2
	MOABC	169997.6	0.746	0.442	0.795	0.97	0.77	0.89	0.93	0.24	0.19	0.32	0.14	0.003	0.015	0.82	2
	AMOSOS	174997.7	0.714	0.366	0.855	1.00	0.74	0.74	1.00	0.24	0.18	0.27	0.15	0.000	0.026	1.00	1
Case 2	NSGA-II	657.8	8.157	0.812	0.606	1.00	1.00	1.00	0.64	0.24	0.25	0.36	0.09	0.034	0.000	0.00	5
	MOPSO	757.8	7.208	0.745	0.742	0.93	0.88	0.92	0.78	0.23	0.22	0.33	0.12	0.021	0.002	0.10	4
	MODE	993.5	7.342	0.762	0.735	0.64	0.90	0.94	0.78	0.16	0.22	0.34	0.11	0.015	0.004	0.23	3
	MOABC	1435.6	7.215	0.723	0.776	0.49	0.88	0.89	0.82	0.12	0.22	0.32	0.12	0.005	0.018	0.79	2
	AMOSOS	1545	5.821	0.622	0.948	0.43	0.71	0.77	1.00	0.10	0.18	0.28	0.15	0.000	0.034	1.00	1

AMOSOS was used to optimize crew assignment to minimize simultaneously the duration and cost of non-unit based repetitive projects. The proposed algorithm has modified the mutualism phase from the original algorithm to balance the exploration and exploitation phases of optimization. AMOSOS has a more powerful global search ability and local search ability than the considered benchmarked algorithms.

A bridge construction project of sufficient complexity with respect to the nonlinear behavior of related activities and their buffers, was analyzed to validate the scheduling method and evaluate the effectiveness of the AMOSOS-TCT model in generating optimal trade-offs between project time and project cost for non-unit based repetitive construction projects with singularity functions. The proposed scheduling method satisfied the research goals and objectives.

AMOSOS outperformed the considered benchmark algorithms in terms of diversity of characteristics and compromise solutions. The Pareto front that was generated by AMOSOS provides information that helps decision-makers in construction projects optimally trade-off the two important considerations of duration and cost. This information should be useful for construction planners and decision-makers who must minimize both the duration and the cost of repetitive construction projects.

In conclusion, the contributions of this study are as follows. (1) The well-known concept of singularity functions is integrated into construction scheduling; (2) a numerical method is used to optimize linear/ non-linear schedules for repetitive construction projects. The AMOSOS-TCT model competes favorably with the traditional linear model in terms of rate of calculation, ease of understanding and ease of computerization. It imposes no limitation on the number of decision variables and objectives.

Project managers have only to define the decision variables and provide objective functions based on the project network and constraints. With small modifications, the proposed system has the potential to solve other multi-objective optimization problems, such as scheduling and resource problems, in the field of construction management.

Non-unit based repetitive project problems that are concerned with total project cost minimization, work continuity, and quality maximization, are frequently encountered in construction management. Further work must be conducted to address the above problems to enable the proposed system to be used to solve complex non-unit based repetitive projects that involve more objectives. Therefore, the scheduling model must be modified. The coefficients of singularity functions, such as the order in the exponent of the curve segment *m,* and the scaling factor *k,* which must be known to obtain the non-linear change in progress, are difficult to evaluate precisely. This difficulty is one of the weaknesses of the model. Future work should investigate the use of linguistic and uncertain terms to specify the values of these coefficients.

Funding

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2018.07.

Disclosure statement

We declare that we have no any competing financial, professional, or personal interests from other parties.

References

Abdullahi, M., & Ngadi, M. A. (2016). Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment. *Plos One*, *11*(6), 0158229. https://doi.org/10.1371/journal.pone.0158229

Abdullahi, M., Ngadi, M. A., & Abdulhamid, S. M. (2016). Symbiotic organism search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems*, *56*, 640-650. https://doi.org/10.1016/j.future.2015.08.006

Agdas, D., Warne, D. J., Osio-Norgaard, J., & Masters, F. J. (2018). Utility of genetic algorithms for solving large-scale construction time-cost trade-off problems. *Journal of Computing in Civil Engineering*, *32*(1), 1-10. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000718

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation*, *2*, 39-52. https://doi.org/10.1016/j.swevo.2011.08.001

Ali, M., Siarry, P., & Pant, M. (2012). An efficient differential evolution based algorithm for solving multi-objective optimization problems. *European Journal of Operational Research*, *217*(2), 404-416.

Chassiakos, A. P., & Sakellaropoulos, S. P. (2005). Time-cost optimization of construction projects with generalized activity constraints. *Journal of Construction Engineering and Management*, *131*(10), 1115-1124. https://doi.org/10.1061/(ASCE)0733-9364(2005)131:10(1115)

Cheng, M.-Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. *Computers & Structures*, *139*, 98-112. https://doi.org/10.1016/j.compstruc.2014.03.007

Cheng, M.-Y., Prayogo, D., & Tran, D.-H. (2016). Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search. *Journal of Computing in Civil Engineering*, *30*(3). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000512

Cheng, M.-Y., Tran, D.-H., & Cao, M.-T. (2013). Chaotic initialized multiple objective differential evolution with adaptive mutation strategy (CA-MODE) for construction project time-cost-quality trade-off. *Journal of Civil Engineering and Management*, *22*(2), 210-223. https://doi.org/10.3846/13923730.2014.897972

Cho, K., Hong, T., & Hyun, C. T. (2013). Space zoning concept-based scheduling model for repetitive construction process. *Journal of Civil Engineering and Management*, *19*(3), 409-421. https://doi.org/10.3846/13923730.2012.757561

Dai, C., Wang, Y., & Ye, M. (2015). A new multi-objective particle swarm optimization algorithm based on decomposition. *Information Sciences*, *325*, 541-557. https://doi.org/10.1016/j.ins.2015.07.018

de Leon, P. G. (2008). *Graphical planning method*. In *PMICOS Annual Conference*. Chicago, USA.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182-197. https://doi.org/10.1109/4235.996017

Duman, S. (2017). Symbiotic organisms search algorithm for optimal power flow problem based on valve-point effect and prohibited zones. *Neural Computing and Applications*, *28*(11), 3571-3585. https://doi.org/10.1007/s00521-016-2265-0

Ezugwu, A. E.-S., & Adewumi, A. O. (2017). Discrete symbiotic organisms search algorithm for travelling salesman problem. *Expert Systems with Applications*, *87*, 70-78. https://doi.org/10.1016/j.eswa.2017.06.007

Feng, C.-W., Liu, L., & Burns, S. A. (1997). Using genetic algorithms to solve construction time-cost trade-off problems. *Journal of Computing in Civil Engineering*, *11*(3), 184-189. https://doi.org/10.1061/(ASCE)0887-3801(1997)11:3(184)

Fondahl, J. W. (1961). *A non-computer approach to the critical path method for the construction industry report #9*. Stanford: Department of Civil Engineering, Stanford University.

Fondahl, J. W. (1987). The history of modern project management. Precedence diagramming methods: origins and early development. *Project Management Journal*, *18*(2), 33-36.

Francis, A., & Miresco, E. T. (2002). Decision support for project management using a chronographic approach. *Journal of Decision Systems*, *11*(3-4), 383-404. https://doi.org/10.3166/jds.11.383-404

Haidu, M. (1996). Pdm time cost trade-off: activities are splittable or non-splittable. *Optimization*, *38*(2), 155-171. https://doi.org/10.1080/02331939608844245

Hajdu, M. (2015a). Continuous precedence relations for better modelling overlapping activities. *Procedia Engineering*, *123*, 216-223. https://doi.org/10.1016/j.proeng.2015.10.080

Hajdu, M. (2015b). Point-to-point versus traditional precedence relations for modeling activity overlapping. *Procedia Engineering*, *123*, 208-215. https://doi.org/10.1016/j.proeng.2015.10.079

Hajdu, M., & Isaac, S. (2016). Sixty years of project planning: history and future. *Organization, Technology and Management in Construction*, *8*(1), 1499-1510. https://doi.org/10.1515/otmcj-2016-0013

Hajdu, M., Lucko, G., & Su, Y. (2017). Singularity functions for continuous precedence relations and nonlinear activity-time-production functions. *Automation in Construction*, *79*, 31-38. https://doi.org/10.1016/j.autcon.2017.01.012

Harris, R. B., & Ioannou, P. G. (1998). Scheduling projects with repeating activities. *Journal of Construction Engineering and Management*, *124*(4), 269-278. https://doi.org/10.1061/(ASCE)0733-9364(1998)124:4(269)

Hazır, Ö., Erel, E., & Günalay, Y. (2011). Robust optimization models for the discrete time/cost trade-off problem. *International Journal of Production Economics*, *130*(1), 87-95. https://doi.org/10.1016/j.ijpe.2010.11.018

Hegazy, T. (1999). Optimization of construction time-cost tradeoff analysis using genetic algorithms. *Canadian Journal of Civil Engineering*, *26*(6), 685-697. https://doi.org/10.1139/l99-031

Hegazy, T., & Nagib, W. (2001). Cost optimization in projects with repetitive nonserial activities. *Journal of Construction Engineering and Management*, *127*(3), 183-191. https://doi.org/10.1061/(ASCE)0733-9364(2001)127:3(183)

Huang, R.-y., & Sun, K.-S. (2005). System development for non-unit based repetitive project scheduling. *Automation in Construction*, *14*(5), 650-665. https://doi.org/10.1016/j.autcon.2005.02.003

Huang, R.-y., & Sun, K.-S. (2006). Non-unit-based planning and scheduling of repetitive construction projects. *Journal of Construction Engineering and Management*, *132*(6), 585-597. https://doi.org/10.1061/(ASCE)0733-9364(2006)132:6(585)

Huang, Y., Zou, X., & Zhang, L. (2016). Genetic algorithm-based method for the deadline problem in repetitive construction projects considering soft logic. *Journal of Management in Engineering*, *32*(4). https://doi.org/10.1061/(ASCE)ME.1943-5479.0000426

Hyari, K. H., El-Rayes, K., & El-Mashaleh, M. (2009). Automated trade-off between time and cost in planning repetitive construction projects. *Construction Management and Economics*, *27*(8), 749-761. https://doi.org/10.1080/01446190903117793

Ipsilandis, P. G. (2007). Multiobjective linear programming model for scheduling linear repetitive projects. *Journal of Construction Engineering and Management*, *133*(6), 417-424. https://doi.org/10.1061/(ASCE)0733-9364(2007)133:6(417)

Isaac, S., Su, Y., Lucko, G., & Dagan, D. (2017). Work-path modeling and spatial scheduling with singularity functions. *Journal of Computing in Civil Engineering*, *31*(4). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000650

Jeeno, M., Brijesh, P., Dileeplal, J., & Tinjumol, M. (2016). Multi objective optimization for scheduling repetitive projects using GA. *Procedia Technology*, *25*, 1072-1079. https://doi.org/10.1016/j.protcy.2016.08.209

Kelley, J. E., & Walker, M. R. (1959). Critical-path planning and scheduling. In *Eastern Joint IRE-AIEE-ACM Computer Conference*. Boston, Massachusetts, USA. https://doi.org/10.1145/1460299.1460318

Khalied, H., & Khaled, E.-R. (2006). Optimal planning and scheduling for repetitive construction projects. *Journal of Management in Engineering*, *22*(1), 11-19. https://doi.org/10.1061/(ASCE)0742-597X(2006)22:1(11)

Kim, S.-G. (2012). CPM schedule summarizing function of the beeline diagramming method. *Journal of Asian Architecture and Building Engineering*, *11*(2), 367-374. https://doi.org/10.3130/jaabe.11.367

Koo, C., Hong, T., & Kim, S. (2015). An integrated multi-objective optimization model for solving the construction time-cost trade-off problem. *Journal of Civil Engineering and Management*, *21*(3), 323-333. https://doi.org/10.3846/13923730.2013.802733

Li, H., & Love, P. (1997). Using improved genetic algorithms to facilitate time-cost optimization. *Journal of Construction Engineering and Management*, *123*(3), 233-237. https://doi.org/10.1061/(ASCE)0733-9364(1997)123:3(233)

Liberatore, M. J., & Pollack-Johnson, B. (2013). Improving project management decision making by modeling quality, time, and cost continuously. *IEEE Transactions on Engineering Management, 60*(3), 518-528. https://doi.org/10.1109/TEM.2012.2219586

Lim, T.-K., Jang, W.-S., Choi, J.-h., & Lee, D.-E. (2015). Stochastic quality-cost optimization system hybridizing multi-objective genetic algorithm and quality function deployment. *Journal of Civil Engineering and Management, 21*(4), 407-422. https://doi.org/10.3846/13923730.2014.890647

Long, L. D., & Ohsato, A. (2009). A genetic algorithm-based method for scheduling repetitive construction projects. *Automation in Construction, 18*(4), 499-511. https://doi.org/10.1016/j.autcon.2008.11.005

Lucko, G. (2009). Productivity scheduling method: Linear schedule analysis with singularity functions. *Journal of Construction Engineering and Management*, *135*(4), 246-253. https://doi.org/10.1061/(ASCE)0733-9364(2009)135:4(246)

Lucko, G. (2011). Integrating efficient resource optimization and linear schedule analysis with singularity functions. *Journal of Construction Engineering and Management*, *137*(1), 45-55. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000244

Lucko, G. (2014). Temporal constraints in linear scheduling with singularity functions: Case of calendarization. *Journal of Computing in Civil Engineering*, *28*(2), 232-243. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000268

Maghsoudlou, H., Afshar-Nadjafi, B., & Akhavan Niaki, S. T. (2017). Multi-skilled project scheduling with level-dependent rework risk; three multi-objective mechanisms based on cuckoo search. *Applied Soft Computing*, *54*, 46-61. https://doi.org/10.1016/j.asoc.2017.01.024

Maghsoudlou, H., Afshar-Nadjafi, B., & Niaki, S. T. A. (2016). A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem. *Computers & Chemical Engineering*, *88*, 157-169. https://doi.org/10.1016/j.compchemeng.2016.02.018

Panda, A., & Pani, S. (2016). A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. *Applied Soft Computing*, *46*, 344-360. https://doi.org/10.1016/j.asoc.2016.04.030

Photios, G. I., & Yang, I. T. (2016). Repetitive scheduling method: Requirements, modeling, and implementation. *Journal of Construction Engineering and Management*, *142*(5).

Roy, G. B. (1959). Théorie des graphes: Contribution de la théorie des graphes á l1 étude de certains problémes linéaries. *Comptes rendus des Séances de l1 Acedémie des Sciences*, 2437-2449.

Saaty, T. L. (1989). *Group decision making and the AHP*. In B. L. Golden, E. A. Wasil, & P. T. Harker (Eds.) *The Analytic Hierarchy Process*. Berlin Heidelberg: Springer. https://doi.org/10.1007/978-3-642-50244-6_4

Senouci, A. B., & Mubarak, S. A. (2016). Multiobjective optimization model for scheduling of construction projects under extreme weather. *Journal of Civil Engineering and Management*, *22*(3), 373-381. https://doi.org/10.3846/13923730.2014.897968

Shannon, B. T., & Lucko, G. (2012). *Algorithm for time-cost tradeoff analysis in construction projects by aggregating activity-level singularity functions.* In *Proceedings of the 2012 Construction Research Congress*.

Srisuwanrat, C., & Ioannou, P. G. (2007). *Optimal scheduling of probabilistic repetitive projects using completed unit and genetic algorithms.* In *2007 Winter Simulation Conference*. Washington, DC, USA. https://doi.org/10.1109/WSC.2007.4419849

Su, Y., & Lucko, G. (2015a). Optimum present value scheduling based on synthetic cash flow model with singularity functions. *Journal of Construction Engineering and Management*, *141*(11). https://doi.org/10.1061/(ASCE)CO.1943-7862.0001011

Su, Y., & Lucko, G. (2015b). Synthetic cash flow model with singularity functions for unbalanced bidding scenarios. *Construction Management and Economics*, *33*(1), 35-54. https://doi.org/10.1080/01446193.2015.1012527

Su, Y., & Lucko, G. (2016). Linear scheduling with multiple crews based on line-of-balance and productivity scheduling method with singularity functions. *Automation in Construction*, *70*, 38-50. https://doi.org/10.1016/j.autcon.2016.05.011

Tejani, G. G., Savsani, V. J., & Patel, V. K. (2016). Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. *Journal of Computational Design and Engineering*, *3*(3), 226-249. https://doi.org/10.1016/j.jcde.2016.02.003

Tejani, G. G., Savsani, V. J., Bureerat, S., & Patel, V. K. (2018). Topology and size optimization of trusses with static and dynamic bounds by modified symbiotic organisms search. *Journal of Computing in Civil Engineering*, *32*(2). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000741

Tran, D.-H., & Cheng, M.-Y. (2014). Two-phase differential evolution for the multiobjective optimization of time-cost tradeoffs in resource-constrained construction projects. *IEEE Transactions on Engineering Management*, *61*(3), 450-461. https://doi.org/10.1109/TEM.2014.2327512

Tran, D.-H., Cheng, M.-Y., & Prayogo, D. (2016). A novel multiple objective symbiotic organisms search (MOSOS) for time-cost-labor utilization tradeoff problem. *Knowledge-Based Systems*, *94*, 132-145. https://doi.org/10.1016/j.knosys.2015.11.016

Vanhoucke, M. (2006). Work continuity constraints in project scheduling. *Journal of Construction Engineering and Management*, *132*(1), 14-25. https://doi.org/10.1061/(ASCE)0733-9364(2006)132:1(14)

Verma, S., Saha, S., & Mukherjee, V. (2017). A novel symbiotic organisms search algorithm for congestion management in deregulated environment. *Journal of Experimental & Theoretical Artificial Intelligence*, *29*(1), 59-79. https://doi.org/10.1080/0952813X.2015.1116141

Wang, L., & Singh, C. (2009). Reserve-constrained multiarea environmental/economic dispatch based on particle swarm optimization with local search. *Engineering Applications of Artificial Intelligence*, *22*(2), 298-307. https://doi.org/10.1016/j.engappai.2008.07.007

Wang, Y.-N., Wu, L.-H., & Yuan, X.-F. (2010a). Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Computing*, *14*(3), 193-209. https://doi.org/10.1007/s00500-008-0394-9

Wang, Y.-N., Wu, L.-H., & Yuan, X.-F. (2010b). Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, *14*(3), 193-209.

Wu, L. H., Wang, Y. N., Yuan, X. F., & Zhou, S. W. (2010). Environmental/economic power dispatch problem using multi-objective differential evolution algorithm. *Electric Power Systems Research*, *80*(9), 1171-1181. https://doi.org/10.1016/j.epsr.2010.03.010

Zhang, L.-h. (2015). *Repetitive project scheduling: Theory and methods*. Elsevier.

Zheng, D. X. M., Ng, S. T., & Kumaraswamy, M. M. (2005). Applying Pareto ranking and niche formation to genetic algorithm-based multiobjective time-cost optimization. *Journal of Construction Engineering and Management, 131*(1), 81-91. https://doi.org/10.1061/(ASCE)0733-9364(2005)131:1(81)

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257-271. https://doi.org/10.1109/4235.797969

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117-132. https://doi.org/10.1109/TEVC.2003.810758