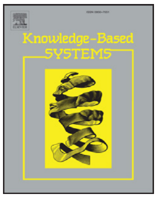
Knowledge-Based Systems 94 (2016) 132–145

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem

Duc-Hoc Tran[a,*], Min-Yuan Cheng[b], Doddy Prayogo[b,c]

[a] Faculty of Project Management, University of Danang—University of Science and Technology, 54 Nguyen Luong Bang, Danang, Vietnam
[b] Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan
[c] Department of Civil Engineering, Petra Christian University, 121-131 Siwalankerto, Surabaya 60236, Indonesia

ARTICLE INFO

Article history:
Received 28 March 2015
Revised 21 November 2015
Accepted 24 November 2015
Available online 1 December 2015

Keywords:
Work shift
Scheduling
Multi-objective analysis
Symbiotic Organisms Search
Time-cost-resource tradeoff

ABSTRACT

Multiple work shifts are commonly utilized in construction projects to meet project requirements. Nevertheless, evening and night shifts raise the risk of adverse events and thus must be used to the minimum extent feasible. Tradeoff optimization among project duration (time), project cost, and the utilization of evening and night work shifts while maintaining with all job logic and resource availability constraints is necessary to enhance overall construction project success. In this study, a novel approach called "Multiple Objective Symbiotic Organisms Search" (MOSOS) to solve multiple work shifts problem is introduced. The MOSOS algorithm is new meta-heuristic based multi-objective optimization techniques inspired by the symbiotic interaction strategies that organisms use to survive in the ecosystem. A numerical case study of construction projects were studied and the performance of MOSOS is evaluated in comparison with other widely used algorithms which includes non-dominated sorting genetic algorithm II (NSGA-II), the multiple objective particle swarm optimization (MOPSO), the multiple objective differential evolution (MODE), and the multiple objective artificial bee colony (MOABC). The numerical results demonstrate MOSOS approach is a powerful search and optimization technique in finding optimization of work shift schedules that is it can assist project managers in selecting appropriate plan for project.

1. Introduction

Labor is a critical construction project resource for construction contractors to be successful on every construction project. Inefficient management of labor resources can result the contractors not able to meet the project deadline and budget requirement. When facing a tight schedule deadline, labor resources has a huge limitation on the number of hours a worker can work per day. Therefore, it requires the use of shift work to meet scheduled deadlines [1]. Using shift work can approximately double the total amount of work hours per day. It also has an advantage over using overtime hours because it prevents worker fatigue and has lower hourly labor costs [2,3]. Furthermore, work shift done during the evening and night is often more efficient due to the quieter, less congested environment around the construction site.

In spite of these advantages, the multiple shift schedules possess several shortcomings including its negative impacts on construction cost, productivity, and safety [1,4]. The multiple shifts might lead in higher overall costs that are required for shift premiums, quality control, nighttime lighting, and safety measures. Additionally, disturbed sleep cycles and stress resulting in higher injury and accident risks, and nighttime construction adversely affects worker health due to circadian rhythm disruption [5–7]. Moreover, recent researches identified that the utilization of evening and night shifts causes higher rates of labor overturn and absenteeism that leads to project delays and cost overruns [2,4]. In order to minimize these negative impacts of utilizing multiple shifts while complying with labor availability constraints, project decision makers need to distribute and utilize the limited labor resources among multiple shifts in the most efficient and effective way to maximize project performance.

Over past decades, a significant amount of research studies have developed optimization models to solve civil engineering problems ranging from structural engineering [8] to construction management [9]. In recent years, there have been notable efforts to solve resource utilization problems using multi-objective optimization models. The most commonly used multi-objective optimization model is the multiple objective genetic algorithm (GA) [10–13]. Other researchers have developed hybrid models of genetic and other algorithms such as particle swarm optimization (PSO) [14], differential evolution (DE) [15] and simulated annealing [16]. However, there are a few reported researches that focus on optimizing the utilization of multiple labor

[*] Corresponding author. Tel.: +84 988922999.
E-mail addresses: tdhoc@dut.udn.vn, duchoc87@gmail.com (D.-H. Tran), myc@mail.ntust.edu.tw (M.-Y. Cheng), doddyprayogo@ymail.com (D. Prayogo).

http://dx.doi.org/10.1016/j.knosys.2015.11.016
0950-7051/© 2015 Elsevier B.V. All rights reserved.

shifts in constructions. Jun and El-Rayes [1] firstly applied a multiple objective genetic algorithm to work shift problem. Therefore, further study is needed to build better optimization models to schedule construction project work shift.

Symbiotic Organisms Search (SOS) is currently one of the most recent metaheuristic algorithms [17]. SOS was first used in a wide variety of highly nonlinear benchmark and engineering problems. The SOS algorithm is simply structured and easy to use, while demonstrating great robustness and fast convergence in solving single objective global optimization problem. Preliminary studies indicate that the new SOS algorithm is superior over the widely used GA, PSO, DE, and bees algorithm (BA) in solving a various continuous benchmark function and engineering problems [17]. Since the SOS algorithm is relatively new, the capability of the SOS algorithm in solving the time cost utilization labor tradeoff (TCUT) problem is very interesting to be further explored and investigated.

This study presents the novel Multiple Objective Symbiotic Organisms Search (MOSOS) algorithm to facilitate a TCUT analysis. The important contribution of this research is that the proposed MOSOS algorithm is a new, multiple objective optimization (MOO) version of the basic SOS algorithm. MOSOS algorithm is developed to fit the TCUT problem because the ability to provide efficient solutions for complex problems simpler operations of SOS is very much attractive and encouraging. The proposed algorithm is designed to attain fast convergence without losing solution diversity on the Pareto front.

The remaining of this paper is organized as follows. In Section 2, the time-cost-utilization resource problem is mathematically formulated. In Section 3, literature related to the establishment of the new optimization model is briefly reviewed. In Section 4, the detailed descriptions of the proposed optimization model for the TCUT problem are presented in details. In Section 5, the performance of the newly developed model is demonstrated using two numerical experiments and result comparisons. Section 6 presents study conclusions.

2. Work shift schedules problem formulation

Using multiple work shifts in a construction project requires that the project planners determine the execution mode of project activities, seek to find the optimal scheduling sequence and assign workers to shifts while satisfying all project constraints. The work shift problem must minimize three contradicting objectives simultaneously including project duration, project cost, and total evening and night shift working hours [1].

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

$$Minimize\ project\ time\ \ T = \sum_{n=1}^{l} T_n^{S_n} = Max_{\forall n}(ES_n + D_n)$$

$$ES_n = \underset{\text{all predecessors } m \text{ of } n}{Maximum} (ES_m + D_m) \quad (1)$$

where $T_n^{S_n}$ is the 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_n 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.

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

$$Minimize\,project\ \text{cost} = \sum_{i=1}^{N} Cost_i^{S_i} \quad (2)$$

where $Cost_i^{S_i}$ is the total cost which includes direct and indirect cost of activity i for a specific option of resources (S_n) and N is the total number of activities.

The third and final objective, minimization of project labor utilization in evening and night shifts, may be calculated as follows:

$$Minimize\,LHEN = LHE + LHN(1+W)\,\text{if SS}$$
$$= 3 \text{ (Three shifts system (SS))} \quad (3)$$

$$LHNE = LHE \ \ \text{if SS} = 2 \text{ (Two shifts system)} \quad (4)$$

$$LHE = \sum_{n=1}^{N} (D_n{}^{*}R_{n,2})^{*}HE \quad (5)$$

$$LHN = \sum_{n=1}^{N} (D_n{}^{*}R_{n,3})^{*}HN \quad (6)$$

where $LHEN$ is the total number of evening and night shift work hours, LHE is the total number of evening shift work hours and LHN is the total number of night shift work hours. Because risks faced in night shift work are typically higher than in other shifts, W is the defined weight that represents the relative importance of minimizing LHN. $R_{n,\,k}$ is the daily labor demand of activity n on shift k. k represents the shift type (e.g., for the 3-shift system, $k = 1$ means day shift, $k = 2$ means evening shift, and $k = 3$ means night shift); HE is the daily evening shift work hours (7.5 h per day); and HN is the daily night shift work hours (7 h per day). In this study, day shift is the period of time for such work during the day (as 8 a.m. to 4 p.m. – 8 h). Evening shift is the work shift during the evening (as 4 p.m. to midnight). Night shift is the work shift during the night (as midnight to 8 a.m.).

3. Literature review

3.1. Review of multiple objective optimization

A MOO problem involves several conflicting objectives simultaneously. The MOO with such conflicting objective functions gives rise to a set of Pareto optimal solutions instead of one optimal solution. Because no one of these solutions can be considered to be better than any other with respect to all objective functions. Generally, the MOO problem consists of n decision variables, k objective functions, m inequality constraints and p equality constraints. It may be mathematically formulated as follows [18–20]:

$$\min_{X \in D} f(X) = [f_1(X), f_2(X), \ldots, f_k(X)] \quad (7)$$

$$\text{s.t}\, g_i(X) \ge 0; \quad i = 1, \ldots, m \quad (8)$$

$$h_j(X) = 0; \quad j = 1, \ldots, p \quad (9)$$

$$D = \{X | g(X) \ge 0, h(X) = 0\} \quad (10)$$

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 equality constraints. 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 (8) and (9).

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 (Deb [18]):

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. $\forall i \in (1, 2, \ldots, k) : f_i(X_1) \le f_i(X_2)$. 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.

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.

Multiple Objective Evolutionary Algorithms (MOEAs) have attracted increasing attention for solving MOO problems [21–24] in recent years. Various researchers from various multi-disciplinary have used MOEAs to solve optimization problems that arise in their own fields [25–27]. As MOO problems become more complex, new MOEAs will continue to emerge.

3.2. Symbiotic Organisms Search algorithm

The SOS algorithm is a new meta-heuristic algorithm developed by Cheng and Prayogo [17]. It is inspired by the biological dependency-based interaction seen among organisms in nature. The dependency-based interaction is often known as symbiosis. Like most population-based meta-heuristic algorithms, SOS shares the similar following features: it uses a population of organisms which contains candidate solutions to seek the global solution over the search space; it has special operators that employ the candidate solutions to guide the searching process; it uses a selection mechanism to preserve the better solutions; it requires a proper setting of common control parameters such as population size and maximum number of evaluations.

However, unlike most meta-heuristic algorithms which have additional control parameters (i.e. GA has crossover and mutation rate; PSO has inertia weight, cognitive factor, and social factor), SOS requires no algorithm-specific parameters. This is considered as an advantage over competing algorithms since SOS does not need additional work for tune the parameters. Improper tuning related to the algorithm-specific parameters might increase the computational time and produce the local optima solution.

In the early stage, a random ecosystem (population) matrix is created, each row representing a candidate solution to the corresponding problem. The number of organisms in the ecosystem, so-called the ecosystem size, is pre-determined by the users. The rows in the matrix are called organisms, same as individuals in other meta-heuristic algorithms. Each virtual organism represents a candidate solution to the corresponding problem/objective. The search begins after the initial ecosystem generated. During the searching process, each organism gains benefit from continuously interacting with one another through three different ways:

1. Mutualism phase: The phase where one organism is developing a relationship that benefits itself and also the other. The interaction between bees and flowers is a classic example to explain the philosophy of mutualism.
2. Commensalism phase: The phase where one organism is developing a relationship that benefits itself while does not impact the other. An example of commensalism is the relationship between remora fish and sharks.
3. Parasitism phase: The phase where one organism is developing a relationship that benefits itself but harms the other. An example of parasitism is the plasmodium parasite, which uses its relationship with the anopheles mosquito to pass between human hosts.

These three phases are adopted from the most common symbioses used by organisms to increase their fitness and survival advantage over the long term. During the interaction, the one who receive a benefit will evolve to a fitter organism while the one who is harmed will perish. The mechanisms for updating the best organism will be conducted after one organism has completed their three phases. The phase will operate until the stopping criterion is achieved. The pseudocode shown in Fig. 1 further summarizes the basic step SOS optimization procedure.

```
Initialization (initial ecosystem, set ecosystem size and maximum iteration)
For counter=1 to maximum iteration
    For each organism in the ecosystem
            Mutualism Phase
            Commensalism Phase
            Parasitism Phase
            Update the best organism
    End For
End For
```

Fig. 1. SOS algorithm pseudocode.

4. The proposed Multiple Objective Symbiotic Organisms Search for time–cost–utilization labor tradeoff model (MOSOS-TCU)

This section describes the Multiple Objective Symbiotic Organisms Search (MOSOS) for solving the TCUT problem developed in this study based on the original SOS algorithm [17]. Fig. 2 shows the overall operational architecture of the proposed algorithm. The following subsections provide further details on the flowchart.

4.1. Ecosystem initialization

This study considers the TCUT problem, in which project cost, project duration, and the utilization of evening and night work shifts are optimized simultaneously. The model requires project information inputs including activity relationship, activity duration (Dur_i), activity cost (C_i), daily labor demand ($R_{i,j}$), shift options (S_i) for each activity, and total number of available labor (RC). In addition, the user also must provide parameter settings for the search engine (MOSOS) such as the value of ecosystem size *ecosize*, number of decision variables D, number of objective functions M, 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 shift options, optimal scheduling sequence and assign available labors to shifts for all construction project activities. With all the necessary information provided, the model is capable of operating automatically without any human intervention.

Population (ecosystem) initialization is the first and the primary task in any optimization algorithm. These two terms, population and ecosystem or, are used interchangeably. Analogous to other population-based algorithms, MOSOS begins with an initial population called the ecosystem. In the initial ecosystem, a group of organisms is generated randomly to the search space as follows:

The initial process generates a point in D-dimensional space $X = \{x_1, x_2, \ldots, x_D\}$ in which $x_1, x_2, \ldots, x_D \in \Re$ and $x_j \in [0, 1]$ have uniform random distributions. The first *ecosize* organisms may be easily generated as follows:

$$X_{i,j}^{G=0} = LB_j + x_{i,j}{}^{*}(UB_j - LB_j);\ i = 1, 2, \ldots, D;\quad j = 1, 2, \ldots, ecosize \tag{11}$$

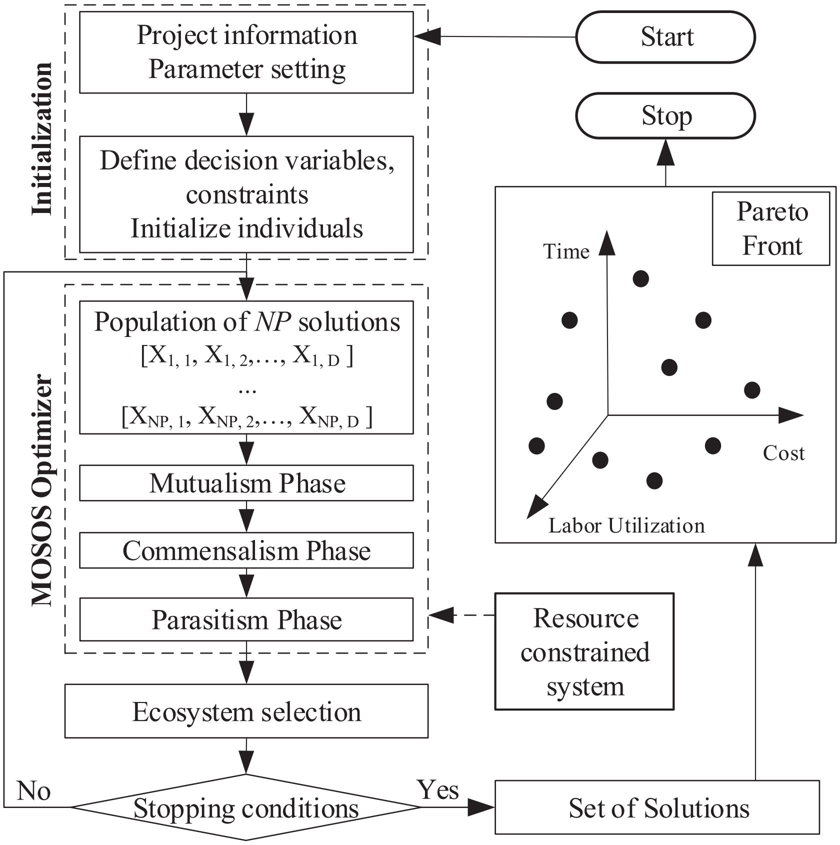


Fig. 2. MOSOS for the TCUT problem flowchart.

4.2. Decision variables and constraints

A candidate solution to the TCUT problems may be represented as a vector with these decision variables: (1) shift option used for each activity; (2) the priority value of each activity; and (3) the labor constraint with (2*N* + 1), (2*N* + 2) elements for two, three shifts system, respectively, as follows:

$$X = \left[\underbrace{x_{i,1}, \ldots, x_{i,j}, \ldots, x_{i,N}}_{\text{Shift-Option } S_n}, \underbrace{x_{i,N+1}, \ldots, x_{i,2N}}_{\text{Priority-value } P_n}, \underbrace{x_{i,2N+1}, x_{i,D}}_{\text{Labor-Constraint } L_k/RC} \right] \tag{12}$$

where *D* is the number of decision variables in the problem at hand. It is obvious that *N* is the number of activities in the project network. Index *i* denotes the *i*th individual in the population.

(1) Shift option: Shift-option (S_n) represents the feasible shift options for activity *n*. Every option has specific combinations of duration, cost and labor demands that lead to different total project durations, total costs and total labor hours. Vector $x_{i,n}$ represents one shift option value for activity *n*. $S_{i,n}$ is an integer number in the range [1,US_n] ($n = 1$ to *N*), meaning one position from US_n shift options. Because the original DE operates with real-value variables, a function is employed to convert the execution mode options of those activities from real values to integer values within the feasible domain.

$$S_{i,n} = Round(x_{i,n} \times US_n); \quad (n = 1, \ldots, N) \tag{13}$$

where $x_{i,\,n}$ is the shift option value of activity *n* at the individual *i*th. US_n represents the total number of shift options for each activity. Round is a function to convert a real number to the nearest integer greater than or equal to it.

(2) Priority value: priority-value (P_n) represents the preference value for each activity in comparison with all other activities. Eq. (14) shows the constraint for this variable.

$$0 \le P_n \le 1; \quad n = N, \ldots, 2N \tag{14}$$

Together with labor constraints and the precedence relationships between activities, P_n values help determine the project scheduling sequence and calculate project duration based the resource constraint subsystem presented in the following subsection.

(3) Labor constraints: labor-constraint (L_k/RC) represents the percentage of total available labors per day for shift *k*. Eq. (15) shows the constraint for this variable.

$$0 \le L_k/RC \le 1; \quad k = 1, K-1 \tag{15}$$

where *RC* is the total number of labors per day available for distribution among all shifts; *k* is shift type; and *K* is the maximum number of allowable shifts per day (e.g. $K = 3$ for three shifts and $K = 2$ for two shifts). This decision variable limits the amount of workers per shift and determines the allocation of available workers. The labor availability (RCS_k) for each shift in the three-shift system may be calculated as follows:

$$MR_k = \max_{n \in All}\{R_{n,k}^{S_n}\} \tag{16}$$

$$REM = RC - \sum_{k=1}^{K} MR_k \tag{17}$$

$$PR_1 = Round(REM^*(L_1/RC)) \tag{18}$$

$$PR_2 = Round\Big((REM - PR_1)^*(L_2/RC)\Big) \tag{19}$$

$$RCS_1 = MR_1 + PR_1 \tag{20}$$

$$RCS_2 = MR_2 + PR_2 \tag{21}$$

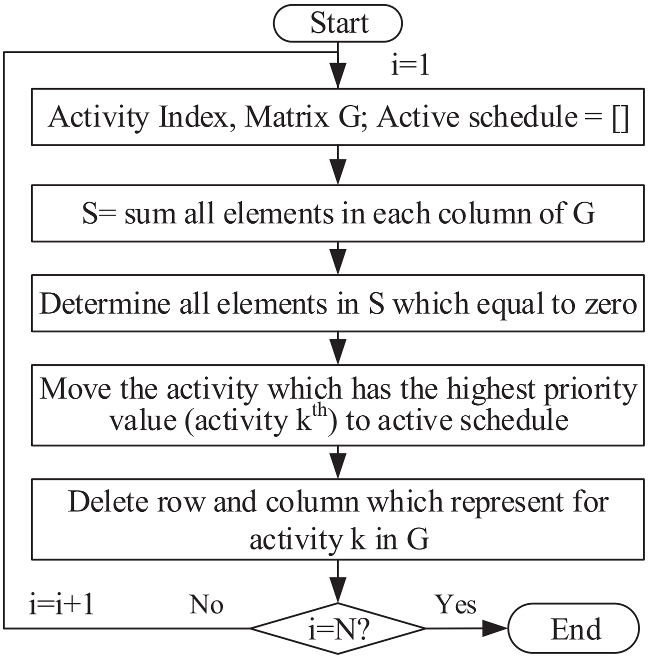

Start
i=1
Activity Index, Matrix G; Active schedule = []
S= sum all elements in each column of G
Determine all elements in S which equal to zero
Move the activity which has the highest priority value (activity kth) to active schedule
Delete row and column which represent for activity k in G
i=i+1
No
i=N?
Yes
End

Fig. 3. Transfer to feasible active schedule.

$$RCS_3 = RC - (RCS_1 + RCS_2) \tag{22}$$

where MR_k is the minimum number of workers for shift *k; REM* is the remaining number of available workers; PR_1 is the additional number of workers available for allocation to the day shift; PR_2 is the additional number of workers available for allocation to the evening shift; RCS_1 is the maximum number of workers allowed to be allocated to the day shift; RCS_2 is the maximum number of workers allowed to be allocated to the evening shift; and RCS_3 is the maximum number of workers allowed to be allocated to the night shift.

4.3. Resource constraint subsystem

Once the MOSOS organism is created, the project duration is calculated through serial method. The shift-option (S_n) values of MOSOS organism defines the execution mode of each activity and then determines the corresponding durations and resource requirements of all activities. The priority value (P_n) of MOSOS organism carries out the sequence of activities. Labor-constraint (L_k) figure out the number of available labors per day for each shift.

The serial method was proposed by Kolisch [28]. It consists of $n = 1,\ldots,N$ stages, in which one activity is selected and scheduled in each stage. When an activity has been checked and currently available amounts of resources are adequate, this activity is scheduled at the earliest precedence time (e.g., the earliest completion times of its predecessors) and resource-feasible time. The serial schedule schema is revised for easier comprehension and implementation using the following two steps:

Step 1: Transfer MOSOS organism sequence of tasks priorities to an active schedule based on precedence constraints.

Denote a set of tasks in project $J = \{1, 2, \ldots, N\}$. We can define priority relations in set *J* as a set of pair $C = \{(i, j)|i$ that must be executed before $j\}$. We introduce the binary relation matrix $V = (v_{ij}, 1 \le i, j \le n)$, $v_{ij} = 1$, if $(i, j) \in C$, $v_{ij} = 0$, if $(i, j) \notin C$, related with a set of priority constraints and define a full-priority relation matrix $G = (g_{ij}, 1 \le i, j \le n)$. This matrix describes all priority relation chains. So, $g_{kj} = 1$ if it is possible to find such a sequence of index pairs that $(k, k_1) \in C$, $(k_1, k_2) \in C$,…,$(k_l, k_j) \in C$. The matrix V has the following property: $v_{ij} = 1 \Rightarrow v_{ji} = 1$. The G matrix shares this feature as well [29]. Fig 3 illustrates the transfer procedure.

Step 2: Calculate project duration based on active schedule

Two important points must be considered before applying the serial method. Firstly, activity A starts when all predecessors are completed (network logic). Secondly, activity A start time depends on resource availability. Thus, activity A is scheduled to start after the completion time of its immediate predecessor on the histogram at the point when sufficient resources are available for activity completion (resource constrained). Fig 4 demonstrates how the serial generation calculates project duration.

The search engine (MOSOS) takes into account the results obtained from the scheduling module and the search for an optimal combination of shift work options, optimal scheduling sequence for each activity and assign available labors to shifts. This research used three contradicting objectives. Section 2 describes the formulae for each objective function.

4.4. Mutualism phase

Let X_i be the organism matched to the *i*th row of the ecosystem population. The organism X_i selects organism X_j as its partner randomly from the ecosystem. Organism X_i is associated to the *j*th row of the ecosystem where *j* is different from *i*. The mutualistic symbiosis between organism X_i and X_j is modeled in Eqs. (23) and (24).

$$X_{inew} = X_i + rand(0, 1)^*(X_{best} - Mutual\_Vector^* BF_1) \tag{23}$$

$$X_{jnew} = X_j + rand(0, 1)^*(X_{best} - Mutual\_Vector^* BF_2) \tag{24}$$

$$Mutual\_Vector = \frac{X_i + X_j}{2} \tag{25}$$

Some notes on the mutualism mathematical model:

1. *rand*(0,1) in Eqs. (23) and (24) is a vector of random numbers between 0 and 1.
2. "*Mutual_Vector*" represents the mutual connection between organism X_i and X_j.
3. X_{best} represents the best organism in an ecosystem. In this model, the X_{best} is arbitrarily chosen among the first non-dominated rank.
4. Organism X_i might benefit significantly when interacting with organism X_j. Meanwhile, organism X_j might only get benefit slightly when interacting with organism X_i. Here, Benefit Factors (BF_1) and (BF_2) are determined stochastically as either 1 or 2. This illustrates whether an organism partially or fully benefits from the interaction.
5. Organisms are evolving to a fitter version only if their new fitness dominates their pre-interaction fitness. In this case, the old X_i and X_j will be replaced immediately by $X_{i\,new}$ and $X_{j\,new}$, respectively. The old X_i and X_j will be moved into advanced population. Otherwise, the $X_{i\,new}$ and $X_{j\,new}$ will be added into advanced population for selecting the next generation ecosystem. In this way, the proposed algorithm can converge faster while maintaining good diversity. Since algorithm may gain some important information from dominated the solution in latter sorting.
6. For each organism X_i, this interaction counts for two function evaluations.

4.5. Commensalism phase

After the mutualism phase is finished, the organism X_i selects again a new partner randomly from the ecosystem, organism X_j. In this circumstance, organism X_i attempts to benefit from the interaction but organism X_j neither benefits nor suffers from the relationship. The commensal symbiosis between organism X_i and X_j is modeled in Eq. (26).

$$X_{inew} = X_i + rand(-1, 1)^*(X_{best} - X_j) \tag{26}$$

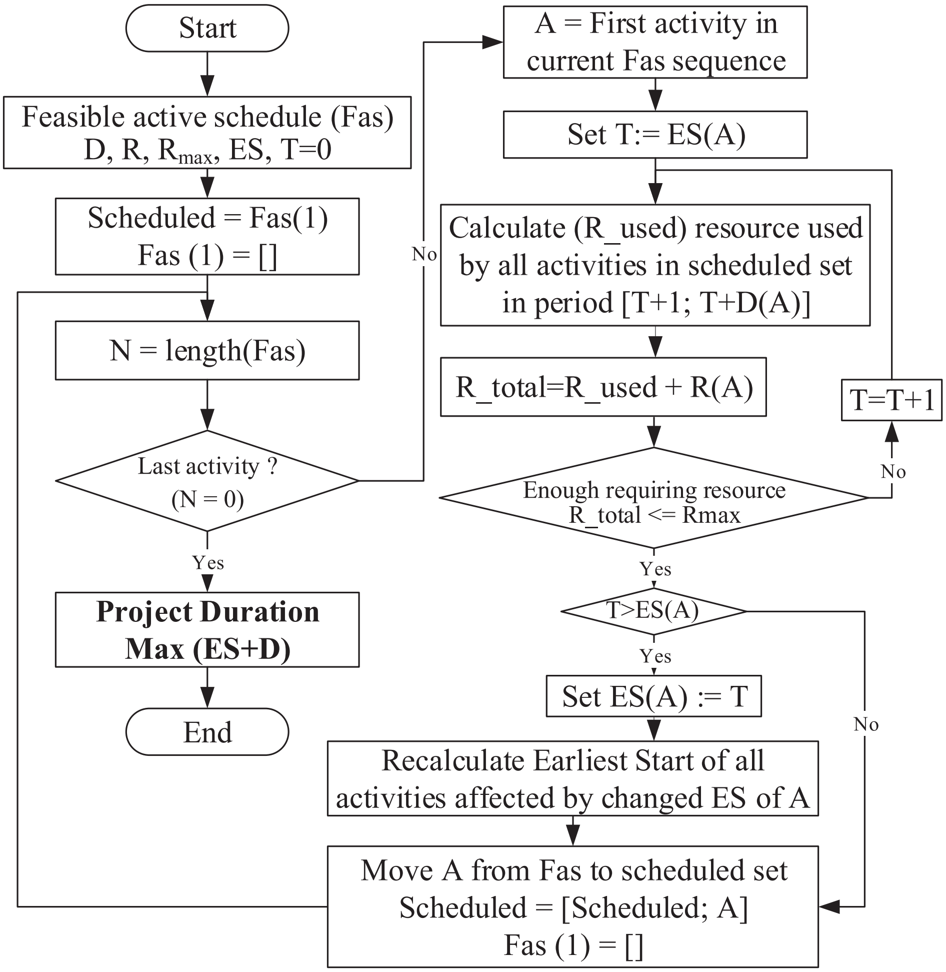


Fig. 4. Serial method.

Some notes on the commensalism mathematical model:

1. *rand*(−1,1) in Eq. (26) is a vector of random numbers between −1 and 1.
2. X_{best} reflects the best organism in the ecosystem, similar to those in the mutualism phase.
3. Organism X_i is updated by $X_{i\,new}$ only if its new fitness dominates its pre-interaction fitness. Then, X_i will be moved into advanced population, otherwise, $X_{i\,new}$. This selection mechanism is analogous to those in the mutualism phase.
4. For each organism X_i, this interaction counts for one function evaluation.

4.6. Parasitism phase

After the commensalism phase is completed, the organism X_i selects again a new organism randomly from the ecosystem, organism X_j. In parasitism, organism X_i is given a role similar to the anopheles mosquito through the creation of an artificial parasite called *"Parasite_Vector"*. Organism X_j serves as a host to the *Parasite_Vector*. During the interaction, the *Parasite_Vector* tries to kill the host X_j and replace X_j in the ecosystem. The organism X_i may gain a benefit, because by cloning it, its influence in the ecosystem may increase while X_j may have to suffer and die.

The creation of *Parasite_Vector* is described as follows:

1. Initial *Parasite_Vector* is created in the search space by duplicating organism X_i. Some decision variables from the initial *Parasite_Vector* will be modified randomly in order to differentiate *Parasite_Vector* with organism X_i.
2. A random number is created within a range from one to the number of decision variables. This random number represents the total number of modified variables.
3. The location of the modified variables is determined stochastically.
4. Finally, the variables are modified using a uniform distribution within the range of the search space. The *Parasite_Vector* is ready for the parasitism phase.

Both *Parasite_Vector* and organism X_j are then evaluated to measure their fitness. If *Parasite_Vector* dominates or non-dominated each other with X_j, it will replace organism X_j in the current ecosystem and X_j will be moved into advanced population. Otherwise, the *Parasite_Vector* will be moved into advanced population. For each organism X_i, this interaction counts for one function evaluation.

4.7. Ecosystem selection procedure

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. [30]. During the optimization process, size of ecosystem remains *ecosize. Ecosize* best (elite) solutions are selected from the combined ecosystem, which mixed of the current and advanced population together. While the "highest fitness value" solution is the best solution

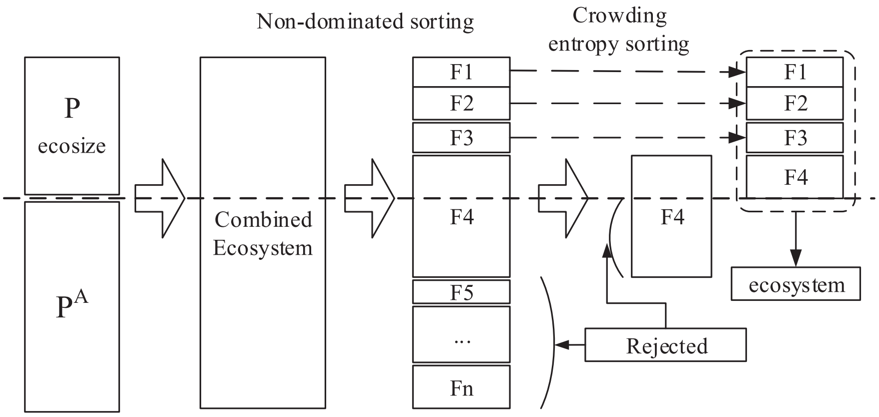


Fig. 5. Ecosystem selection procedure.

in the single objective solution scenario, a two-solution dominance approach is used in multi-objective scenarios. Note that the total size of the combined population is larger than *ecosize*. However, population size during the optimization process remains *ecosize*. Thus, *ecosize* solutions are selected based on the technique as follow. At first, thus, the fast non-dominated sorting technique [31] is employed to sort the combined population into non-dominated sets (F_1, F_2, …). The solutions belonging to the best non-dominated set (Set F_1) are selected first to enter the main population. If size of F_1 is smaller than *ecosize*, the remaining members of the population are chosen from subsequent non-dominated fronts in rank order (F_2, F_3 …). This procedure continues until no further sets can be accommodated. Assume that F_k is the last non-dominated set beyond which no other set can be accommodated. In general, the number of solutions in all sets F_1 to F_k is greater than *ecosize*. To select the optimal *ecosize* population members using crowding entropy sorting technique [32], it is necessary to first fill all population slots in descending order of distance. Fig 5 provides an overview of this procedure.

4.8. Stopping conditions

The optimization process terminates when the stopping conditions are achieved. The user can set these types of 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 ends, the final set of optimal solutions, called the Pareto front, is presented to the user. Obtaining the entire Pareto front is of great importance because it assists planners to evaluate the pros and cons of each potential solution based on qualitative and experience-driven considerations.

5. Case study

This study analyzed a numerical case to demonstrate the effectiveness of the proposed MOSOS for the TCUT problem, with obtained results compared against four approaches also employed to handle the TCUT problem, including NSGA-II, MOPSO, MODE and MOABC. The case project adapted was a previous study of a construction project by Jun and El-Rayes [1]. The project comprised 15 construction activities, each of which has a number of possible shift alternatives. In the case study, a three shifts system (SS = 3) was utilized in combination with a total of 70 available daily labors (RC = 70). The weight for night shift in Eq. (3) was set as $W = 80\%$. Fig. 6 shows the precedence relationships of the network projects. Table 1 illustrates project information data including allowable types of shift operation for each activity (n) and its direct cost, duration, and daily labor demand for each shift and Shift-option (S_n) for project. Project with an average of seven execution modes for each of the 15 activities generate multiple billions 7^{15} of possible combinations for completing the entire projects. Each possible combination has a unique impact on project performance, which means that DMs must search in a large number of potential solutions to find those that establish an optimal tradeoff/balance among construction duration, cost, and the utilization of evening and night work shifts. The newly developed multi-objective optimization model was used to search the many potential solutions.

5.1. Optimization result of MOSOS-TCUT

Since the original SOS is the core mechanism in the proposed MOSOS-TCUT. Only two common control parameters which are population size and maximum number of generations are needed to be

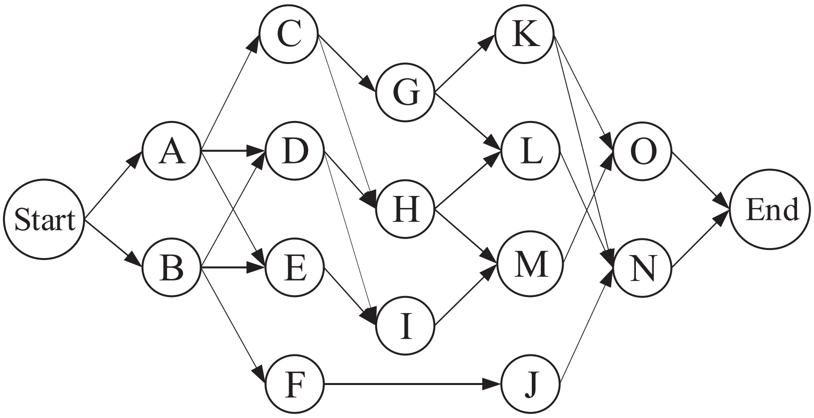


Fig. 6. Network of projects.

Table 1
Case study data.

Act	Shift option	Dur. (Days)	Cost	Worker demand		
				$j = 1$	$j = 2$	$j = 3$
A	1	4	$12,600	4	4	4
	2	5	$11,400	4	4	0
	3	5	$12,700	4	0	4
	4	6	$15,000	0	4	4
	5	8	$11,100	4	0	0
	6	11	$14,800	0	4	0
	7	12	$18,600	0	0	4
B	1	2	$8400	8	8	8
	2	3	$7400	8	8	0
	3	3	$8200	8	0	8
	4	4	$9700	0	8	8
	5	5	$6400	8	0	0
	6	6	$8700	0	8	0
	7	7	$10,900	0	0	8
C	1	3	$13,700	7	7	7
	2	4	$12,400	7	7	0
	3	4	$13,800	7	0	7
	4	5	$16,200	0	7	7
	5	6	$11,600	7	0	0
	6	8	$15,600	0	7	0
	7	9	$19,600	0	0	7
D	1	2	$9100	6	6	6
	2	3	$8200	6	6	0
	3	3	$9100	6	0	6
	4	4	$10,800	0	6	6
	5	5	$7800	6	0	0
	6	6	$10,500	0	6	0
	7	7	$13,100	0	0	6
E	1	5	$18,100	5	5	5
	2	6	$16,400	5	5	0
	3	6	$18,200	5	0	5
	4	7	$21,500	0	5	5
	5	10	$15,700	5	0	0
	6	13	$21,100	0	5	0
	7	15	$26,400	0	0	5
F	1	2	$8600	5	5	5
	2	3	$7800	5	5	0
	3	3	$8700	5	0	5
	4	4	$10,200	0	5	5
	5	5	$7500	5	0	0
	6	7	$10,000	0	5	0
	7	8	$12,600	0	0	5
G	1	3	$6800	4	4	4
	2	4	$6000	4	4	0
	3	5	$6600	4	0	4
	4	5	$7800	0	4	4
	5	8	$5200	4	0	0
	6	10	$7000	0	4	0
	7	11	$8800	0	0	4
H	1	3	$5600	4	4	4
	2	4	$5000	4	4	0
	3	4	$5500	4	0	4
	4	5	$6500	0	4	4
	5	6	$4300	4	0	0
	6	8	$5800	0	4	0
	7	9	$7300	0	0	4
I	1	4	$15,400	8	8	8
	2	5	$13,600	8	8	0
	3	5	$15,100	8	0	8
	4	6	$17,800	0	8	8
	5	8	$11,800	8	0	0
	6	11	$16,000	0	8	0
	7	12	$20,000	0	0	8
J	1	2	$7500	8	8	8
	2	3	$6600	8	8	0
	3	3	$7300	8	0	8
	4	3	$8700	0	8	8
	5	4	$5700	8	0	0
	6	5	$7800	0	8	0
	7	6	$9700	0	0	8
K	1	4	$16,400	6	6	6
	2	5	$14,800	6	6	0
	3	5	$16,500	6	0	6
	4	6	$19,400	0	6	6
	5	8	$14,000	6	0	0
	6	11	$18,800	0	6	0
	7	12	$23,600	0	0	6
L	1	1	$6100	8	8	8
	2	2	$5500	8	8	0
	3	2	$6100	8	0	8
	4	2	$7200	0	8	8
	5	3	$5100	8	0	0
	6	4	$6900	0	8	0
	7	4	$8600	0	0	8
M	1	2	$3300	3	3	3
	2	3	$2900	3	3	0
	3	3	$3300	3	0	3
	4	4	$3900	0	3	3
	5	5	$2500	3	0	0
	6	6	$3500	0	3	0
	7	7	$4300	0	0	3
N	1	4	$13,400	7	7	7
	2	5	$11,900	7	7	0
	3	5	$13,200	7	0	7
	4	6	$15,600	0	7	7
	5	8	$10,300	7	0	0
	6	11	$14,000	0	7	0
	7	12	$17,500	0	0	7
O	1	2	$7700	8	8	8
	2	3	$6800	8	8	0
	3	3	$7500	8	0	8
	4	3	$8900	0	8	8
	5	4	$5900	8	0	0
	6	6	$8000	0	8	0
	7	6	$10,000	0	0	8

Note: shift option $S_n = 1$: three shifts (day, evening, and night shifts), $S_n = 2$: two shifts (day and evening shifts), $S_n = 3$: two shifts (day and night shifts), $S_n = 4$: two shifts (evening and night shifts), $S_n = 5$: one shift (day shift), $S_n = 6$: one shift (evening shift), $S_n = 7$: one shift (night shift).

manually determined in this experiment. These parameters are set based on proposed values form the literature [1,17,33] and several experiments (trial and error) like other metaheuristic algorithms as following settings: the population size was set as 200 and maximum number of generations was set as 300. The other parameters were set as default. Thirty independent optimization runs were conducted. Table 2 describes the first 8 non-dominated solutions in descending order of time, cost, the utilization labor and compromised of case study, respectively, along with optimal shift option combination, scheduling sequence, and labor for each shift. It can be seen that solution 1 generates the smallest project duration value for project while solution 3 and 5 generate the smallest values for cost and total evening and night shift working hours for project, and other solutions strike a balance among the three objectives. Fig. 7 presents schedules of three selected non-dominated solutions (1, 3, and 7) and their corresponding time, cost and total evening and night shift working hours for project. Based on the generated solutions, project managers might choose the optimal solutions for a specific project scenario based on experience, preferences, and specific conditions. If a manager needs to prioritize time, solution 1 is optimal. If a manager needs to prioritize budget and labor utilization in evening and night shifts, solution 3, 5 are optimal. On the other hand, if a manager wants to strike a measured balance between the three objectives, solution 7 provides a centrist solution that offers an acceptable project duration (27 days), a moderate cost ($128700), a moderate total labor hour value in evening and night shifts (998 h) for the project. Fig. 8 shows Pareto fronts obtained using MOSOS for case study. The Pareto front presents the relationships among project duration, cost, and labor utilization. This three-dimensional visualization of the tradeoffs may help project managers evaluate the impact on project performance of the various potential resource-utilization plans.

Table 2
Best non-dominated solutions obtained by MOSOS-TCUT.

Solutions	Partial set	Activity shift-option, scheduling sequence, labor availability in shift *k*	Project performance		
			Time (days)	Cost ($)	LHEN (hours)
1	Sorted by time	{1.2.1.2.1.5.1.2.2.5.1.5.1.1.1}{A.B.F.C.D.E.G.H.J.I.K.L.M.N.O}{32.20.18}	18	144,500	3710
2		{1.2.1.2.2.5.1.2.2.5.1.5.1.2.1}{A.B.F.C.D.E.G.H.J.I.K.L.N.M.O}{32.20.18}	19	141,300	3132
3	Sorted by cost	{5.5.5.5.5.5.5.5.5.5.5.5.5.5.5}{A.B.F.C.D.E.J.G.H.I.K.L.M.N.O}{30.20.20}	38	124,900	0
4		{2.5.5.5.5.5.5.5.5.5.5.5.5.5.5}{A.B.C.D.E.F.J.G.H.I.K.L.M.N.O}{32.26.12}	35	125,200	150
5	Sorted by LHEN	{5.5.5.5.5.5.5.5.5.5.5.5.5.5.5}{A.B.F.C.D.E.J.G.H.I.K.L.M.N.O}{30.20.20}	38	124,900	0
6		{5.5.5.5.5.5.2.5.5.5.5.5.5.5.5}{A.B.F.C.D.E.J.G.H.I.K.L.M.N.O}{38.14.18}	35	125,700	120
7	Compromised	{2.5.5.5.2.5.1.5.5.5.5.5.2.2.5}{A.B.C.D.E.F.J.G.H.I.K.L.M.N.O}{36.19.15}	27	129,500	946
8		{2.5.2.5.2.5.2.5.5.5.2.5.2.5.5}{A.B.C.D.E.F.G.H.J.I.K.L.M.N.O}{35.16.19}	27	128,700	998

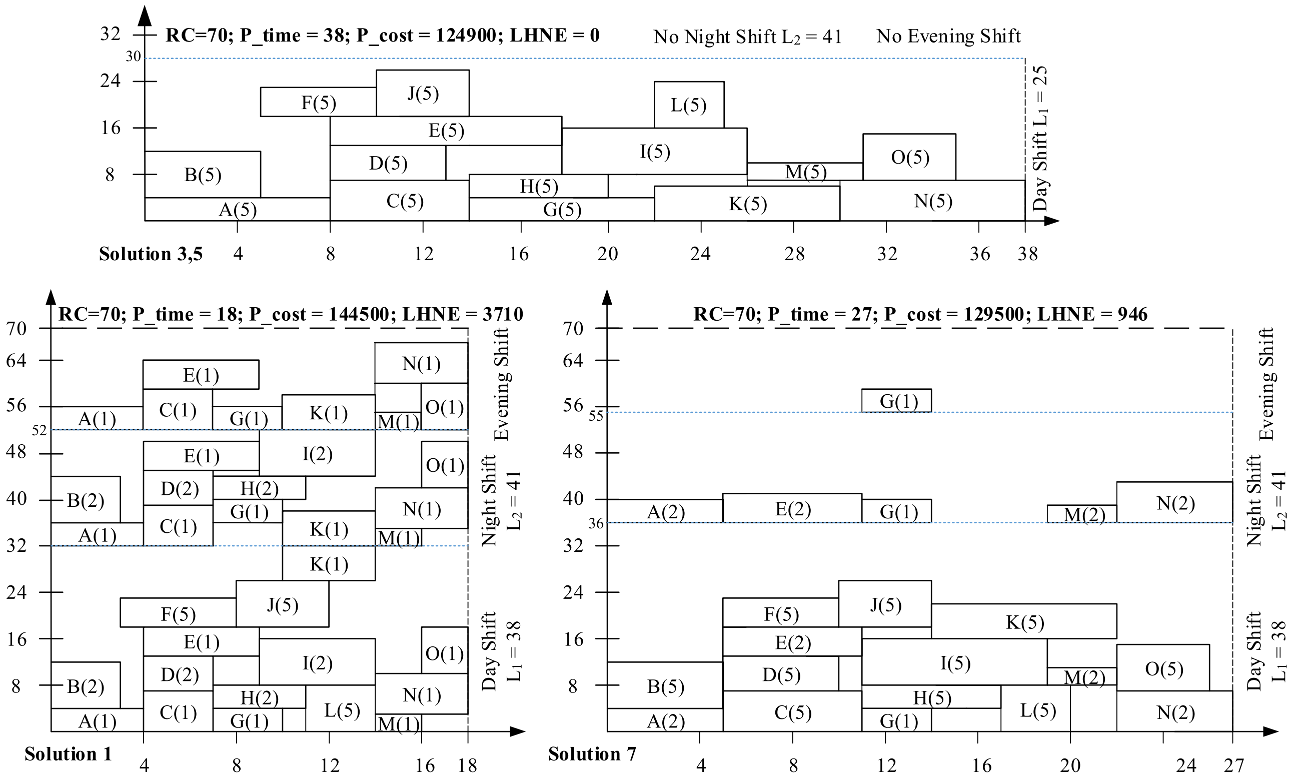


Fig. 7. Schedules related to three non-dominated solutions of case study.

Figs. 8–10 show the two objectives relationship between time and cost, cost and labor utilization, and time and labor utilization, respectively, on a two-dimensional plane. It can be seen from the time-cost curve example (Fig. 9) that lower project funding correlates with longer project completion duration and vice versa. Nevertheless, Figs. 8–10 might not be good representatives of the entire tradeoff surface in the three-dimensional space. In fact, the two-dimensional tradeoff surface, when projected from three to two dimensions, might 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 MOSOS performance against NSGA-II [31], MOPSO [34], MODE [30] and MOABC [35] to assess comparative effectiveness. For comparison purposes, all five algorithms used an equal number of function evaluations, had a population size of 200 and a maximum of 300 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 are both chosen at 2, and the inertia factor w is set in range of 0.3–0.7. In MODE, the crossover probability CR is set to 0.8, and the scaling factor F equals to 0.5. MOSOS control parameters remained the same as stated in previous subsection. Thirty independent runs were carried out for all experiments in case study.

Much research effort has been invested in recent years to develop methods able to evaluate the performance of multi-objective optimization models. Its complex characteristics mean that multi-objective optimization results cannot be evaluated directly, unlike those of single-objective optimization. In the literature, the researchers have suggested numerous quality indicators [31,36–38]. This study used the following four evaluation methods.

1. Number of solutions found in the Pareto-optimal front: the goal of multi-objective optimization is to obtain the Pareto-optimal front that contains the non-dominated solutions of the problem under investigation. No single solution in the Pareto-optimal front may be objectively evaluated as being better than its peers [39]. Therefore, it is preferable to find as many solutions within the Pareto-optimal front as possible.
   Table 3 shows that MOSOS earned the highest number of solutions found in the Pareto front evaluation criterion with 20.9 solutions on average.
2. C-metric (C): C-metric is often used to assess the quality of the true Pareto front of optimized problems [40]. Let S_1, $S_2 \subseteq S$ be

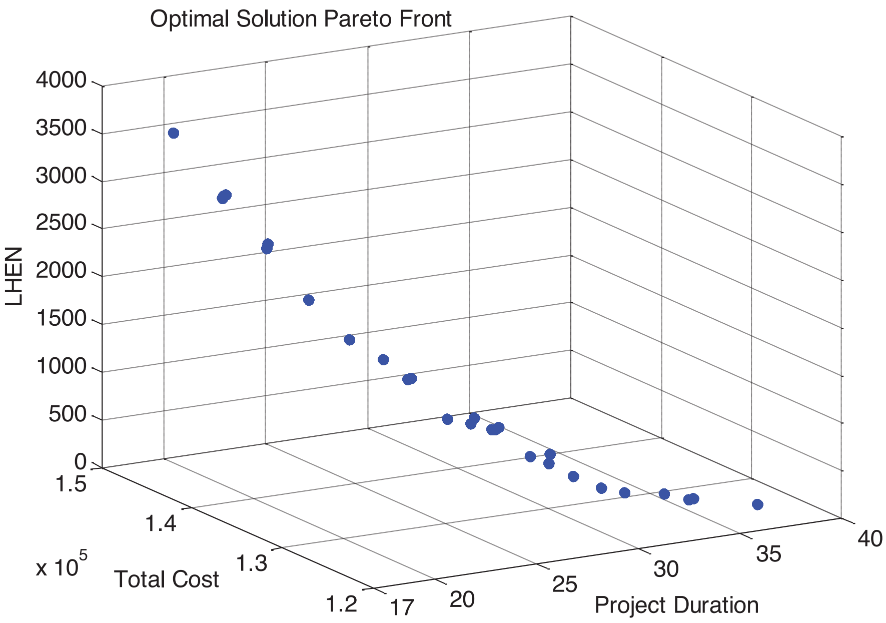


Fig. 8. Time–cost–utilization labor tradeoff Pareto front using MOSOS.

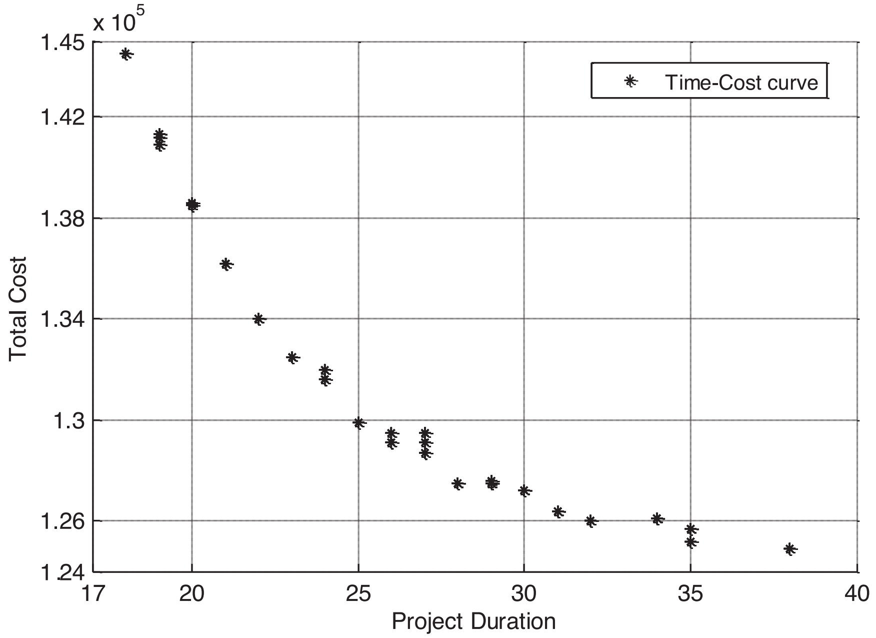


Fig. 9. Time–cost tradeoff curve.

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]:

$$C(S_1,S_2)=\frac{|\{a_2\in S_2;\exists a_1\in S_1:a_1\le a_2\}|}{|S_2|} \tag{24}$$

The numerator in Eq. (24) denotes that the number of solutions in S_2 is dominated by at least one solution in S_1, the denominator is the total solutions in S_2. Provided that $C(S_1,S_2)=1$, all solutions in S_2 are dominated by or equal to solutions in S_1. If $C(S_1,S_2)=0$, then S_1 covers none of the solutions in S_2. Both $C(S_1,S_2)$ and $C(S_2,S_1)$ should be checked in the comparison because C-metric is not symmetrical in its arguments [41]. Table 4 illustrates comparison results among four algorithms in terms of C-metric, where A1, A2, A3, A4 and A5 indicate MOSOS, MODE, MOABC, MOPSO, and NSGA-II, respectively. Results show that MOSOS dominates more than 67.2% of MODE solutions, 94.4% of MODE solutions, 100.0% of MOPSO solutions, and 100.0% of NSGA-II solutions on average.

3. Spread (SP): this indicator [32] measures the extent of spread achieved among the non-dominated solutions. The mathematical definition of SP may be given as:

$$SP=\frac{\sum_{i=1}^{k}d(E_i,\Omega)+\sum_{X\in\Omega}\left|d(X,\Omega)-\overline{d}\right|}{\sum_{i=1}^{k}d(E_i,\Omega)+(|\Omega|-k)\overline{d}} \tag{25}$$

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

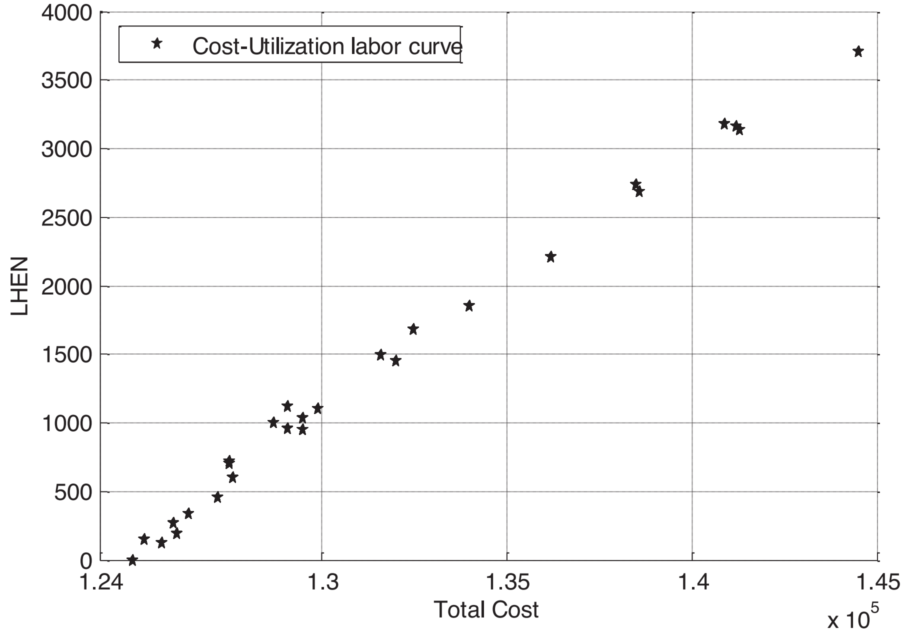


Fig. 10. Cost–utilization labor tradeoff curve.

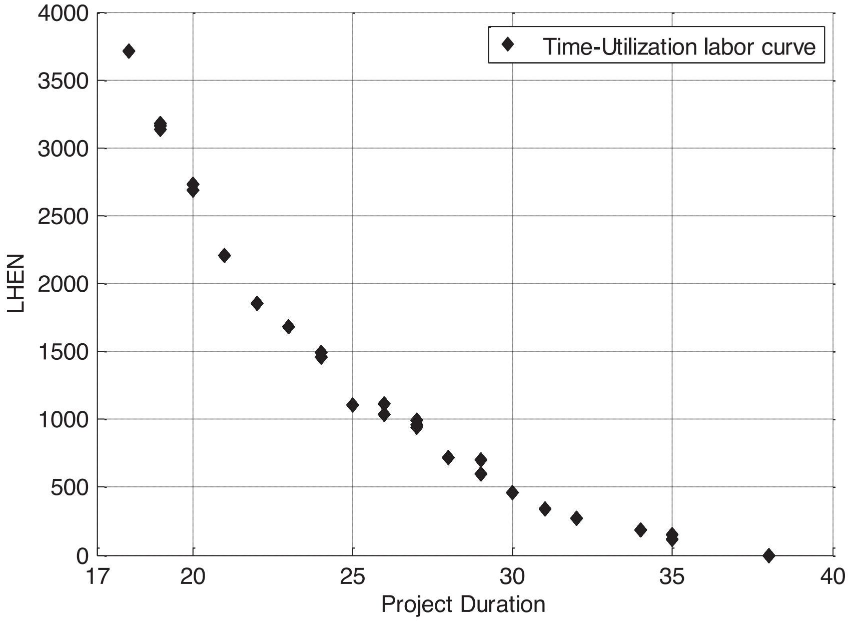


Fig. 11. Time–utilization labor tradeoff curve.

minimum Euclidean distance between solution X and its neighboring solutions in the obtained non-dominated Ω set; $\overline{d} = \frac{1}{|\Omega|}\sum_{X\in\Omega} d(X,\Omega)$ is the mean value of all $d(X, \Omega)$, $|\Omega|$ is the total solutions in Ω set. A value of zero for this metric indicates all members of the 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 different algorithms. This supports that the average performance of the MOSOS is superior to that of the other four algorithms in case study.

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 [37,42]. 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 can be found simply by constructing a vector of worst objective

Table 3
Comparison of numbers of solutions found in Pareto front.

Performance measurement	MOSOS	MODE	MOABC	MOPSO	NSGA-II
Best	25.00	21.00	18.00	14.00	13.00
Worst	16.00	10.00	6.00	7.00	6.00
Average	20.90	17.10	11.20	11.50	9.00
Std.	2.69	3.90	3.88	2.17	2.62

Note: Std. = standard deviation.

Table 4
Comparison of C-metric for different algorithms.

Performance measurement	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	0.900	0.273	1.000	0.080	1.000	0.000	1.000	0.000
Worst	0.455	0.044	0.563	0.000	1.000	0.000	1.000	0.000
Average	0.672	0.158	0.944	0.008	1.000	0.000	1.000	0.000
Std.	0.130	0.076	0.140	0.025	0.000	0.000	0.000	0.000

Table 5
Comparison of SP-metric for different algorithms.

Performance measurement	MOSOS	MODE	MOABC	MOPSO	NSGA-II
Best	0.757	0.592	0.703	0.628	0.602
Worst	1.124	1.142	1.307	1.249	1.445
Average	0.904	0.969	1.088	0.999	0.991
Std.	0.120	0.159	0.191	0.198	0.308

Table 6
Comparison of HV-metric for different algorithms.

Performance measurement	MOSOS	MODE	MOABC	MOPSO	NSGA-II
Best	0.547	0.234	0.028	0.079	0.103
Worst	0.974	0.670	0.391	0.472	0.483
Average	0.796	0.494	0.195	0.248	0.184
Std.	0.126	0.146	0.118	0.121	0.113

function values. Thereafter, a union of all hypercubes is found and its HV is calculated as:

$$HV = \bigcup_{i=1}^{|\Omega|} v_i \tag{26a}$$

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 in case study, which means that MOSOS has better convergence and diversity performance than the other four algorithms.

5.3. Statistical analyses

5.3.1. One tail t-test

A hypothesis test was performed to further demonstrate the superiority of the MOSOS over the other approaches. In all indicators, the hypothesis tests only considered the MOSOS 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. MOSOS versus standard MODE in term of C-metric (Table 4).
H_0: There is no difference in the C-metric of the MOSOS algorithm and that of the MODE algorithm.
H_1: The MOSOS algorithm is significantly better than the MODE algorithm.

MOSOS $s_1 = 0.130$; MODE: $s_2 = 0.076$; $n_1 = n_2 = n = 30$;

$$\nu = \frac{\left(s_1^2/n_1 + s_2^2/n_2\right)^2}{\frac{\left(s_1^2/n_1\right)^2}{n_1-1} + \frac{\left(s_2^2/n_2\right)^2}{n_2-1}} = \frac{\left(0.130^2/30 + 0.076^2/30\right)^2}{\frac{\left(0.130^2/30\right)^2}{30-1} + \frac{\left(0.076^2/30\right)^2}{30-1}}$$
$$= 46.8 \text{ (closest to 47)}$$

Table 7
Hypothesis test results between MOSOS and other approaches.

Indicators	$\bar{x}_1$	$\bar{x}_2$	s_1	s_2	t	ν	$t_{\alpha;\nu} = t_{0.05;\nu}$
C-metric (C)	0.6720	0.1583	0.1301	0.0763	18.655	47	1.678
Spread (SP)	0.9045	0.9685	0.1200	0.1593	−1.758	54	−1.674
Hyper-volume (HV)	0.7957	0.4942	0.1262	0.1458	8.567	57	1.672

Critical value: with significant level of t-test $\alpha = 0.05$; $\nu = 47$; we have $t_{\alpha;\nu} = t_{0.05;47} = 1.678$

$$\text{Statistical test: } t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{s_1^2/n_1 + s_2^2/n_2}} = \frac{(0.6720 - 0.1583)}{\sqrt{0.130^2/30 + 0.076^2/30}}$$
$$= 18.655 > 1.678 = t_{0.05;47}$$

where n is the sample size (number of experimental runs), ν 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 (MOSOS and MODE). 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 MOSOS 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 MOSOS 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 MOSOS produced results that were significantly better than other approaches in terms of the C-metric, spread, and hyper-volume ($t = 18.655 > 1.678 = t_{0.05;47}$; $t = -1.758 < -1.674 = -t_{0.05;54}$ *and* $t = 8.567 > 1.672 = t_{0.05;57}$).

5.3.2. Wilcoxon's signed ranks test

The proposed algorithm is also analyzed statistically with other algorithms using non parametric Wilcoxon's signed ranks test [43]. Wilcoxon's test is defined as follows. Let d_i be the difference between the performance scores of the two algorithms on ith out of n solutions. The differences are ranked according to their absolute values; in case of ties, the practitioner can apply one of the available methods existing in the literature [44] such as ignore ties, assign the highest rank, compute all the possible assignments and average the results obtained in every application of the test, and so on. This study uses the average ranks for dealing with ties (for example, if two differences are tied in the assignation of ranks 1 and 2, assign rank 1.5 to both differences).

Let R^+ be the sum of ranks for the solutions in which the proposed algorithm MOSOS outperformed the compared algorithm, and R^- the sum of ranks for the opposite. Ranks of $d_i = 0$ are split evenly among the sums; if there is an odd number of them, one is ignored:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$
$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{27}$$

Table 8
Wilcoxon test results between MOSOS and other approaches.

Algorithms	Number of solutions found			Spread			Hyper-volume		
MOSOS vs.	R^+	R^-	Critical values	R^+	R^-	Critical values	R^+	R^-	Critical values
MODE	432	33	151	143	322	151	465	0	151
MOABC	465	0	151	40	425	151	465	0	151
MOPSO	465	0	151	137	328	151	465	0	151
NSGA-II	465	0	151	150	315	151	465	0	151

Note: Critical value: $t_{\alpha;n} = t_{0.05;30} = 151$.

Let T be the smaller of the sums, $T = \min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for n degrees of freedom ([45], Table B.12), the null hypothesis of equality of means is rejected; this will mean that proposed algorithm outperforms the other one.

Table 8 displays Wilcoxon's signed ranks test results of proposed algorithm and benchmarked algorithms for number of solutions found, spread and hyper volume indicators, respectively. It can be seen from Table 8 that the MOSOS outperformed the compared algorithms in all indicators since ($T <$ *Critical value*).

6. Conclusions

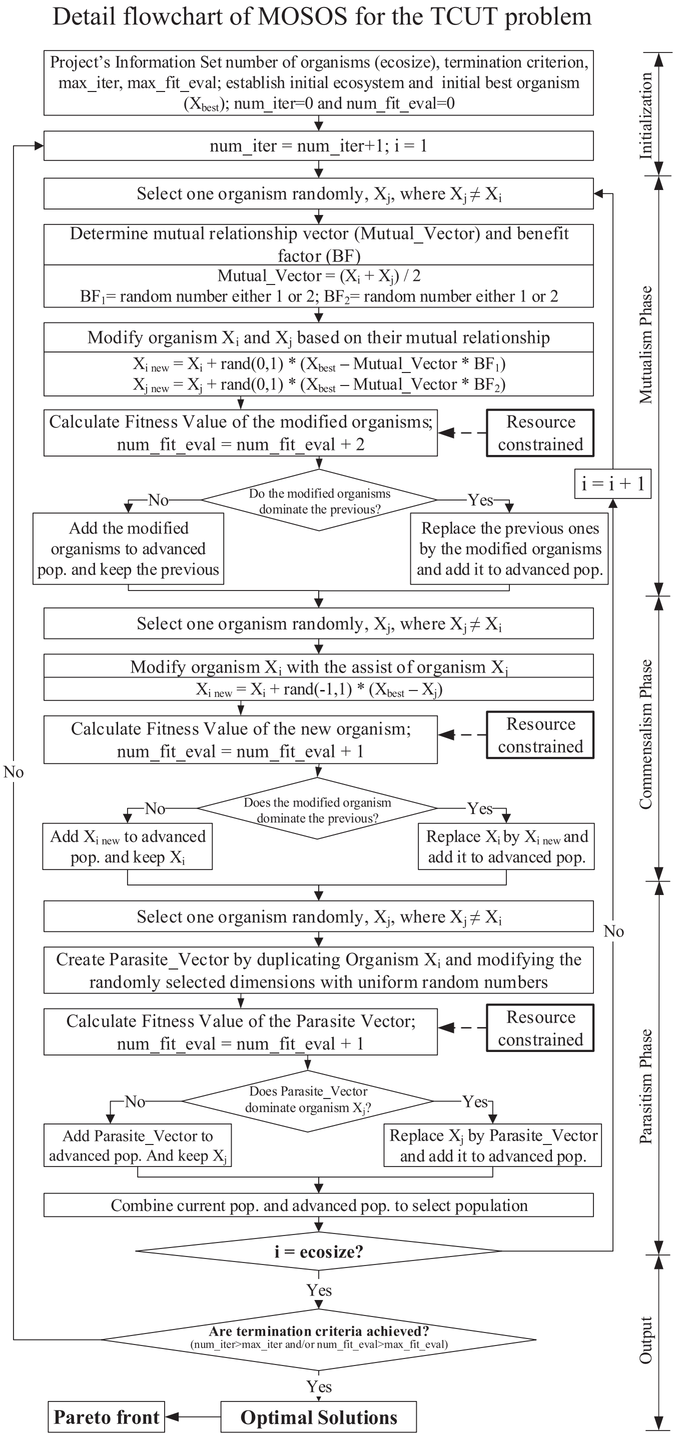
A novel Multiple Objective Symbiotic Organisms Search optimization algorithm has been introduced for optimizing work shift schedules. MOSOS is a population based meta-heuristic algorithm which imitates the biological interactions between organisms in an ecosystem. Three phases of mutualism, commensalism, and parasitism inspire MOSOS to find the non-dominated solutions of given multiple objectives. The proposed algorithm run a construction project to demonstrate its efficacy in finding optimal schedules that simultaneously minimize project duration (time), cost, and the utilization of evening and night work shifts while satisfying with all precedence and labor availability constraints. A project was conducted to illustrate the impact of three shifts systems on project performance. Experimental results shows that the proposed MOSOS approach efficiently solves multi-objective TCUT problems and finds Pareto optimal solutions in one simulation run. Results obtained from the proposed approach have been compared with those obtained from widely used multi-objective evolutionary algorithms such as MOABC, MODE, MOPSO, and NSGA-II. MOSOS displayed better diversity characteristics, yielded better compromise solutions, and attained a higher degree of satisfaction. It is also observed that the proposed approach provides a competitive performance in terms of diversity characteristics, compromise solutions and degree of satisfactions.

The Pareto front generated by MOSOS provides useful information that assists construction-project decision makers determine the optimal tradeoff among the three important project considerations of project duration, cost, and labor utilization.

The proposed MOSOS is simple, robust and efficient. It does not impose any limitation on the number of objectives and can be extended to include more objectives. Further minor modifications of the proposed MOSOS 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; time, cost and safety tradeoffs; and resource-constrained and resource-leveling in project scheduling activities.

Appendix

Detail flowchart of MOSOS for the TCUT problem.


Detail flowchart of MOSOS for the TCUT problem
Project's Information Set number of organisms (ecosize), termination criterion, max_iter, max_fit_eval; establish initial ecosystem and initial best organism (X_best); num_iter=0 and num_fit_eval=0
num_iter = num_iter+1; i = 1
Select one organism randomly, Xj, where Xj ≠ Xi
Determine mutual relationship vector (Mutual_Vector) and benefit factor (BF)
Mutual_Vector = (Xi + Xj) / 2
BF1= random number either 1 or 2; BF2= random number either 1 or 2
Modify organism Xi and Xj based on their mutual relationship
Xi new = Xi + rand(0,1) * (Xbest – Mutual_Vector * BF1)
Xj new = Xj + rand(0,1) * (Xbest – Mutual_Vector * BF2)
Calculate Fitness Value of the modified organisms; num_fit_eval = num_fit_eval + 2
Resource constrained
Do the modified organisms dominate the previous?
No
Yes
Add the modified organisms to advanced pop. and keep the previous
Replace the previous ones by the modified organisms and add it to advanced pop.
i = i + 1
Select one organism randomly, Xj, where Xj ≠ Xi
Modify organism Xi with the assist of organism Xj
Xi new = Xi + rand(-1,1) * (Xbest – Xj)
Calculate Fitness Value of the new organism; num_fit_eval = num_fit_eval + 1
Resource constrained
Does the modified organism dominate the previous?
No
Yes
Add Xi new to advanced pop. and keep Xi
Replace Xi by Xi new and add it to advanced pop.
No
Select one organism randomly, Xj, where Xj ≠ Xi
Create Parasite_Vector by duplicating Organism Xi and modifying the randomly selected dimensions with uniform random numbers
Calculate Fitness Value of the Parasite Vector; num_fit_eval = num_fit_eval + 1
Resource constrained
Does Parasite_Vector dominate organism Xj?
No
Yes
Add Parasite_Vector to advanced pop. And keep Xj
Replace Xj by Parasite_Vector and add it to advanced pop.
Combine current pop. and advanced pop. to select population
i = ecosize?
No
Yes
Are termination criteria achieved?
(num_iter>max_iter and/or num_fit_eval>max_fit_eval)
No
Yes
Pareto front
Optimal Solutions
Initialization
Mutualism Phase
Commensalism Phase
Parasitism Phase
Output

References

[1] D.H. Jun, K. El-Rayes, Optimizing the utilization of multiple labor shifts in construction projects, Autom. Constr. 19 (2010) 109–119.
[2] A. Hanna, C. Chang, K. Sullivan, J. Lackney, Impact of shift work on labor productivity for labor intensive contractor, J. Constr. Eng. Manag. 134 (2008) 197–204.
[3] D. Jun, K. El-Rayes, Optimizing labor utilization in multiple shifts for construction projects, in: Proceedings of Construction Research Congress, 2010, pp. 1185–1193.
[4] U. Spiegel, L.D. Gonen, M. Weber, Duration and optimal number of shifts in the labour market, Appl. Econ. Lett. 21 (2014) 429–432.
[5] A. Hanna, C. Taylor, K. Sullivan, Impact of extended overtime on construction labor productivity, J. Constr. Eng. Manag. 131 (2005) 734–739.
[6] S. Folkard, P. Tucker, Shift work, safety and productivity, Occup. Med. 53 (2003) 95–101.
[7] G. Costa, The impact of shift and night work on health, Appl. Ergon. 27 (1996) 9–16.
[8] M. Aldwaik, H. Adeli, Advances in optimization of highrise building structures, Struct. Multidiscip. Optim. 50 (2014) 899–919.
[9] T.W. Liao, P.J. Egbelu, B.R. Sarker, S.S. Leu, Metaheuristics for project and construction management – a state-of-the-art review, Autom. Constr. 20 (2011) 491–505.
[10] P. Ghoddousi, E. Eshtehardian, S. Jooybanpour, A. Javanmardi, Multi-mode resource-constrained discrete time–cost–resource optimization in project scheduling using non-dominated sorting genetic algorithm, Autom. Constr. 30 (2013) 216–227.
[11] D. Heon Jun, K. El-Rayes, Multiobjective optimization of resource leveling and allocation during construction scheduling, J. Constr. Eng. Manag. 137 (2011) 1080–1088.
[12] K. El-Rayes, A. Kandil, Time–cost–quality trade-off analysis for highway construction, J. Constr. Eng. Manag. 131 (2005) 477–486.
[13] T. Hegazy, Optimization of resource allocation and leveling using genetic algorithms, J. Constr. Eng. Manag. 125 (1999) 167–175.
[14] B. Ashuri, M. Tavakolan, Fuzzy enabled hybrid genetic algorithm–particle swarm optimization approach to solve TCRO problems in construction project planning, J. Constr. Eng. Manag. 138 (2012) 1065–1074.
[15] M.-Y. Cheng, D.-H. Tran, Two-phase differential evolution for the multiobjective optimization of time-cost tradeoffs in resource-constrained construction projects, IEEE Trans. Eng. Manag. 61 (2014) 450–461.
[16] P.-H. Chen, S.M. Shahandashti, Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints, Autom. Constr. 18 (2009) 434–443.
[17] M.-Y. Cheng, D. Prayogo, symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[18] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, 2001.
[19] J. Lu, G. Zhang, D. Ruan, Multi-objective Group Decision Making: Methods, Software and Applications With Fuzzy Set Techniques, Imperial College Press, 2007, p. 408.
[20] D. Cai, W. Yuping, A new uniform evolutionary algorithm based on decomposition and CDAS for many-objective optimization, Knowledge-Based Syst. 85 (2015) 131–142.
[21] S. Hiwa, M. Nishioka,, T. Hiroyasu, M. Miki, Novel search scheme for multiobjective evolutionary algorithms to obtain well-approximated and widely spread Pareto solutions, Swarm Evolut. Comput. 22 (2015) 30–46.
[22] D.-H. Tran, M.-Y. Cheng, M.-T. Cao, Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem, Knowledge-Based Syst. 74 (2015) 176–186.
[23] G. Nicosia, S. Rinaudo, E. Sciacca, An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization, Knowledge-Based Syst. 21 (2008) 175–183.
[24] A.-C. Ză, voianu, E. Lughofer, W. Koppelstätter, G. Weidenholzer, W. Amrhein, E.P. Klement, Performance comparison of generational and steady-state asynchronous multi-objective evolutionary algorithms for computationally-intensive problems, Knowledge-Based Syst. 87 (2015) 47–60.
[25] 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 Evolut. Comput. 1 (2011) 32–49.
[26] J. Xu, X. Song, Multi-objective dynamic layout problem for temporary construction facilities with unequal-area departments under fuzzy random environment, Knowledge-Based Syst. 81 (2015) 30–45.
[27] M. Almasi, M.S. Abadeh, Rare-PEARs: A new multi objective evolutionary algorithm to mine rare and non-redundant quantitative association rules, Knowledge-Based Syst. 89 (2015) 366–384.
[28] R. Kolisch, Serial and parallel resource-constrained project scheduling methods revisited: theory and computation, Eur. J. Oper. Res. 90 (1996) 320–333.
[29] L. Sakalauskas, G. Felinskas, Optimization of resource constrained project schedules by genetic algorithm based on the job priority list, Inf. Technol. Control 35 (2006) 412–418.
[30] M. Ali, P. Siarry, M. Pant, An efficient Differential Evolution based algorithm for solving multi-objective optimization problems, Eur. J. Oper. Res. 217 (2012) 404–416.
[31] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2002) 182–197.
[32] 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. A.: Fusion Found. Methodol. Appl. 14 (2010) 193–209.
[33] M. Cheng, D. Prayogo, D. Tran, Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search, J. Comput. Civil Eng. (2015) 04015036.
[34] I. Yang, Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis, J. Constr. Eng. Manag. 133 (2007) 498–505.
[35] R. Akbari, R. Hedayatzadeh, K. Ziarati, B. Hassanizadeh, A multi-objective artificial bee colony algorithm, Swarm Evolut. Comput. 2 (2012) 39–52.
[36] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput 3 (1995) 1–16.
[37] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G.D. Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2003) 117–132.
[38] C.A. Coello Coello, Evolutionary multi-objective optimization: a historical view of the field, IEEE Comput. Intell. Mag. 1 (2006) 28–36.
[39] K.C. Tan, T.H. Lee, E.F. Khor, Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons, in: Proceedings of the 2001 Congress on Evolutionary Computation, 2001.
[40] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evolut. Comput. 3 (1999) 257–271.
[41] 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.
[42] 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.
[43] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (2009) 617–644.
[44] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evolut. Comput. 1 (2011) 3–18.
[45] J.H. Zar, Biostatistical Analysis, United States of America, Pearson Education, Inc., 2009.