


## Research Article

# Developing Shuffled Frog-Leaping Algorithm (SFLA) Method to Solve Power Load-Constrained TCRTO Problems in Civil Engineering

Xingyu Tao,<sup>1</sup> Heng Li,<sup>2</sup> Chao Mao,<sup>3</sup> Chen Wang ,<sup>4,5</sup> Jeffrey Boon Hui Yap,<sup>6</sup> Samad Sepasgozar,<sup>7</sup> Sara Shirowzhan,<sup>8</sup> and Timothy Rose<sup>9</sup>

<sup>1</sup>School of Construction Management and Real Estate, Chongqing University, Chongqing 400044, China

<sup>2</sup>Chair Professor of Construction Informatics, Department of Building and Real Estate, Faculty of Construction and Environment, Hong Kong Polytechnic University, Room No. ZS734, Hung Hom, Kowloon, Hong Kong

<sup>3</sup>Associate Professor, School of Construction Management and Real Estate, Chongqing University, Room No. 504, Chongqing 400044, China

<sup>4</sup>Professor, College of Civil Engineering, Huaqiao University, 361021 Xiamen, China

<sup>5</sup>Director of Intelligence and Automation in Construction Fujian Province Higher-Educational Engineering Research Centre, 361021 Xiamen, China

<sup>6</sup>Assistant Professor, Department of Surveying, Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman (UTAR), 43000 Kajang, Selangor, Malaysia

<sup>7</sup>Senior Lecturer, Faculty of Built Environment, University of New South Wales Sydney, Kensington, Australia

<sup>8</sup>Research Fellow, Faculty of Built Environment, University of New South Wales Sydney, Kensington, Australia

<sup>9</sup>Senior Lecturer, Faculty of Science and Engineering, Queensland University of Technology, Brisbane, Australia

Correspondence should be addressed to Chen Wang; [derekisleon@gmail.com](mailto:derekisleon@gmail.com)

Received 7 January 2019; Accepted 4 April 2019; Published 2 May 2019

Academic Editor: Roberto Nascimbene

Copyright © 2019 Xingyu Tao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is extensively acknowledged that excessive on-site electricity power load often causes power failure across a construction site and surrounding residential zones and can result in unforeseen schedule delay, construction quality problems, life inconvenience, and even property loss. However, energy management, such as power load optimization, has long been ignored in construction scheduling. This study aims to develop a modified shuffled frog-leaping algorithm (SFLA) approach in project scheduling to aid decision-makers in identifying the best Pareto solution for time-cost-resource trade-off (TCRTO) problems under the constraint of precedence, resource availability, and on-site peak electricity power load. A mathematical model including three objective functions and five constraints was established followed by the application of the modified SLFA on real-case multiobjective optimization problems in construction scheduling. The performance of SLFA was compared with that of the nondominated sorting genetic algorithm (NSGA II). The results showed that the developed new approach was superior in identifying optimal project planning solutions, which could essentially assist on-site power load-oriented schedule decision-making for construction teams.

## 1. Introduction

The objectives of sustainable construction have resulted in increasing demand to adopt strategies to facilitate energy management throughout the entire lifecycle of buildings [1]. Multipronged research efforts, related to technology and policy aspects, have been applied to scheduling energy

supply and consumption in the building operation phase [2, 3]. However, very few previous studies have proposed methods to enable energy optimization in the construction stage, given a lack of awareness in scheduling energy supply and multiple on-site restrictions. Energy management, especially electricity power load optimization, has long been ignored in construction scheduling. With the rapid

development of new construction techniques, an increased amount of electric-powered construction equipment has been adopted on-site, placing a significant burden on the electric power supply. Especially, this occurs when more equipment is used simultaneously to push for target deadlines, resulting in a phenomenon called “electricity snatching.” This results in an increased risk of circuit breaker tripping due to power overload. An on-site power outage can generate negative project impacts, including mechanical breakdowns, schedule delays, and possible quality problems. For instance, the sudden stop of concrete pumps can have a negative impact on the efficiency of pouring and setting time, which is crucial to structural quality. Further, temporary electricity supply on construction sites is often sourced from surrounding residential zones. Peak loading or overloading on construction sites can result in power failure of residential areas, which can result in inconvenience for residents and property loss. The control of on-site power load has become a major issue to both the construction team and the public. Therefore, the need to integrate power load constraints into construction planning poses an obvious and urgent challenge in the construction industry.

Many previous studies have attempted to find solutions to facilitate multiobjective optimization, such as minimization of construction duration, cost, and resource fluctuation [4, 5]. Meanwhile, various methods have been introduced to optimize construction scheduling while considering constraints such as the workforce, resource availability, or construction site layout that have been considered in earlier studies [6–8]. Nevertheless, optimizing and balancing on-site peaking power load has received little attention. Given the advances in electrical technology, on-site power supply reliability has been enhanced. But excessive power load and power failure occur occasionally because contractors usually manage project scheduling and power load requirements separately. The on-site power supply cannot be easily matched with actual construction demand without integrating power load into the construction schedule. A method that can be implemented to simultaneously optimize total duration, total project cost, and resource variation, while considering issues related to resource and power load constraints, is necessary and worth of developing.

To address this research gap, this paper developed a novel shuffled frog-leaping algorithm (SFLA) to solve time-cost-resource trade-off (TCRTO) problems under the constraint of precedence, resource availability, and on-site peak electricity power load. SFLA is a promising and reliable approach for improving project planning solutions since it integrates the strengths of two evolutionary algorithms (i.e., particle swarm optimization and shuffling complex evolution algorithm) simultaneously [9, 10]. By optimizing the combination of on-site electric devices, the power load at any time instant could be automatically controlled within a safety range. The frequency of on-site power outage that led to mechanical breakdown, project quality problems, and inconvenience for surrounding communities was significantly reduced. The developed SFLA could help construction teams to explore better and more effective planning solutions with shorter duration, lower cost, and less resource fluctuation. In specific, this study proposed a mathematical

model to optimize and to integrate power load constraints and developed a SFLA method to solve the power load-constrained TCRTO problems in construction planning. The modified SFLA model was implemented in a real case scenario compared with a typical existing method to assess its optimization capability.

## 2. Construction Scheduling Optimization and Shuffled Frog-Leaping Algorithm

Generally, objectives that need to be optimized usually contain projected duration and financial and resource aspects. Balancing these three parameters is crucial to ensure the achievement of project objectives, namely, minimizing the project duration (TD), project cost (TC), and resource fluctuation (RF). Another aspect of optimization is constraint parameters. Determining appropriate constraints is a critical factor for solving optimization problems. For example, precedence relationship requires an activity can start only when all the previous work has been completed. Further, the start time of each activity must be earlier later than its latest start time [11]. Other limitations, such as resource constraint, space constraint, information constraint, site layout, and safety, have also been considered as factors in recent years. Optimization parameters defined in previous studies are shown in Table 1. Unfortunately, energy management, such as on-site power load scheduling, has not been considered as an optimization parameter in previous studies, despite that uncontrolled on-site electricity power load often causes power failure on construction sites and surrounding residential zones, resulting in inconvenience, property loss, and even safety problems. This is the gap in knowledge that this research is focused on.

### 2.1. Methods for Construction Scheduling Optimization

**2.1.1. Mathematical Methods.** Two mainstream mathematical approaches were used in optimizing construction schedules. (1) Critical path method (CPM): although CPM is widely used in scheduling in the construction industry [23], a major limitation of CPM is that this method cannot deal with optimization problems with multiple objectives [24]. (2) Integer programming (IP) and linear programming (LP): IP, LP, and hybrid IP/LP are analytical algorithms that have been commonly used to solve construction scheduling optimization problems due to their innate efficiency and accuracy [25, 26]. In most cases, IP or LP approaches focus on a single target. A few studies have applied this method in multiobjective optimization problems by providing different weights to different objectives to convert this problem to a single objective problem [27]. However, how to set convincing weights prevents the usage of LP/IP methods in multiobjective optimization problems.

### 2.1.2. Metaheuristic Methods

(1) *GA and NSGA II.* GA is a method to search for finding the optimal solution by simulating the natural evolutionary

TABLE 1: Optimization parameters and methods for the project scheduling problem.

Research studies	Optimization parameters									Optimization method
	Time	Cost	Resource fluctuation	Quality	Cash flow	Resource availability	Site layout	Productivity	Safety	
[11]	√	√								PSO
[12]	√	√								ACA
[4]	√		√			√				Commercial planning software
[13]	√	√				√				GA
[14]			√			√				GA
[15]	√	√	√							GA (NSGA II)
[10]	√	√	√							SFLA
[16]		√	√			√				IP
[17]			√							BCA
[18]	√	√		√		√				GA and evidential reasoning
[19]			√			√				Evolutionary algorithm
[5]	√	√	√							GA (NPGA)
[8]	√	√	√			√				GA
[20]	√					√		√		Metaheuristics
[6]		√					√			GA
[21]	√	√	√		√	√				NSGA II
[22]	√	√				√	√			GA

process [28]. Previous studies have attempted to solve the multimode resource-constrained time-cost trade-off problem using GA or improved GA method [13, 29]. In multiobjective optimization problems, nondominated sorting genetic algorithm (NSGA II) is often used to achieve the Pareto front and a set of optimal solutions can be obtained. The results showed that this method can help construction planners achieve all the projects' multiple objectives under certain limits [21, 30]. GA is an efficient method to obtain an optimal or a set of optimal solutions in construction scheduling. However, improperly selected fitness function and parameters may cause the algorithm to be trapped in local optimal, where global optimal cannot be guaranteed. In addition, it is difficult to determine the stopping criterion of a GA algorithm [31].

(2) *Ant Colony Optimization (ACO)*. Ant colony algorithm uses the ant's walking path to represent the feasible solution of the problem to be optimized, and all the paths of the whole ant colony constitute the solution space of the problem to be optimized. Many studies have used this method to solve single and multiple objective problems [32–34]. Although ACO is a powerful tool, several limitations, including the premature convergence phenomenon, the stopping criterion, and the parameter determining the method, need to be addressed to improve the application of the ACO algorithm.

(3) *Particle Swarm Optimization (PSO)*. PSO is a population-based search process. Due to the poor efficiency of traditional PSO, many researchers have proposed modified PSO for solving the construction scheduling problem by introducing new crossover operators that operate on those coupled particles selected from half of the particle population [11, 35]. However, the selection of the PSO parameters, which are mainly selected based on empirical

results, could cause convergence, divergence, and oscillation of the particles.

2.2. *Shuffled Frog-Leaping Algorithm*. The emergence of shuffled frog-leaping algorithm (SFLA) offers a promising and effective solution for multiobjective and combinatorial optimization problems. Two main steps are involved in SFLA, namely, local search and global search [9]. Specific optimization process of SFLA is illustrated in Section 4.

The performance of SFLA has been compared with that of the genetic algorithm (GA) by testing 11 theoretical test functions and two applications. The result indicates that the SFLA is superior to, or at least, comparable with the GA. Also, SFLA and other existing optimal algorithms were employed in four realistic engineering problems, and the results of SFLA were extremely promising in terms of robustness in finding global solutions and solution speed [36, 37]. The advantages of SFLA have been highlighted within different industrial contexts in recent years. For example, Samuel and Rajan proposed a hybrid particle swarm optimization-based SFLA for solving long-term generation maintenance scheduling problems, and the results demonstrated great superiority of SFLA in searching solutions for combinatorial optimization problems [38]. Luo et al. introduced a novel hybrid SFLA for vehicle routing problems (VRP). The experimental results show that the presented algorithm is very effective for handling VRP when compared with other state-of-the-art heuristics [39]. Kaur et al. presented an augmented SFLA-based technique for resource provisioning and workflow scheduling. The efficacy of ASFLA has been assessed over some well-known scientific workflows, and the simulation results show a marked improvement in the performance criteria of achieving minimum execution cost and meeting schedule deadlines [40].

Compared with other existing optimal algorithms, three main advantages of SFLA can be highlighted: (1) SFLA is more powerful in solving complex combinatorial optimization problems; (2) SFLA has a faster search capability; and most importantly, (3) SFLA is robust in determining the global solution because every memplex evolution and shuffle process can enhance the quality of individuals, thus a better set of solutions is generated.

Despite these advantages, very few studies have adopted this novel method in the context of the construction schedule. Ashuri and Tavakolan established a construction project planning model using SFLA, which allows the interruption of project activities, optimizing three objectives [10]. However, no research has been identified that uses SFLA to enable planners to simultaneously achieve reduced total project duration, less total project cost, and less resource variation under the constraint of precedence relationships, resource availability, and on-site power load.

### 3. Mathematical Model for TCRTO Optimization

**3.1. Duration, Cost, and Resource Model.** The multiple variables to be optimized were the total duration (TD), the total cost (TC), and resource fluctuation (RF), under the constraint of on-site power load, precedence relationships, and resource availability. In this study, the number of construction activities was  $J$  ( $1, 2, \dots, j, \dots, J$ ). Each activity  $j$  could be performed in different methods (modes)  $M_j$  ( $M_j = \{1, 2, \dots, m, \dots, M_j\}$ ), and only one mode  $m$  could be chosen as the execution mode for  $j$ . Once the execution mode of each activity was selected, the corresponding duration, cost, resources requirements, and power load generated by a set of equipment were determined [21]. Parts of the TD, TC, and RF model were based on the study of Ghoddousi et al. [15]:

$$TD = \max : f_j,$$

$$f_j - \sum_{m \in M_j} (x_{jm} \times d_{jm}) \geq f_i,$$

$$TC = \sum_j \sum_{m \in M_j} (x_{jm} \times c_{jm}) + TD \times c_{ind} + y \times c_p \times (TD - T_{contract}),$$

$$RF = \sum_{k=1}^K \sum_{t=1}^{TD} (r_k(t) - \bar{r}_k)^2,$$

$$\bar{r}_k = \frac{1}{TD} \sum_{t=1}^{TD} r_k(t),$$

(1)

where  $f_j$  is the finish time of activity  $j$  in mode  $m$ ;  $x_{jm}$  is a decision variable which equals to 1 when activity  $j$  is executed in mode  $m$ , otherwise  $x_{jm}$  equals to 0;  $d_{jm}$  is the duration of  $j$  in mode  $m$ ;  $i$  are the predecessors activities of  $j$ .  $c_{jm}$  is the direct cost when activity  $j$  performs in mode  $m$ ;  $c_{ind}$

is the indirect cost which is a constant amount;  $T_{contract}$  is the project deadline that is defined in the contract;  $c_p$  is a penalty if delay happens; and  $y$  is a decision variable which equals to 1 when  $TD > T_{contract}$ , otherwise  $y$  equals to 0. The squared deviation is used to evaluate RF;  $k$  represents resource type;  $r_k(t)$  is the resource usage of  $k$  in period  $t \in \{1, 2, \dots, TD\}$ ; and  $\bar{r}_k$  is the average resource usage.

**3.2. On-Site Power Load Constraints.** The power supply system was composed of a set of loops ( $l = 1, 2, \dots, L$ ), where different construction equipment was connected to different loops. Two constraint conditions were proposed: (1) for each time instant  $t$  and for each loop  $l$ , the total power load of loop  $l$  is not allowed to exceed  $P_{peak}^l$ , which is the threshold or peak load of loop  $l$ . (2) For each time instant  $t$ , the power load of all loops is not allowed to exceed  $P_{peak}$ , which means the threshold or peak load of the on-site power supply system. For example, three activities named 1, 2, and 3 were currently processed at time instant  $t$ . Their optional execution mode and the corresponding electric construction equipment are listed in Table 2, respectively. Assuming these three activities are all executed in mode 1, the equipment set, as well as the loop each equipment classification belongs to, can be identified as follows:  $a$ -loop 1,  $b$ -loop 2,  $c$ -loop 1,  $e$ -loop 2,  $f$ -loop 3,  $g$ -loop 1,  $h$ -loop 3, and  $i$ -loop 1. According to the power load constraints, a set of functions are provided as

$$\begin{aligned} P_a^1 + P_c^1 + P_g^1 + P_i^1 &= P^1 \leq P_{peak}^1, \\ P_b^2 + P_e^2 &= P^2 \leq P_{peak}^2, \\ P_f^3 + P_h^3 &= P^3 \leq P_{peak}^3, \\ \sum_{l=1}^3 P^l + P_{extra} &\leq P_{peak}, \end{aligned} \quad (2)$$

where  $P_a^1$  indicates the power load of equipment  $a$  in loop 1 and so on. In general,  $P_a^l = (P_{rate-a})/\epsilon$ , where  $P_{rate-a}$  indicates rate power of  $a$ ;  $\epsilon$  indicates the loss factor ( $\epsilon = 0.9$  for lighting equipment and  $\epsilon = 0.8$  for construction equipment);  $P_{peak}^l$  indicates the threshold or peak load of loop 1, and so on.  $P_{extra}$  indicates the extra power supply by the on-site electric generator.

$$\begin{aligned} \sum_{j \in A_t} \sum_{m \in M_j} (x_{jm} \times P_{jm}^l) &\leq P_{peak}^l, \quad l = 1, 2, \dots, L, \\ A_t &= \{j \mid f_j - d_j < t \leq f_j\}, \end{aligned} \quad (3)$$

$$\begin{aligned} \sum_{l=1}^L \sum_{j \in A_t} \sum_{m \in M_j} (x_{jm} \times P_{jm}^l) + P_{extra} &\leq P_{peak}, \quad l = 1, 2, \dots, L, \\ A_t &= \{j \mid f_j - d_j < t \leq f_j\}. \end{aligned} \quad (4)$$

Constraint set (3) indicated that for each time instant  $t$  and for each loop  $l$ , the power load of the activities which were currently processed (i.e.,  $A_t$ ) and connected to loop  $l$  could not exceed the threshold or peak load of loop  $l$ , where

TABLE 2: Example of on-site power load constraint.

Act ID	Execution mode	Equipment	Loop	Rate power	Loss factor $\epsilon$
1	1	<i>a</i>	1	$P_{\text{rate}-a}$	0.8
		<i>b</i>	2	$P_{\text{rate}-b}$	0.8
		<i>c</i>	1	$P_{\text{rate}-c}$	0.8
	2	<i>d</i>	1	$P_{\text{rate}-d}$	0.8
		<i>c</i>	1	$P_{\text{rate}-c}$	0.8
2	1	<i>e</i>	2	$P_{\text{rate}-e}$	0.8
		<i>f</i>	2	$P_{\text{rate}-f}$	0.8
3	1	<i>g</i>	1	$P_{\text{rate}-g}$	0.8
		<i>h</i>	3	$P_{\text{rate}-h}$	0.8
		<i>i</i>	1	$P_{\text{rate}-i}$	0.8
	2	<i>g</i>	1	$P_{\text{rate}-g}$	0.8
		<i>j</i>	3	$P_{\text{rate}-j}$	0.8

Two power load constraints were presented as follows:

$P_{jm}^l$  was the power load of loop  $l$  generated by activity  $j$ , if it was executed in mode  $m$ . Constraint set (4) indicated that for each time instant  $t$ , the total power load of the activities which were currently processed (i.e.,  $A_t$ ) could not exceed the threshold or peak load of on-site supply.

#### 4. Development of Shuffled Frog-Leaping Algorithm for TCRTO Problems

This effort was to optimize the time-cost-resource trade-off formulated as a multiobjective optimization problem that searches for nondominated schedules to minimize the TD, TC, and RF simultaneously under the constraints of precedence relationship, resource availability, and on-site power load. In this nondominated multiobjective scenario, some of the defined criteria were in conflict with each other, i.e., an improvement in one objective could only be achieved by sacrificing improvement in another. Moreover, some criteria were not commensurable to predefine preferences among the criteria. As a result, the fitness value, which was the key parameter in SFLA, could not be determined for each individual. Due to these reasons, this study developed a modified SFLA method by integrating the advantages of an elitist nondominated sorting genetic algorithm (NSGA II) to search a set of Pareto optimal solutions. To successfully use the developed SFLA method in TCRTO problem, eight key steps were followed: (a) data gathering and setting up preliminary requirements of the model; (b) selection of SFLA parameters; (c) solution encoding; (d) forming Pareto front based on domination relation; (e) partitioning frogs into memplexes; (f) memetic evolution; (g) shuffled complex evolution (SCE); and (h) iteration. These steps were further specified as follows.

**4.1. Data Gathering and Setting Up Model Preliminary Requirements.** The setting up of project parameters included defining the number of activities, precedents and successors of activities, resources and their availability in each period during project execution, execution modes for each activity with corresponding durations, resource requirements and direct costs, the project deadline, and the penalty for the

delay as defined in the contract. Further, the project parameters included a definition of the equipment used in each activity and their corresponding rate power, peak power load values of each loop, and the total circuit load.

**4.2. Selection of SFLA Parameters.** This stage comprised the setting of the size of population  $F$  and setting the iteration number for global exploration and local exploration.

**4.3. Solution Encoding.** The information reflected in each solution or “frog” representation should contain two aspects: (1) the activities’ order for the scheduling process and (2) the mode assignment for activities’ execution. With this in mind, this step considered a chromosome representation used by Ghoddousi [15]. Each individual was represented by a double list: an activity list (AL) and a mode assignment list (MSL). The AL was a precedence feasible list of the activities. The MSL showed the mode selected for each of the activities. Figure 1 shows an example of the structure of a chromosome for a project instance with seven activities.

**4.4. Forming Pareto Front Based on Domination Relation.** TD, TC, and RF were calculated for every solution in the population  $F$ , according to the mathematical model developed by this study. The combined population  $F$  was sorted into nondominated fronts ( $F_1, \dots, F_n$ ) according to the fast nondominated sorting procedure, the first front being a completely nondominant set in the current population, the second front being dominated by the individuals in the first front only, and the fronts go on. Fitness values of these frogs were given based on the fronts they are allocated to (e.g., frogs in the first front were given a fitness value of 1).

**4.5. Partitioning Frogs into Memplexes.** Solutions  $F = m * n$  were randomly partitioned into several memplexes ( $m$ ), each consisting of an equal number of solutions ( $n$ ). Each memplex evolved independently. Figure 2 shows the memplex partition process.

However, in the nondomination multiobjective optimization scenario, the best frog or solution was not unique, since solutions in the Pareto front 1 represented all the best solutions. The random assignment could cause the problem that the best frogs were assigned into the same memplex and the worst frogs were assigned to another memplex, and thus it was meaningless to continue the memetic evolution. Therefore, an improved allocation method was introduced as follows: (a) determining the number of memplexes ( $m$ ) by using the number of frogs in front  $X$ , which contains the largest number of frogs (the number of total fronts was the frogs in each memplex ( $n$ )), and (b) sorting frogs. The first frog in front 1, the first frog in front 2, . . . , the first frog in front  $N$  were assigned to the first memplex; the second frog in front 1, the second frog in front 2, . . . , the second frog in front  $N$  are assigned to the second memplex, and so on. The partition process would not stop until the last frog is assigned in the  $N$ th memplex. If there were not enough

1	2	3	4	5	7	6	→ Activity list (AL)
2	1	1	3	2	1	2	→ Mode assignment list (MSL)

FIGURE 1: Structure of solution.

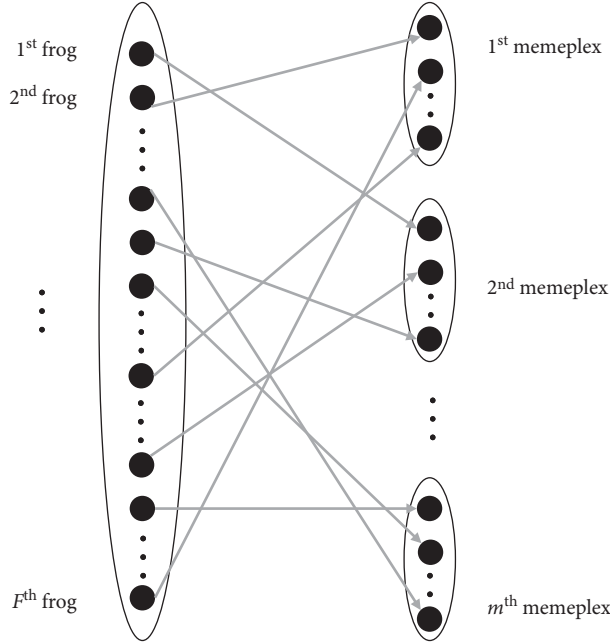


FIGURE 2: Schematic illustration of memplex partitioning process [10].

frogs to fill the memplex, new frogs would be generated randomly to facilitate memetic evolution.

For example,  $F$  contains 12 frogs and forms 3 Pareto front based on domination relation, namely,  $F_1 = \{U_1, U_2, U_3, U_4, U_5\}$ ,  $F_2 = \{U_6, U_7, U_8\}$ , and  $F_3 = \{U_9, U_{10}, U_{11}, U_{12}\}$ .  $F_1$  represents the best front and  $U_1, U_2, U_3, U_4$ , and  $U_5$  are nondominant frogs set in front 1,  $F_2$  represents the moderate front and  $U_6, U_7$ , and  $U_8$  are nondominant frogs set in front 2, and  $F_3$  represents the worst front and  $U_9, U_{10}, U_{11}$ , and  $U_{12}$  are nondominant frogs set in front 3. Since  $F_1$  has the largest number of frogs among all fronts, the number of memplex can be determined as 5. Each memplex contains 3 frogs because there are 3 fronts. Thus, the allocation results (memplexes) can be shown as:  $\{U_1, U_6, U_9\}$ ,  $\{U_2, U_7, U_{10}\}$ ,  $\{U_3, U_8, U_{11}\}$ ,  $\{U_4, U_5, U_{12}\}$ , and  $\{U_5, U_j, U_k\}$ , where  $U_i, U_j$ , and  $U_k$  are new randomly generated solutions. This improved method solves the difficulty of parameter setting. Figure 3 presents the improved method for the memplex partitioning process.

**4.6. Memetic Evolution.** The goal of this process was to improve the quality of feasible solutions through “leaping.” The distance that the worst frog “leaps” was

$$S_i = \text{rand}(0, 1) \times (P_b - P_w), \quad (5)$$

where  $S_i$  is the “leaping” distance and  $P_b$  and  $P_w$  are the best and worst frogs in this memplex, respectively.  $\text{Rand}(0,1)$  is

a random value between 0 and 1. The new position of the worst frog  $P_w(k+1)$  is then computed by

$$P_w(k+1) = P_w(k) + S_i. \quad (6)$$

If this “leaping” does not improve the performance of the worst frog, the “leaping” distance will be recalculated by using the best frog of the whole population  $P_g$ , instead of  $P_b$ . If the position of  $P_w$  has not changed, a new frog will be generated randomly to replace the old  $P_w$ .

As solutions for this TCROT optimization problem must be a set of integers, the “leaping” distance calculated by the method presented above may be nonintegers due to “rand(0,1).” In addition, the structure of the solution, as shown in Figure 1, is not suitable for emetic evolution formulas, since the activities list, as well as the mode assignment list, cannot be pulsed and subtracted.

Therefore, an adaptive evolution method was developed by integrating genetic operators and the one-point crossover method introduced by Hartmann considering the activity list concept, precedence feasibility, and the mode assignment [41]. Specifically, the best frog  $P_b$  and the worst frog  $P_w$  were selected in a memplex and integer  $q_1$  with  $1 \leq q_1 \leq J$  is randomly chosen. The two “improved” frogs (i.e.,  $P_{\text{imp}1}$  and  $P_{\text{imp}2}$ ) activity lists (ALs) and mode lists were generated. The  $q_1$  positions in  $P_{\text{imp}1}$  structure were inherited from  $P_b$ . The positions from  $q_1 + 1$  to  $J$  were inherited from  $P_w$  by a left-to-right scan. In the same way, the  $q_1 + 1$  to  $J$  position of  $P_b$  and 1 to  $q_1$  position of  $P_w$  comprised  $P_{\text{imp}2}$ . These two “improved” frogs were then compared with  $P_w$ , and the best solution was selected as the new frog. If  $P_{\text{imp}1}$  and  $P_{\text{imp}2}$  were not superior to  $P_w$ , another best frog in the population ( $P_g$ ) could be chosen to replace  $P_b$  to repeat the evolution process and another two “improved” frogs,  $P_{\text{imp}3}$  and  $P_{\text{imp}4}$ , are generated. If  $P_{\text{imp}3}$  and  $P_{\text{imp}4}$  were still not superior to  $P_w$ , another new frog could be randomly generated to replace  $P_w$ . Figure 4 shows this adaptive memetic evolution method.

**4.7. Shuffled Complex Evolution (SCE).** This step mixed the memplexes that completed memetic evolution and new memplexes were generated through the shuffling process. New frogs were randomly generated in steps  $e$  and  $f$ ; thus, the number of total frogs was beyond the initial population size  $F$ . Redundant frogs, which were identified as the worst frogs based on new Pareto front formation, could be eliminated to enable the shuffled evolution process. This shuffling enhanced the quality of the memes.

**4.8. Iteration.** This stage involved repeating Steps  $d$ – $g$  until no new project planning solutions were found (i.e., when the new population set of project planning options was equal to the current set of project planning options), and the final population set represented a Pareto optimal set of project

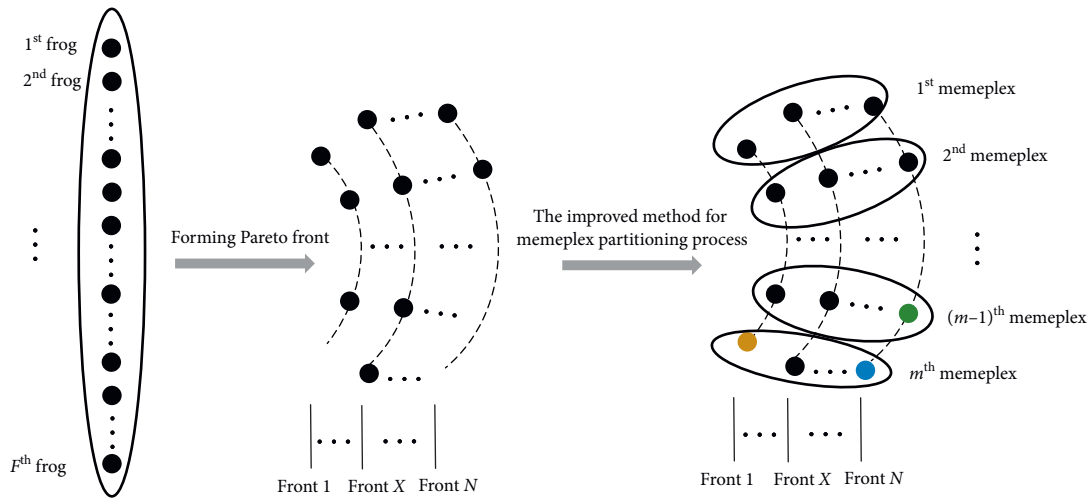


FIGURE 3: The improved method for memplex partitioning process.

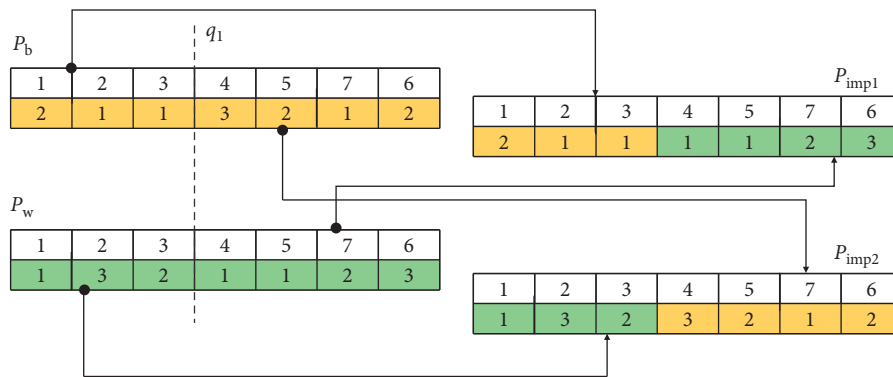


FIGURE 4: The adaptive memetic evolution method.

planning solutions for the TCRTO problem. Figure 5 shows the flow chart of the modified SFLA.

### 5. Case Study

To assess the effectiveness of the developed SFLA method, a simplified construction project case with 18 activities was selected to implement the model. In addition, a comparison study of optimization results between the SFLA method and a common existing optimization method, nondominated sorting genetic algorithm, was conducted. Both methods were used to solve project planning problems and to find Pareto optimal project planning solutions. The SFLA algorithm was coded in MATLAB R2016a. Results showed the superiority of the proposed method.

The activity network of the project is shown in Figure 6, and project information is shown in Table 3. In this case study, different activities had different execution modes from which to select, and each activity  $j$  could be executed in only one mode without splitting. For each mode, corresponding duration, cost and resource requirements, and power loads were presented. The workforce resource was included in the optimization problem and its per period availability was 75. The indirect cost of this project was \$250.

The contract project deadline was 250 days from commencement, and the contractor must pay another \$550 per day when a delay occurred. The threshold of each electric loop was 60 kW and the peak power load value was 188 kW, with an extra generated power supply of 8 kW by an on-site power generator.

**5.1. Implementation of SFLA.** By providing a “time attribute” and “optimal gap” to every activity, the developed SFLA method enabled decision-makers to quickly select effective solutions and to extend the solution range. The “time attribute” defined the early start time and duration of activity  $j$  in mode  $m$ . A traversal approach was adopted in the MATLAB coding process to calculating power load and resource usage at every time instance without the requirement of generating new network diagrams or resource usage histograms, which speeded up the optimization process. The traversal approach only needs to check the activities set at each time instance and calculate the corresponding value. This process is presented in Figure 7, where the broken lines represent the sequence of calculation. In a certain solution,  $j_1$  (from day 6 to 10),  $j_2$  (from day 10 to 20),  $j_3$  (from day 10 to 24), and  $j_4$  (from day 5 to 13) represent

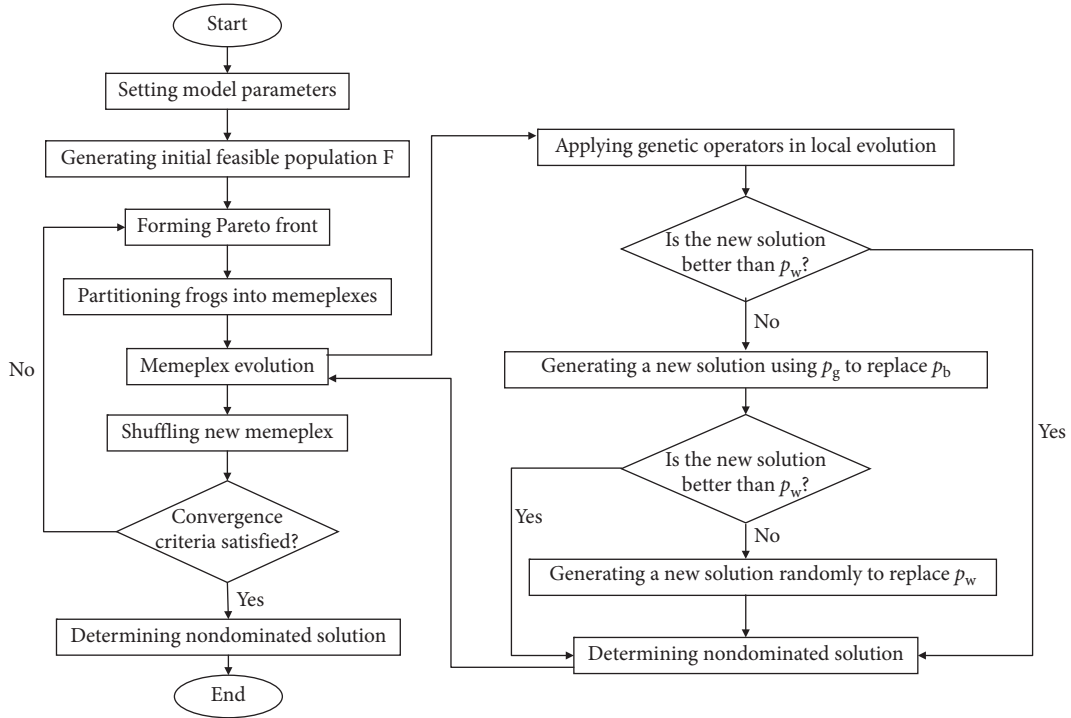


FIGURE 5: Flow chart of developed shuffled frog-leaping algorithm.

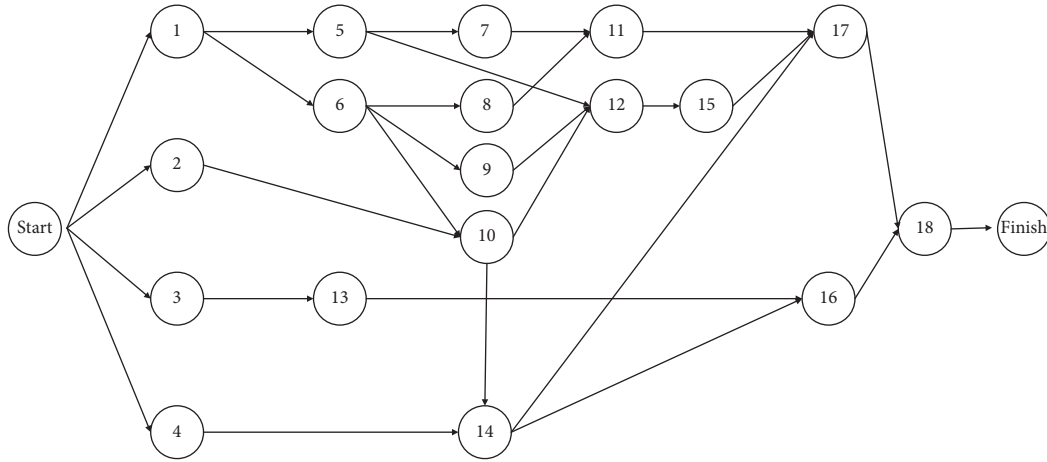


FIGURE 6: Project network activities in the case study.

different activities with their “time attribute.” When the traversal process arrived at  $t=8$ , only the activities  $j_1$  and  $j_4$  should be considered. When  $t=12$ , the resource requirement value and power load of  $j_2$  and  $j_3$  and  $j_4$  should be calculated. Further,  $j_2$  and  $j_3$  should be calculated when  $t=15$ . Instead of providing solutions that did not satisfy the constraints in TCRTO mathematic model (e.g., exceeding resource availability in a time instance) after initial traversal calculation, SFLA converted this invalid value to a valid value by automatically delaying the start time of some activities to balance the power load or resource usage. The delay value “optimal gap” was randomly generated by SFLA so that more effective solutions could be found. For these initial effective solutions (combinations), “optimal gap” was

also used to identify better Pareto fronts. There were three advantages in seeking effective solutions using the SFLA: (a) a faster calculation process by using the traversal approach, (b) broader solution range by converting invalid values to valid values, and (c) better optimization solutions by using the “optimal gap” to facilitate time, cost, and resource fluctuation trade-offs.

Figures 8(a)–8(e) show the process of fast convergence towards the Pareto optimal solutions set, while maintaining diversity in solutions through increasing iteration times. Seven nondominated solutions were identified after the 5th iteration; 9 solutions were identified after the 10th iteration; 13 solutions were identified after the 20th iteration; and 17 solutions were identified after the 50th iteration. It should be



TABLE 3: Activity data of case study.

Activity number	Preceding activities	Execution mode	Time (days)	Cost (\$)	Resource (people)	Equipment type	Equipment number	Rated power (kW)	Electricity loop
1		1	14	2400	8	Eqpt. 1	1	5.5	1
						Eqpt. 2	1	4	2
		2	15	2150	10	Eqpt. 1	1	5.5	1
						Eqpt. 3	1	3.5	2
2		3	16	1900	4	Eqpt. 4	2	6	3
		1	15	3000	12				
		2	18	2400	14				
3		1	15	4500	18	Eqpt. 5	1	7.5	2
						Eqpt. 6	1	3	1
		2	22	4000	15	Eqpt. 5	1	7.5	2
4						Eqpt. 7	1	4	3
		1	12	45000	14	Eqpt. 8	1	32	1
		2	16	35000	12	Eqpt. 8	1	32	1
5	1	1	22	20000	7				
		1	14	40000	21	Eqpt. 8	1	32	1
6	1					Eqpt. 9	2	6.5	2
		2	18	32000	18	Eqpt. 8	1	32	1
						Eqpt. 9	2	6.5	2
		3	24	18000	16	Eqpt. 10	1	24.5	3
7	5					Eqpt. 9	2	6.5	2
		1	9	30000	19	Eqpt. 11	1	6	2
		2	15	24000	16	Eqpt. 11	1	6	2
8	6	1	14	220	8				
		2	15	215	7				
		3	16	200	5				
9	6	1	15	300	11				
		2	18	240	9				
		1	15	450	12	Eqpt. 8	1	32	1
10	2, 6					Eqpt. 9	1	6.5	2
		2	22	400	13	Eqpt. 8	1	32	1
						Eqpt. 9		6.5	2
		3	33	320	14	Eqpt. 8	1	32	1
11	7, 8					Eqpt. 9	1	6.5	2
		1	12	450	4				
		1	22	2000	15	Eqpt. 4	1	6	3
12	5, 9, 10					Eqpt. 3	1	3.5	2
		2	24	1750	14	Eqpt. 4	1	6	3
						Eqpt. 3	1	3.5	2
13	3	1	14	4000	20				
		2	18	3200	22				
		3	24	1800	23				
		1	9	3000	15				
14	4, 10	2	15	2400	17				
		3	18	2200	17				
		1	16	3500	9	Eqpt. 8	1	32	1
15	12					Eqpt. 2	1	4	2
		1	20	3000	10	Eqpt. 8	2	32	1
		2	22	2000	11	Eqpt. 10	2	24.5	3
		3	24	1750	14	Eqpt. 10	2	24.5	3
16	13, 14	1	14	4000	12				
		2	18	3200	14				
		3	24	1800	17				
17	11, 14, 15	2	18	3200	14				
		3	24	1800	17				
		1	9	3000	13	Eqpt. 12	1	9.2	2
18	16, 17	2	15	2400	15	Eqpt. 12	1	9.2	2

mentioned that all 19 solutions were found in the 100th iteration, where Pareto front solutions did not change and convergence criteria were reached. Only approximately 37% of nondominated solutions were achieved after the 5th

iteration. After the 10th iteration, the percentage of the nondominated solution was enhanced to 47%, where dominated solutions were eliminated through iterations and replaced by new-found nondominated solutions. Respectively,

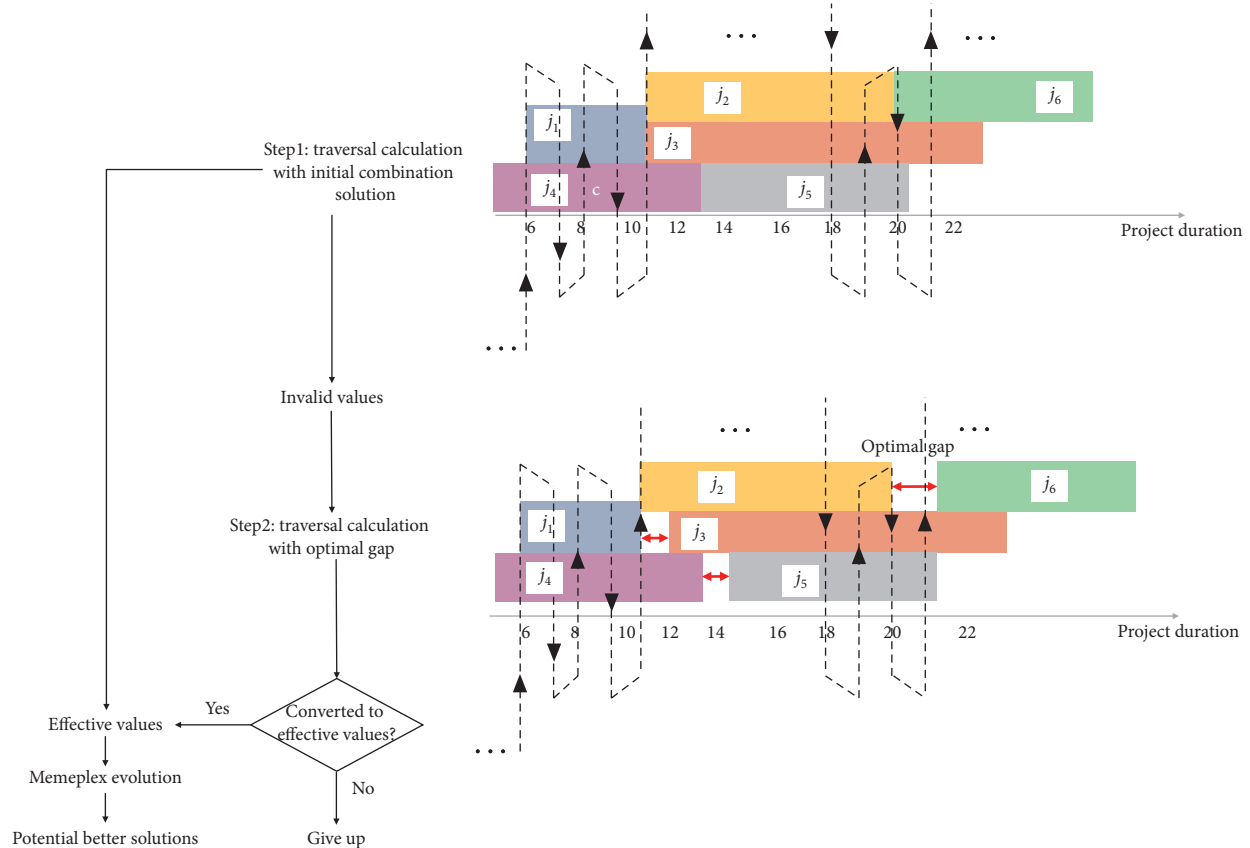


FIGURE 7: Process of seeking effective solutions in SFLA.

60% solutions and 89% solutions were found after 20th and 50th iterations. Finally, by reaching the maximum iteration number of 100 as a termination condition, all nondominated solutions were found and the optimal frog population was formed.

Schedules generated by the modified SFLA and their corresponding time, cost, and resource moment fluctuation are presented in Table 4. The distribution of 19 nondominated points is shown in Figure 8(e), in a three-dimensional space after the 100th iteration. The range of time deviation in Pareto front solutions of the TCROTO model was 141 to 196. The range of cost deviation was \$151540 to \$158618, and the range of resource moment deviation was 3552.20 to 5156.86. In addition to the optimization results, information of these 19 best solutions in the Pareto front can also be obtained. For example, Table 5 shows the specific information of solution no. 1, including the execution mode combination and the start time of each activity.

Data analysis in two dimensions is presented in Figure 9 including the relationship between “TD-TC,” “TD-RF,” and “TC-RF.” An interesting phenomenon occurs where the best solutions may contradict each other. For example, considering solutions no. 2 (178 days, \$200,640) and no. 3 (155 days, \$190,965), solution no. 2, with a large duration (178 days) and greater cost (\$200,640), is still considered as the optimal solution. This is because the results cannot be interpreted by considering only time and cost, while ignoring the third dimension resource variation. Compared

with no. 3 whose RF is 5,204.40, no. 2 remains in optimal population since it has a smaller resource moment deviation (3,716.53) and thus has a smoother resource usage histogram. That is also the reason why there is no obvious trend or regularity between any two objectives.

**5.2. Comparison Study of Modified SFLA and NSGA II.** A comparison study between the modified SFLA method and nondominated sorting genetic algorithm (NSGA II) was conducted with respect to the simultaneous minimization of TD, total TC, and RF. NSGA II was selected as a comparative method because it is a popular and widely used nondominate-based algorithm for multiobjective optimization [42]. The optimal scheduling solutions obtained by NSGA II and the solutions obtained by the SFLA are presented in the same three-dimensional space in Figure 10, where the number of solutions generated by SFLA are larger than that of NSGA II (i.e., 31 solutions were generated by the modified SFLA, while 10 solutions were generated by NSGA II). To demonstrate the advantage of the proposed algorithm, the data analysis is presented in Figure 11, where optimal project planning solutions derived by the two algorithms are displayed in the two-dimensional space of project planning objectives. The results show that the modified SFLA can find optimal solutions with the shortest TD (i.e., 141 days compared to 143 days obtained by the NSGA II algorithm) and less total variation of resources

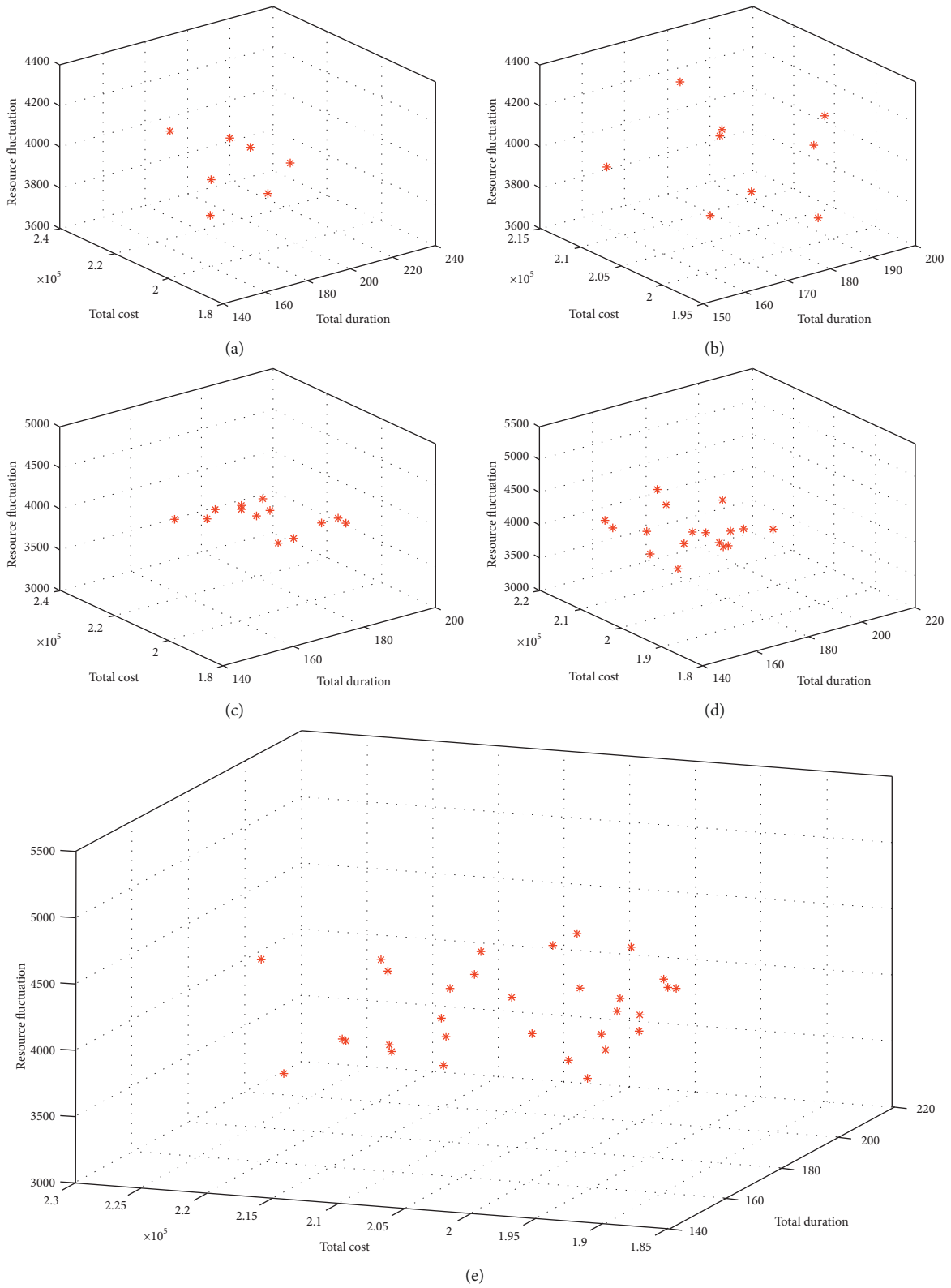


FIGURE 8: Optimal solutions generated by SFLA. (a) Nondominated solutions after 5th generation. (b) Nondominated solutions after 10th generation. (c) Nondominated solutions after 20th generation. (d) Nondominated solutions after 50th generation. (e) Pareto front solutions of the TCRTO model after 100th generation.

TABLE 4: Optimal solutions for case study.

Solution	Total duration	Total cost	Resource fluctuation
<i>Solutions of SFLA</i>			
1	162	189665	4536.24
2	178	200640	3716.53
3	155	190965	5204.40
4	169	220420	3802.89
5	145	207850	4935.05
6	180	199640	3561.16
7	157	213350	4222.84
8	164	214605	4128.29
9	167	195800	4404.09
10	151	203850	4668.06
11	141	216100	5294.22
12	157	195300	4989.14
13	163	193510	4594.50
14	159	210190	4305.84
15	163	207140	4615.77
16	147	193350	5033.71
17	177	195020	4175.24
18	175	197225	4008.82
19	151	208305	4258.49
20	172	195700	4403.29
21	205	215760	3464.22
22	158	189120	4219.71
23	142	199600	4986.46
24	169	201510	4151.20
25	146	195005	5136.67
26	165	207405	4153.21
27	149	201580	4906.96
28	160	201080	4846.13
29	161	188840	4917.31
30	180	195700	4257.91
31	149	208190	4607.44
<i>Solutions of NSGA II</i>			
1	183	201210	4033.26
2	155	188740	4806.04
3	179	223555	3761.24
4	143	224860	5170.60
5	145	225655	5114.27
6	143	224860	5245.04
7	163	207790	4725.61
8	170	196090	4638.32
9	170	220005	4156.89
10	152	214455	5001.21

allocation (i.e., 3,464.22 compared to 3,761.24 obtained by the NSGA II algorithm). Although the slightly lower total project cost (\$ 188,740) was obtained by NSGA II in comparison to the SFLA (\$188,840), more solutions with lower cost (and less duration) were obtained by the SFLA. For example, in the “TD-TC” relationship in Figure 11, when the duration is less than 150 days, solutions obtained by SFLA were superior to that of NSGA II, due to the lower cost. In “TD-RF” and “TC-RF” relationships, it is clear that with the same, or near the same, value on the X-axis, the position of most “blue points” is beyond the lowest “red points.” This indicates that in relation to duration or cost solutions, the SFLA can always find solutions that have a lower resource fluctuation than that obtained by NSGA II. The two most significant contributions of the proposed

TABLE 5: Specific information of solution no. 1

Activity number	Start time	Execution mode
1	23	1
2	9	2
3	14	2
4	8	1
5	39	3
6	45	2
7	75	2
8	69	2
9	76	1
10	74	1
11	106	1
12	91	1
13	37	3
14	98	1
15	113	1
16	123	3
17	129	3
18	153	1

algorithm over NSGA II are that the modified SFLA can find better solutions for multiobjective problems and is capable of finding additional optimal project scheduling solutions. This is because more possible frogs (solutions) have the chance to participate in memplex evolution, due to the improved method to convert some invalid solutions into effective solutions by using the “optimal gap,” and as a result, more solutions are obtained. Instead of random evolution used in NSGA II, memetic evolution in SFLA ensures “worst solutions” can “leap” to a better position in every evolution process, which enables the identification of more optimal solutions.

## 6. Conclusion and Future Work

Uncontrolled on-site electricity power loading can often result in major site disruption including mechanical breakdown, schedule delays, and increased quality risks. As an exploratory effort to improve on-site power load scheduling, a shuffled frog-leaping algorithm (SFLA) approach that is commonly applied in solving complex combinatorial optimization problems was developed in this paper to aid decision-makers in identifying the best Pareto solution for time-cost-resource trade-off (TCRTO) problems under the constraints of precedence, resource availability, and on-site peak electricity power load. Improvements have been made to the SFLA method to suit the research context (i.e., an adaptive frog presentation, a new method for partitioning frogs into memplexes and an improved memetic evolution process). The effectiveness of the developed SFLA method was verified through a real case and a comparison study between SFLA and a nondominated sorting genetic algorithm (NSGA II) indicating that the developed method could identify a larger amount and a wider range of optimal solutions, establishing the potential and value-addedness of this modified SFLA in delivering better power load optimization. The adoption of this novel SFLA in the planning of on-site power loading can provide

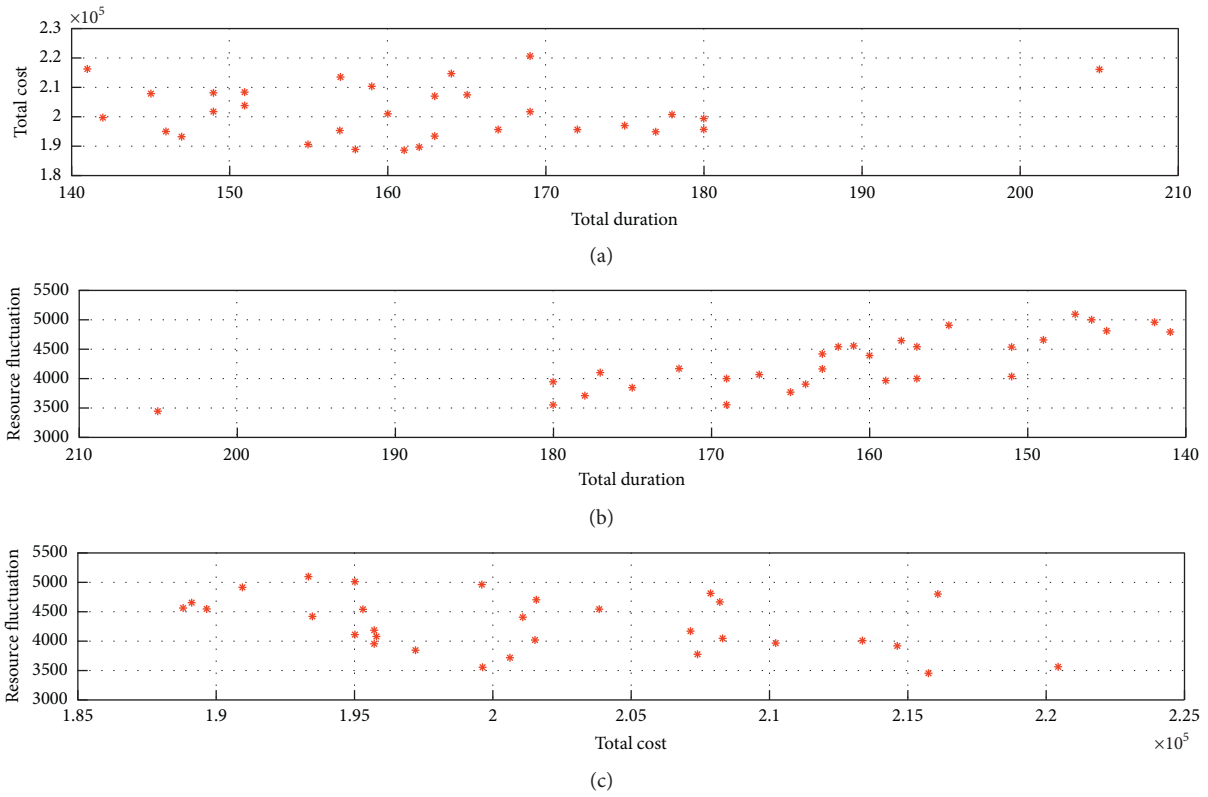


FIGURE 9: Optimal solutions in two-dimensional space.

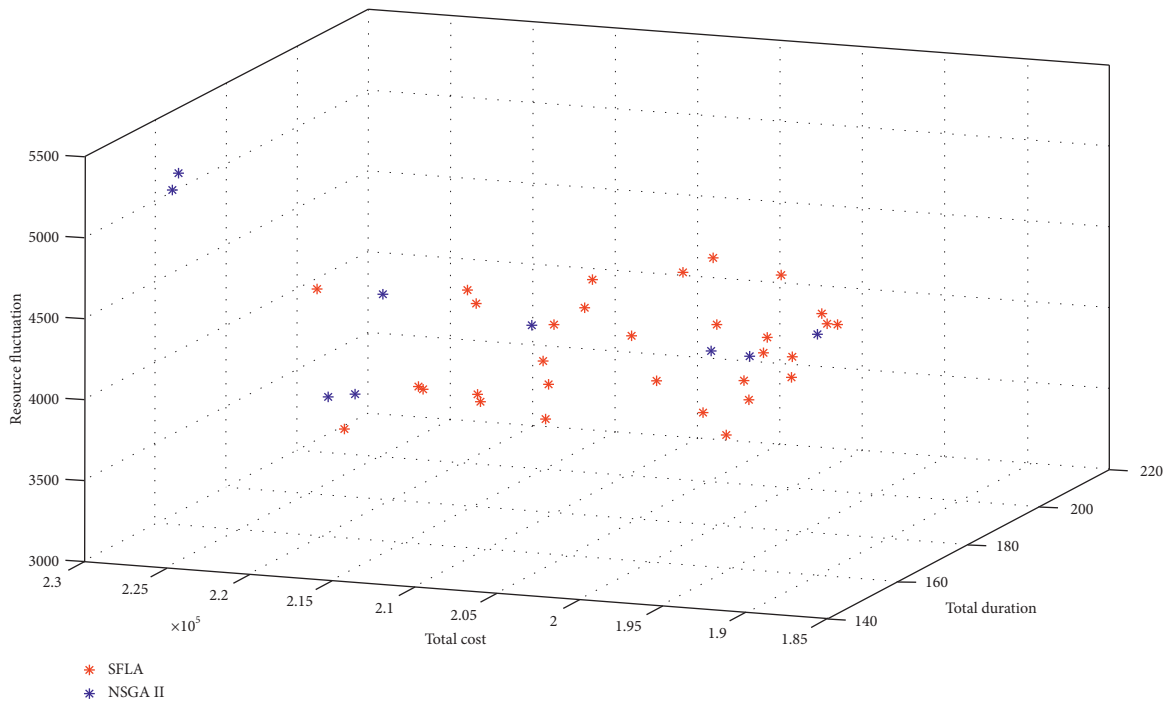


FIGURE 10: Optimal solutions generated by SFLA and NSGA II.

strong theoretical and technical support through enhanced energy management solutions to guide construction teams to achieve both financial and environmental objectives.

The developed SFLA has limitations. Only precedence, resource availability, and on-site peak electricity power load constraints have been considered. Future research should

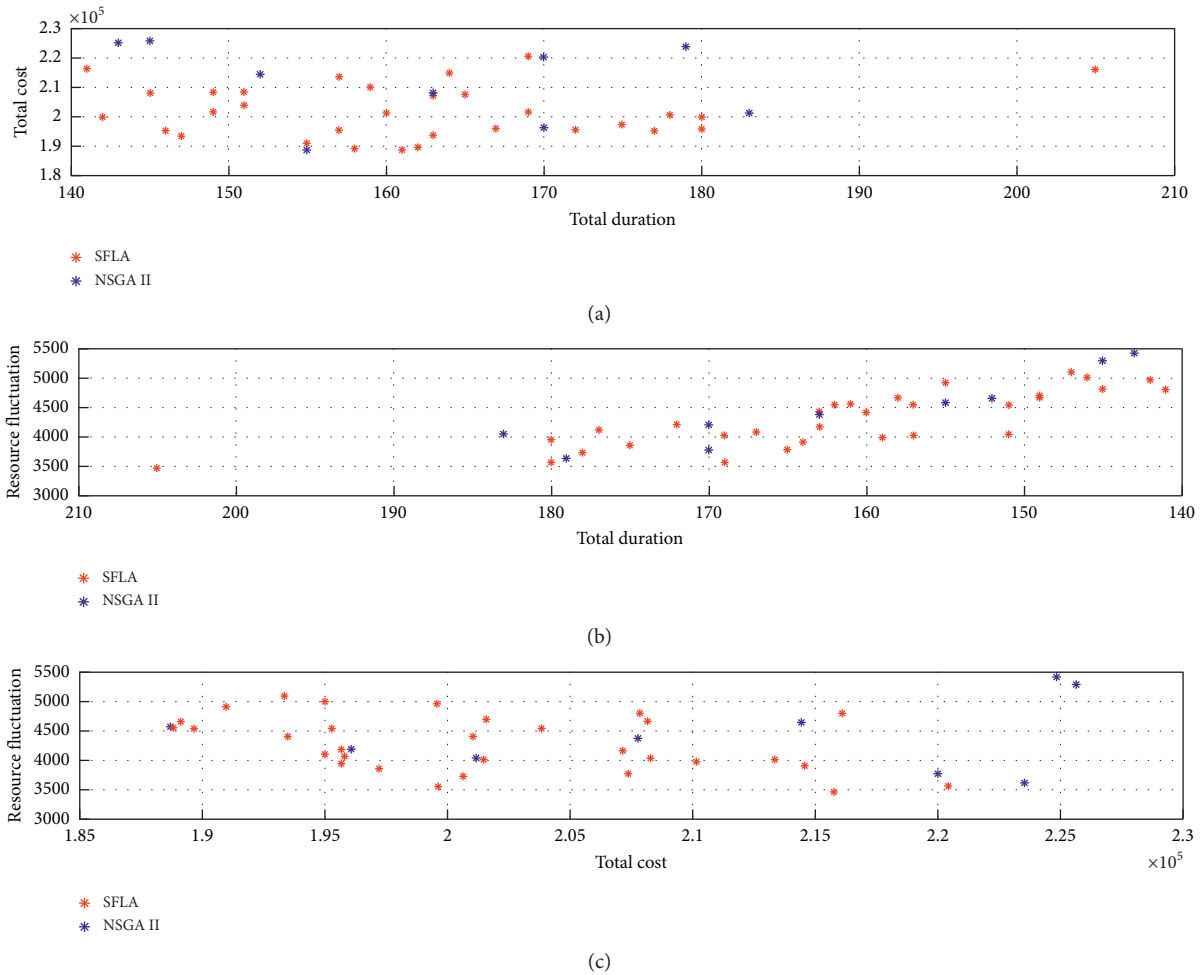


FIGURE 11: Solution comparison in two-dimensional space.

explore including a wider range of constraints such as materials, working areas, and cash flows to better reflect situational project constraints and improve robustness. Future work may also investigate the application of SFLA to real-time energy consumption optimization through the use of integrated wireless smart meters and wireless networks. Furthermore, extending this SFLA approach to household energy management is also important for more efficient energy consumption.

### Data Availability

Data generated or analyzed during the study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This research study was supported by the General Research Fund (GRF) Grant (BRE/PolyU 152099/18E) entitled “Proactive Monitoring of Work-Related MSD Risk Factors

and Fall Risks of Construction Workers Using Wearable Insoles.” The authors also acknowledge funding supports from the People’s Republic of China Ministry of Science and Technology through the National Key R&D Program of China (grant no. 2016YFC0701807) and the Chongqing University through the Special Social Science Research Program of the Central Universities Fundamental Research Foundation (grant no. 2017CDJJK03XK14).

### References

- [1] J. Hong, G. Q. Shen, C. Mao, Z. Li, and K. Li, “Life-cycle energy analysis of prefabricated building components: an input-output-based hybrid model,” *Journal of Cleaner Production*, vol. 112, pp. 2198–2207, 2016.
- [2] C. Peng, “Calculation of a building’s life cycle carbon emissions based on ecotect and building information modeling,” *Journal of Cleaner Production*, vol. 112, pp. 453–465, 2016.
- [3] M. Hajdukiewicz, D. Byrne, M. M. Keane, and J. Goggins, “Real-time monitoring framework to investigate the environmental and structural performance of buildings,” *Building and Environment*, vol. 86, pp. 1–16, 2015.
- [4] D. Heon Jun and K. El-Rayes, “Multiobjective optimization of resource leveling and allocation during construction

- scheduling,” *Journal of Construction Engineering and Management*, vol. 137, no. 12, pp. 1080–1088, 2011.
- [5] K. Kim, J. Walewski, and Y. K. Cho, “Multiobjective construction schedule optimization using modified niched pareto genetic algorithm,” *Journal of Management in Engineering*, vol. 32, no. 2, article 04015038, 2015.
- [6] I. N. Papadaki and A. P. Chassiakos, “Multi-objective construction site layout planning using genetic algorithms,” *Procedia Engineering*, vol. 164, pp. 20–27, 2016.
- [7] X. Ning, J. Qi, C. Wu, and W. Wang, “A tri-objective ant colony optimization based model for planning safe construction site layout,” *Automation in Construction*, vol. 89, pp. 1–12, 2018.
- [8] B. Anvari, P. Angeloudis, and W. Y. Ochieng, “A multi-objective GA-based optimisation for holistic Manufacturing, transportation and Assembly of precast construction,” *Automation in Construction*, vol. 71, pp. 226–241, 2016.
- [9] M. Eusuff, K. Lansey, and F. Pasha, “Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization,” *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.
- [10] B. Ashuri and M. Tavakolan, “Shuffled frog-leaping model for solving time-cost-resource optimization problems in construction project planning,” *Journal of Computing in Civil Engineering*, vol. 29, no. 1, article 04014026, 2013.
- [11] H. Guo, K. Zhu, C. Ding, and L. Li, “Intelligent optimization for project scheduling of the first mining face in coal mining,” *Expert Systems with Applications*, vol. 37, no. 2, pp. 1294–1301, 2010.
- [12] Y. Zhang and S. T. Ng, “An ant colony system based decision support system for construction time-cost optimization,” *Journal of Civil Engineering and Management*, vol. 18, no. 4, pp. 580–589, 2012.
- [13] C. Koo, T. Hong, and S. Kim, “An integrated multi-objective optimization model for solving the construction time-cost trade-off problem,” *Journal of Civil Engineering and Management*, vol. 21, no. 3, pp. 323–333, 2015.
- [14] J. L. Ponz-Tienda, V. Yepes, E. Pellicer, and J. Moreno-Flores, “The resource leveling problem with multiple resources using an adaptive genetic algorithm,” *Automation in Construction*, vol. 29, pp. 161–172, 2013.
- [15] P. Ghoddousi, E. Eshtehardian, S. Jooybanpour, and A. Javanmardi, “Multi-mode resource-constrained discrete time-cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm,” *Automation in Construction*, vol. 30, pp. 216–227, 2013.
- [16] A. Khalili and D. K. Chua, “Integrated prefabrication configuration and component grouping for resource optimization of precast production,” *Journal of Construction Engineering and Management*, vol. 140, no. 2, article 04013052, 2013.
- [17] M. Yahya and M. P. Saka, “Construction site layout planning using multi-objective artificial bee colony algorithm with levy flights,” *Automation in Construction*, vol. 38, pp. 14–29, 2014.
- [18] S. Monghasemi, M. R. Nikoo, M. A. Khaksar Fasaei, and J. Adamowski, “A novel multi criteria decision making model for optimizing time-cost-quality trade-off problems in construction projects,” *Expert Systems with Applications*, vol. 42, no. 6, pp. 3089–3104, 2015.
- [19] M.-F. F. Siu, M. Lu, and S. AbouRizk, “Resource supply-demand matching scheduling approach for construction workforce planning,” *Journal of Construction Engineering and Management*, vol. 142, no. 1, article 04015048, 2015.
- [20] M. Arashpour, R. Wakefield, B. Abbasi, E. Lee, and J. Minas, “Off-site construction optimization: sequencing multiple job classes with time constraints,” *Automation in Construction*, vol. 71, pp. 262–270, 2016.
- [21] M. S. El-Abbasy, A. Elazouni, and T. Zayed, “Generic scheduling optimization model for multiple construction projects,” *Journal of Computing in Civil Engineering*, vol. 31, no. 4, article 04017003, 2017.
- [22] B. Peng, F. L. Flager, and J. Wu, “A method to optimize mobile crane and crew interactions to minimize construction cost and time,” *Automation in Construction*, vol. 95, pp. 10–19, 2018.
- [23] M. Lu and H. Li, “Resource-activity critical-path method for construction planning,” *Journal of Construction Engineering and Management*, vol. 129, no. 4, pp. 412–420, 2003.
- [24] J.-L. Kim and R. D. Ellis Jr., “Comparing schedule generation schemes in resource-constrained project scheduling using elitist genetic algorithm,” *Journal of Construction Engineering and Management*, vol. 136, no. 2, pp. 160–169, 2010.
- [25] A. M. Elazouni and A. A. Gab-Allah, “Finance-based scheduling of construction projects using integer programming,” *Journal of Construction Engineering and Management*, vol. 130, no. 1, pp. 15–24, 2004.
- [26] F. Berthaut, R. Pellerin, N. Perrier, and A. Hajji, “Time-cost trade-offs in resource-constraint project scheduling problems with overlapping modes,” *International Journal of Project Organisation and Management*, vol. 6, no. 3, pp. 215–236, 2014.
- [27] P. G. Ipsilandis, “Multiobjective linear programming model for scheduling linear repetitive projects,” *Journal of Construction Engineering and Management*, vol. 133, no. 6, pp. 417–424, 2007.
- [28] S. Kaiafa and A. P. Chassiakos, “A genetic algorithm for optimal resource-driven project scheduling,” *Procedia Engineering*, vol. 123, pp. 260–267, 2015.
- [29] E. N. Afruzi, A. A. Najafi, E. Roghanian, M. J. C. Mazinani, and O. Research, “A multi-objective imperialist competitive algorithm for solving discrete time,” *cost and quality trade-off problems with mode-identity and resource-constrained situations*, vol. 50, pp. 80–96, 2014.
- [30] M. Shahriari, “Multi-objective optimization of discrete time-cost tradeoff problem in project networks using non-dominated sorting genetic algorithm,” *Journal of Industrial Engineering International*, vol. 12, no. 2, pp. 159–169, 2016.
- [31] J. Zhou, P. E. D. Love, X. Wang, K. L. Teo, and Z. Irani, “A review of methods and algorithms for optimizing construction scheduling,” *Journal of the Operational Research Society*, vol. 64, no. 8, pp. 1091–1105, 2013.
- [32] S. T. Ng and Y. Zhang, “Optimizing construction time and cost using ant colony optimization approach,” *Journal of Construction Engineering and Management*, vol. 134, no. 9, pp. 721–728, 2008.
- [33] S. Lakshminarayanan, A. Gaurav, and C. Arun, “Multiobjective optimization of time-cost-risk using ant colony optimization,” *International Journal of Project Planning and Finance*, vol. 1, no. 1, pp. 22–38, 2010, [http://www.academia.edu/1421329/multi-objective\\_optimization\\_of\\_time-cost-risk\\_using\\_ant\\_colony\\_optimization](http://www.academia.edu/1421329/multi-objective_optimization_of_time-cost-risk_using_ant_colony_optimization).
- [34] R. Shrivastava, S. Singh, and G. Dubey, “Multi-objective optimization of time cost quality quantity using multi colony ant algorithm,” *International Journal of Contemporary Mathematical Sciences*, vol. 7, no. 16, pp. 773–784, 2012, <https://pdfs.semanticscholar.org/3da1/001cbc6cb506679e4ac9aa0b0a15e9013a11.pdf>.

- [35] H. Zhang, H. Li, and C. M. Tam, "Particle swarm optimization for resource-constrained project scheduling," *International Journal of Project Management*, vol. 24, no. 1, pp. 83–92, 2006.
- [36] A. Rahimi-Vahed, A. H. Mirzaei, and I. Engineering, "A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 642–666, 2007.
- [37] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.
- [38] G. G. Samuel and C. C. A. Rajan, "Hybrid: particle swarm optimization–genetic algorithm and particle swarm optimization–shuffled frog leaping algorithm for long-term generator maintenance scheduling," *International Journal of Electrical Power & Energy Systems*, vol. 65, pp. 432–442, 2015.
- [39] J. Luo, X. Li, M.-R. Chen, and H. Liu, "A novel hybrid shuffled frog leaping algorithm for vehicle routing problem with time windows," *Information Sciences*, vol. 316, pp. 266–292, 2015.
- [40] P. Kaur, S. Mehta, and D. Computing, "Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm," *Journal of Parallel and Distributed Computing*, vol. 101, pp. 41–50, 2017.
- [41] S. Hartmann, "Project scheduling with multiple modes: a genetic algorithm," *Annals of Operations Research*, vol. 102, no. 1–4, pp. 111–135, 2001.
- [42] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.





**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

