Hindawi Publishing Corporation
Advances in Civil Engineering
Volume 2015, Article ID 108780, 8 pages
http://dx.doi.org/10.1155/2015/108780

Research Article

Optimizing Construction Project Labor Utilization Using Differential Evolution: A Comparative Study of Mutation Strategies

Nhat-Duc Hoang, Quoc-Lam Nguyen, and Quang-Nhat Pham

Faculty of Civil Engineering, Duy Tan University, P809-K7/25 Quang Trung, Danang 550000, Vietnam

Correspondence should be addressed to Nhat-Duc Hoang; hoangnhatduc@dtu.edu.vn

Received 18 May 2015; Accepted 12 August 2015

Academic Editor: M. C. Deo

In construction management, the task of planning project schedules with consideration of labor utilization is very crucial. However, the commonly used critical path method (CPM) does not inherently take into account this issue. Consequently, the labor utilization of the project schedule derived from the CPM method often has substantial low ebbs and high peaks. This research proposes a model to obtain project schedule with the least fluctuation in labor demand while still satisfying the project deadline and maintain the project cost. The Differential Evolution (DE), a fast and efficient metaheuristic, is employed to search for the most desirable solution of project execution among numerous combinations of activities' crew sizes and start times. Furthermore, seven DE's mutation strategies have also been employed for solving the optimization at hand. Experiment results point out that the *Target-to-Best 1* and a new hybrid mutation strategy can attain the best solution of project schedule with the least fluctuation in labor demand. Accordingly, the proposed framework can be an effective tool to assist decision-makers in the project planning phase.

1. Introduction

Resources in construction projects typically consist of manpower, machinery, materials, money, information, and management decisions [1]. Needless to say, good resource management is essential to ensure that the construction project can be accomplished on schedule and within budget [2–4]. To some extent, it is reasonable to state that the act of construction project management involves nothing but management of resources [5].

In practice, the CPM, as a commonly used tool for project scheduling, has shown to be helpful when the project deadline is unfixed and the resources are also free from restraints. Since the CPM normally does not incorporate a deadline or resource limits, supplementary procedures, such as resource leveling [6, 7] and allocation [7], must be employed separately after the initial schedule is established [8].

In most practical situations, project resources are available in certain quantities and a fluctuated resource profile is proved to be very costly for the contractors [6, 9]. Thus far, the most challenging problem in project planning is to achieve an optimal project execution which results in an appropriate resource utilization with fixed duration. The resource demand should be made as smooth as possible to alleviate short-term peaks or low ebbs [2, 10].

In almost countries, manpower, or labor, is the most crucial resource; thus, planning a schedule which features a smooth labor utilization with low peaks is indeed beneficial for the construction contractors. Needless to say, this leads to reduced costs of temporary facilities for workers and enhancement of on-site safety. Therefore, developing scheduling model that optimizes the project schedule without sacrificing the project completion deadline as well as the project cost is a practical need of construction managers.

Based on the literature review, it is recognizable that employing metaheuristic approaches to solve complex engineering problems has been a major trend in the research community [11–17]. Among metaheuristic approaches, the Differential Evolution (DE) [18] has received an increasing

attention and this algorithm has been applied in a wide span of problem domain [19–22]. The DE employs an efficient way of self-adapting mutation strategies for function optimization over continuous space. The advantages of this method are its simple structure, the ease of implementation, fast convergence, quality of found solution, and robustness [23].

Recently, various research works have been dedicated in harnessing the DE's capability as well as improving its searching efficiency. Brest et al. [24], Zhang and Sanderson [25], Qin et al. [26], and Zheng et al. [20] presented self-adaptive versions of DE in which novel mechanisms of parameter setting are utilized. Hoang [27] introduced a probabilistic similarity-based selection operator that can enhance the DE's selection process. Rahnamayan et al. [28] put forward an opposition-based DE (ODE) which exploits the concept of opposition-based learning for population initialization and generation jumping. Coelho et al. [19], Lu et al. [29], and Cheng and Tran [30] applied the chaotic mapping to improve the DE's searching diversity. Yong et al. [31] proposed a variant of DE with composite trial vector generation strategies and control parameters.

Therefore, this study employed the DE algorithm to tackle the problem of optimizing the labor utilization by means of intelligently scheduling the project's activities. To achieve a more even resource profile, noncritical activities are allowed to shift along available floating times. Moreover, different from previous works in resource leveling [9, 32, 33], the crew sizes of activities are also optimized. Such framework not only is more realistic but also can enhance the flexibility in project scheduling and potentially bring about better solutions.

The rest of the paper is organized as follows. Section 2 provides the research method. The proposed model for optimizing construction project schedule is described in Section 3. Section 4 reports the experimental result and comparison. Some conclusions of the research are stated in Section 5.

2. Research Method

2.1. The Problem of Optimizing Labor Utilization for Construction Project. As mentioned earlier, the CPM normally does not integrate a deadline constraint and resource limits. Furthermore, it does not take into account the efficiency of labor utilization during project execution. Therefore, a process of optimization is often required to adjust the CPM schedule. Herein, the objective is to shift the noncritical activities along their available float times (Figure 1) and select appropriate crew sizes for all activities (Figure 2) so that the labor profile is as smooth as possible. In these two figures, activities A and B have a start-to-start (SS) relationship; activities B and C have a finish-to-start (FS) relationship. The original activity B can be finished in 2 shifts with a crew of 30. Alternatively, a crew of 15 can accomplish the activity B in 4 shifts.

It is noted that altering the crew size of an activity directly accelerates or decelerates its production rate and therefore changes the project duration. Furthermore, the alteration of project schedule must not extent the total project duration and cost. Noticeably, changing the crew size of an activity

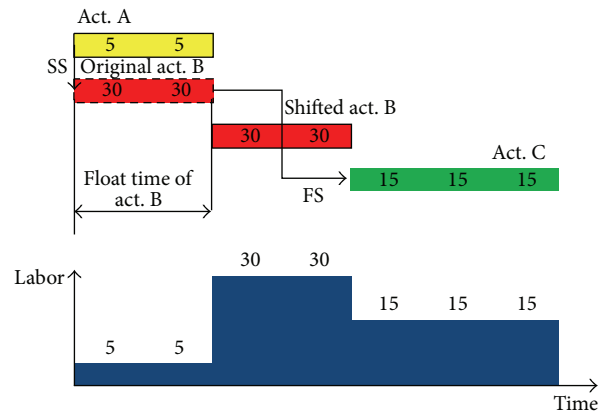

Figure 1: Labor leveling by shifting activity B's start time within its float.

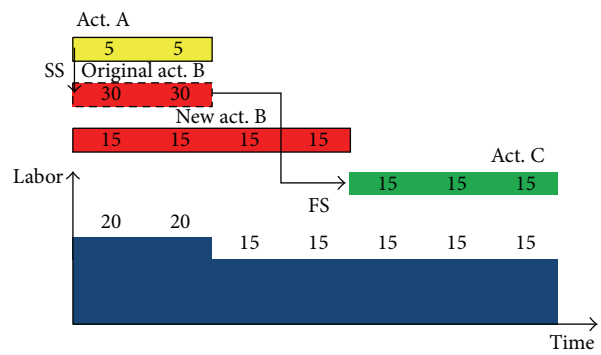

Figure 2: Labor leveling by changing crew size of activity B.

does not increase its direct cost. In addition, since the total project duration is not allowed to prolong, the indirect cost of the project is also maintained. Therefore, it is reasonable to state that the optimization process does not alter the total project cost.

Similar to the resource leveling problem [34], the moment of daily labor demand around the time axis is employed as the objective function:

$$f = \sum_{i=1}^{T} L_i^2, \tag{1}$$

where T represents the project duration; L_i denotes the total labor requirements of all activities performed at time unit i.

The constraints of the optimization problems can be stated as follows:

(1) The total project duration, which is the completion time of the last activity in the network, is fixed.

(2) The precedence constraints between an activity and all the activities in its successor set must be respected.

(3) All the crew sizes of activities are integers within the lower and upper boundaries.

(4) The duration of an activity (measured in shift) is computed as follows:

$$D_i = \frac{W_i}{\mathrm{CS}_i \cdot \mathrm{HR}} \,(\text{shift}), \tag{2}$$

where D_i denotes the duration (shift) of the activity i. W_i is the required working hour of the activity i. CS_i is the crew size of the activity i. HR is the number of working hours in a shift; typically HR = 8 (hour).

(5) All the start times of activities are nonnegative integers within their available float times.

2.2. Differential Evolution (DE). The DE [18] is currently one of the most powerful metaheuristics for solving complex optimization problems. The algorithm generally consists of four phases which are initialization, mutation, crossover, and selection. The whole process is repeated until the termination condition is satisfied. Given the fact that the problem of interest is to minimize a cost function $f(X)$, where the number of decision variables is D, we can describe each phase of DE in details.

2.2.1. Initialization. The DE initiates the optimization process by randomly generating NP number of D-dimensional parameter vectors $X_{i,g}$, where $i = 1, 2, \ldots, \mathrm{NP}$ and g represents the current generation.

2.2.2. Mutation. For each target vector (a vector in the current population), a mutant vector is produced by the following strategies [23, 35]:

DE/Rand/1:

$$V_{i,g+1} = X_{r1,g} + F\left(X_{r2,g} - X_{r3,g}\right), \tag{3}$$

DE/Rand/2:

$$V_{i,g+1} = X_{r1,g} + F\left(X_{r2,g} - X_{r3,g}\right) + F\left(X_{r4,g} - X_{r5,g}\right), \tag{4}$$

DE/Best/1:

$$V_{i,g+1} = X_{\mathrm{best},g} + F\left(X_{r1,g} - X_{r2,g}\right), \tag{5}$$

DE/Best/2:

$$V_{i,g+1} = X_{\mathrm{best},g} + F\left(X_{r2,g} - X_{r3,g}\right) + F\left(X_{r4,g} - X_{r5,g}\right), \tag{6}$$

DE/Target-to-Best/1:

$$V_{i,g+1} = X_{i,g} + F\left(X_{\mathrm{best},g} - X_{i,g}\right) + F\left(X_{r1,g} - X_{r2,g}\right), \tag{7}$$

where $r1$, $r2$, $r3$, $r4$, and $r5$ are random indexes lying between 1 and NP. These randomly chosen integers are also selected to be different from the index i of the target vector. F denotes the mutation scale factor, which controls the amplification of the differential variation. $V_{i,g+1}$ represents the newly created mutant vector.

In addition to the above five strategies, this research proposes investigating two mutation schemes:

DE/Target-to-Best/2:

$$V_{i,g+1} = X_{i,g} + F\left(X_{\mathrm{best},g} - X_{i,g}\right) + F\left(X_{r1,g} - X_{r2,g}\right) + F\left(X_{r3,g} - X_{r4,g}\right), \tag{8}$$

Hybrid DE/Rand/1 and DE/Best/1:

$$V_{i,g+1} = \lambda \cdot X_{\mathrm{best},g} + (1-\lambda) \cdot X_{r1,g} + F\left(X_{r2,g} - X_{r3,g}\right), \tag{9}$$

where $\lambda = 1 - \exp(-g/\delta)$ controls the contribution of the best vector and a randomly chosen vector $X_{r1,g}$. δ is a free parameter. The idea is that as the generation proceeds, the value of λ increases gradually from 0 to 1, and thus the best vector (X_{best}) has more influence over the mutation process. Meanwhile, the effect of randomness is reduced with the hope of accelerating the algorithm convergence.

2.2.3. Crossover. This stage diversifies the current population by exchanging components of target vector and mutant vector. In this stage, a trial vector is created as follows:

$$U_{j,i,g+1} = \begin{cases} V_{j,i,g+1}, & \text{if } \mathrm{rand}_j \le \mathrm{Cr} \text{ or } j = \mathrm{rnb}(i) \\ X_{j,i,g}, & \text{if } \mathrm{rand}_j > \mathrm{Cr} \text{ and } j \ne \mathrm{rnb}(i), \end{cases} \tag{10}$$

where $U_{j,i,g+1}$ is called the trial vector. j denotes the index of element for any vector. rand_j denotes a uniform random number lying between 0 and 1. Cr is the crossover probability. $\mathrm{rnb}(i)$ is a randomly chosen index of $\{1, 2, \ldots, \mathrm{NP}\}$ which guarantees that at least one parameter from the mutant vector ($V_{j,i,g+1}$) is copied to the trial vector ($U_{j,i,g+1}$).

2.2.4. Selection. The trial vector is compared to the target vector. If the trial vector can yield a lower objective function value than its parent, then the trial vector replaces the position of the target vector. The selection operator is expressed as follows:

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{if } f\left(U_{i,g}\right) \le f\left(X_{i,g}\right) \\ X_{i,g} & \text{if } f\left(U_{i,g}\right) > f\left(X_{i,g}\right). \end{cases} \tag{11}$$

3. The Proposed Schedule Optimization Model

This section of the paper describes the proposed construction project schedule optimization model. The model, named as the DE-based Labor utilization Optimization for Construction Project (DeLOCP), aims at intelligently shifting noncritical activities' start time and determining all activities' crew size to achieve the most desirable labor profile for the project. Meanwhile, it must retain the total project cost and duration.

The desirable labor profile is achieved by minimizing the fluctuations of daily labor demand. The model necessitates inputs of project information including precedence relationship, required work hour, and lower and upper boundaries

```
Begin Algorithm
Set Precedence Relationship, Required Work Hour, Lower and Upper Boundaries of Crewsize
// activity information
Set Crewsize // crew information of each activity
Calculate Duration // durations of activities is computed by (2)
For i = 1 : ActNum // Performing forward CPM calculation
        // ActNum = number of activities
        Calculate   ES(i) // Early start of activity i
                    EF(i) // Early finish of activity i
End For
LF(ActNum) = EF(ActNum)
LS(ActNum) = LF(ActNum) − Duration(ActNum)
j = ActNum − 1
While j ≥ 1 // Performing backward CPM calculation
        Calculate   LF(i) // Late finish of activity j
                    LS(i) // Late start of activity j
End While
Return ES, LS, EF, LF
End Algorithm
```

Algorithm 1: CPM scheduling.

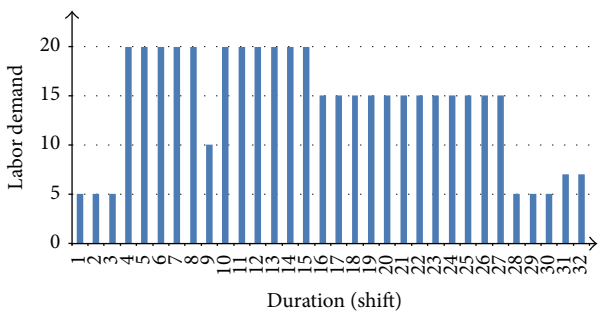

Figure 3: A typical early start labor profile.

Table 1: Project information.

Activity number	Activity relationship	Required working hour (hour)
1		100
2	1FS	220
3	1FS, 2SS	380
4	1FS	210
5	2FS, 3FS	520
6	2FS, 4FS	220
7	4FS, 6SS	370
8	5FS, 7FS	380
9	5FS, 6FS, and 8SS	320
10	7FS	630
11	8FS, 9FS, and 10FS	100

of crew size. With such information, Algorithm 1 (CPM Scheduling) is used to calculate the CPM schedule of the project. In Algorithm 2, the project labor utilization is evaluated by computing the fitness function which is composed of the moment of daily resource demand around the time axis and a penalty term used when the actual project duration is greater than the project deadline. When the parameters of the DE algorithm, including the maximum number of generations ($G_{\max}$), the population size (*PopulationSize*), the mutation scale (F), and the crossover probability (Cr), are set, Algorithm 3 (DeLOCP) can perform the optimization process to obtain the project schedule which features the least fluctuated labor profile.

4. Experimental Result

In this section, the capability of the proposed DeLOCP is illustrated via a construction project which consists of 11 activities (Table 1). Table 1 describes the activities' relationships as well as the required work load, reflected by the required working hours, of all activities. Furthermore, the example of calculating activity durations based on information of the required work load and crew size has been provided in Table 2. The project must be completed in 16 days. Assuming that one day contains 2 shifts, the contractor must accomplish the project within 32 shifts. The *crew size* is allowed to vary between 1 and 20. Figure 3 demonstrates the labor profile obtained from a typical early start schedule with the crew size calculated in Table 2.

Since the decision variables include the start times and crew size of 11 activities, the number of decision variables in the problem at hand is $D = 22$. Based on the recommendation from previous works [18, 19], the population size, the mutation scale, the crossover probability, and maximum generation of the DE are selected as $PopulationSize = 6 \cdot D$, $F \sim \text{Normal}(0.5, 0.15^2)$, $\text{Cr} = 0.8$, and $G_{\max} = 3000$. Moreover,

Table 2: Example of activity duration calculation.

Activity number	Required working hour (hour)	Crew size (person)	Calculated duration (shift)	Rounded duration (shift)
1	100	5	2.5 = 100/(5 ∗ 8)	3 = ceil(2.5)
2	220	5	5.5	6
3	380	10	4.8	5
4	210	5	5.3	6
5	520	5	13.0	13
6	220	5	5.5	6
7	370	10	4.6	5
8	380	10	4.8	5
9	320	5	8.0	8
10	630	10	7.9	8
11	100	7	1.8	2

Note: each shift lasts 8 hours.

Begin Algorithm
Set *ProjectDeadline, WorkingHoursPerShift* // project information
Define *PenaltyFactor* // used when the constraint of project deadline is violated
Define ES, LS, EF, LF of each activity using Algorithm 1
Define Start Time of Activity: $StartTime(i)$ // $\mathrm{ES}(i) \leq StartTime(i) \leq \mathrm{LS}(i)$
Calculate Finish Time of Activity: $FinishTime(i) = StartTime(i) + Duration(i)$
Calculate Daily Labor Demand: $L(i)$
Calculate Moment of L around time axis M_x by (1)
Calculate $DeadlinePenalty = PenaltyFactor \cdot \max(0, ProjectDuration - ProjectDeadline)$;
Calculate $Fitness = M_x + DeadlinePenalty$
Return *Fitness*
End Algorithm

Algorithm 2: Evaluating project labor utilization.

Begin Algorithm
Define *PopulationSize, MuationScaleFactor, CrossoverProbablility*
Initialize Population of vectors
Calculate Fitness of each vector using Algorithm 2
Identify *BestSolution*
For $i = 1 : G_{\max}$ // $G_{\max}$ = maximum number of generations
    For $j = 1 : PopulationSize$
        Perform Mutation using one strategy among (3), (4), (5), (6), (7), (8), and (9)
        Perform Crossover using (10)
        Perform Selection using (11)
        Update *BestSolution*
    End For
End For
Return *BestSolution*
End Algorithm

Algorithm 3: The DeLOCP.

7 mutation strategies (from (3) to (9)) are employed in the mutation operator. When the strategy of Hybrid DE/Rand/1 and DE/Best/1 is used, the free parameter δ is set to be 100 on the basis of experiment.

In the experiment, the DeLOCP with each mutation strategy is run 20 times and the best result, the average result, the standard deviation of the result, and the worst result are reported in Table 3. It is observable that the DeLOCP with the mutation strategy of DE/Target-to-Best/1 and Hybrid DE/Rand/1 *and* DE/Best/1 have produced the best solution: fitness function = 3054 (with average labor demand = 13.8, maximum labor demand = 16.0,

Table 3: Result comparison.

Mutation strategy	Fitness	Average labor demand	Maximum labor demand	Minimum labor demand	Project duration	Generation found best
DE/Rand/1						
Best	3066.0	13.8	16.0	7.0	32.0	59.0
Average	3122.6	13.8	17.1	9.4	32.0	268.7
Standard deviation	68.3	0.1	1.8	0.9	0.0	110.4
Worst	3253.5	14.0	21.0	10.0	32.0	461.0
DE/Rand/2						
Best	3247.0	13.9	17.0	4.0	30.0	37.0
Average	3388.3	14.2	21.1	6.5	31.8	163.7
Standard deviation	69.9	0.3	2.6	1.7	0.5	75.8
Worst	3515.0	14.8	26.0	10.0	32.0	315.0
DE/Best/1						
Best	3085.0	13.8	16.0	5.0	32.0	25.0
Average	3166.1	13.8	18.3	8.2	32.0	74.7
Standard deviation	78.8	0.1	1.7	1.8	0.0	53.4
Worst	3347.0	14.0	21.0	12.0	32.0	223.0
DE/Best/2						
Best	3055.5	13.7	15.0	7.0	32.0	50.0
Average	3118.5	13.8	17.0	10.2	32.0	133.2
Standard deviation	51.3	0.1	1.1	1.5	0.0	49.3
Worst	3257.0	14.1	19.0	13.0	32.0	223.0
DE/Target-to-Best/1						
Best	3054.0	13.7	15.0	7.0	32.0	69.0
Average	3092.7	13.8	17.0	9.8	32.0	148.0
Standard deviation	29.1	0.1	1.5	1.8	0.0	72.7
Worst	3168.0	13.9	20.0	13.0	32.0	344.0
DE/Target-to-Best/2						
Best	3126.0	13.8	15.0	4.0	31.0	31.0
Average	3245.5	14.0	18.4	7.5	32.0	123.1
Standard deviation	57.9	0.1	1.9	2.0	0.2	50.2
Worst	3357.0	14.3	23.0	11.0	32.0	217.0
Hybrid DE/Rand/1 and DE/Best/1						
Best	3054.0	13.7	15.0	8.0	32.0	69.0
Average	3099.0	13.8	17.4	10.4	32.0	146.4
Standard deviation	27.4	0.0	1.8	1.0	0.0	73.7
Worst	3151.0	13.9	23.0	12.0	32.0	272.0

minimum labor demand = 12, and project duration = 32.0 (shift)). The optimized crew sizes and start times of all activities are [13, 7, 4, 9, 6, 4, 6, 7, 8, 8, 13] and [1, 2, 2, 6, 14, 9, 9, 25, 27, 17, 32], respectively. The optimized daily labor demand is illustrated in Figure 4.

Thus, considering the best found solution, the DE/Target-to-Best/1 and Hybrid DE/Rand/1 *and* DE/Best/1 have both found the best solution (3054), followed by the DE/Best/2 (3055.5), DE/Rand/1 (3066.0), DE/Best/1 (3085.0), DE/Target-to-Best/2 (3126.0), and DE/Rand/2 (3247.0). Moreover, in terms of average fitness function, the strategy DE/Target-to-Best/1 has produced most desirable outcome (3092.6); the DeLOCP with the strategy of Hybrid DE/Rand/1 *and* DE/Best/1 has yielded the second best result (3099.0). On the other hand, when considering the standard deviation of the result and the worst result, the mutation strategy of Hybrid DE/Rand/1 *and* DE/Best/1 shows better performance than that of the DE/Target-to-Best/1.

Moreover, the convergence property of each mutation strategy can be judged by analyzing the "Generation found best." The "Generation found best" denotes the number of generations where the best solution was found by the DeLOCP. It can be seen that the strategy DE/Best/1 tends to converge very fast. On average, the algorithm only needs 74.7

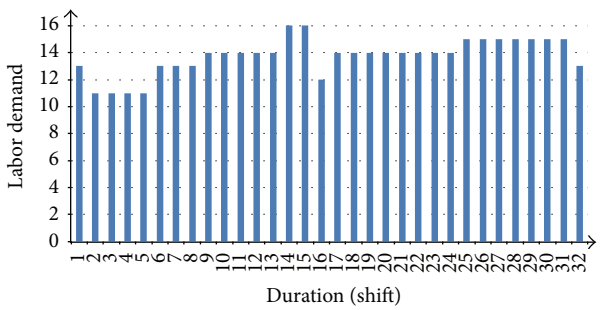

Figure 4: The labor profile optimized by the DeLOCP.

generations to converge. On the other hand, the convergence of the DeLOCP which uses the strategy DE/Rand/1 is the slowest (average "Generation found best" = 268.7 generation). Nevertheless, both of the above strategies seem to get stuck in some local optimal.

Furthermore, the two strategies (the Hybrid DE/Rand/1 *and* DE/Best/1 and the DE/Target-to-Best/1) converge faster than the DE/Rand/1 and slower than the DE/Best/1. Interestingly, the average "Generation found best" of Hybrid DE/Rand/1 *and* DE/Best/1 (146.4) and the DE/Target-to-Best/1 (148) is almost equivalent. Based on that, it can be stated that these two mutation schemes possess almost the same convergence property. Thus, compared to other mutation strategies, the Hybrid DE/Rand/1 *and* DE/Best/1 and the DE/Target-to-Best/1 manifest better compromise between the convergence property and the quality of solution.

5. Conclusion

This research proposes a model, named DeLOCP, for optimizing construction project schedule with consideration of labor utilization. The DeLOCP, based on the DE algorithm, intelligently shifts noncritical activities' start times and determines activities' crew sizes to attain the most desirable labor profile. Therefore, the approach does not alter the total project cost and duration. Experimental result shows that the proposed method has successfully optimized the project schedule which features a smooth labor profile with insignificant peaks and ebbs. This study also investigates 7 mutation strategies of the DE algorithm. The result comparison has demonstrated that the DE/Target-to-Best/1 and the newly proposed Hybrid DE/Rand/1 *and* DE/Best/1 have attained the best optimization performance. Future developments of the current research includes applying the proposed method for solving large scale construction projects and investigating the potentiality of hybridization of metaheuristic methods for tackling the problem at hand.

Conflict of Interests

The authors Nhat-Duc Hoang, Quoc-Lam Nguyen, and Quang-Nhat Pham declare that there is no conflict of interests regarding the publication of this paper.

References

[1] D. W. Halpin and R. W. Woodhead, *Construction Management*, Wiley, Hoboken, NJ, USA, 1998.

[2] Y. Tang, R. Liu, and Q. Sun, "Two-stage scheduling model for resource leveling of linear projects," *Journal of Construction Engineering and Management*, vol. 140, no. 7, Article ID 04014022, 2014.

[3] N. Glišović, "Comparison of a fuzzy genetic and simulated annealing algorithm approach for project time-cost tradeoff," *Journal of Applied Mathematics*, vol. 2014, Article ID 817921, 12 pages, 2014.

[4] J.-L. Kim and J. R. D. Ellis, "Permutation-based elitist genetic algorithm for optimization of large-sized resource-constrained project scheduling," *Journal of Construction Engineering and Management*, vol. 134, no. 11, pp. 904–913, 2008.

[5] M. Park, "Model-based dynamic resource management for construction projects," *Automation in Construction*, vol. 14, no. 5, pp. 585–598, 2005.

[6] 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.

[7] T. Hegazy, "Optimization of resource allocation and leveling using genetic algorithms," *Journal of Construction Engineering and Management*, vol. 125, no. 3, pp. 167–175, 1999.

[8] T. Hegazy and W. Menesi, "Heuristic method for satisfying both deadlines and resource constraints," *Journal of Construction Engineering and Management*, vol. 138, no. 6, pp. 688–696, 2012.

[9] H.-H. Tran and N.-D. Hoang, "A novel resource-leveling approach for construction project based on differential evolution," *Journal of Construction Engineering*, vol. 2014, Article ID 648938, 7 pages, 2014.

[10] B. Afshar-Nadjafi, Z. Khalaj, and E. Mehdizadeh, "A branch and bound approach to solve the preemptive resource leveling problem," *International Journal of Manufacturing Engineering*, vol. 2013, Article ID 930920, 7 pages, 2013.

[11] T. W. Liao, P. J. Egbelu, B. R. Sarker, and S. S. Leu, "Metaheuristics for project and construction management—a state-of-the-art review," *Automation in Construction*, vol. 20, no. 5, pp. 491–505, 2011.

[12] A. Majumdar, B. Nanda, D. K. Maiti, and D. Maity, "Structural damage detection based on modal parameters using continuous ant colony optimization," *Advances in Civil Engineering*, vol. 2014, Article ID 174185, 14 pages, 2014.

[13] J.-S. Chou, N.-T. Ngo, and A.-D. Pham, "Shear strength prediction in reinforced concrete deep beams using nature-inspired metaheuristic support vector regression," *Journal of Computing in Civil Engineering*, 2015.

[14] O. Adekanmbi and P. Green, "Conceptual comparison of population based metaheuristics for engineering problems," *The Scientific World Journal*, vol. 2015, Article ID 936106, 9 pages, 2015.

[15] M.-Y. Cheng and N.-D. Hoang, "Risk score inference for bridge maintenance project using evolutionary fuzzy least squares support vector machine," *Journal of Computing in Civil Engineering*, vol. 28, no. 3, Article ID 04014003, 2014.

[16] Y. Qu and A. Deng, "The optimisation for local coupled extreme learning machine using differential evolution," *Mathematical Problems in Engineering*, vol. 2015, Article ID 946292, 9 pages, 2015.

[17] A. Pham, N. Hoang, and Q. Nguyen, "Predicting compressive strength of high-performance concrete using metaheuristic-optimized least squares support vector regression," *Journal of Computing in Civil Engineering*, 2015.

[18] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[19] L. D. S. Coelho, H. V. H. Ayala, and V. C. Mariani, "A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization," *Applied Mathematics and Computation*, vol. 234, pp. 452–459, 2014.

[20] F. Zheng, A. C. Zecchin, and A. R. Simpson, "Self-adaptive differential evolution algorithm applied to water distribution system optimization," *Journal of Computing in Civil Engineering*, vol. 27, no. 2, pp. 148–158, 2013.

[21] E. R. F. A. Schneider and R. A. Krohling, "A hybrid approach using TOPSIS, Differential Evolution, and Tabu Search to find multiple solutions of constrained non-linear integer optimization problems," *Knowledge-Based Systems*, vol. 62, pp. 47–56, 2014.

[22] M.-Y. Cheng and D.-H. Tran, "Two-phase differential evolution for the multiobjective optimization of time-cost tradeoffs in resource-constrained construction projects," *IEEE Transactions on Engineering Management*, vol. 61, no. 3, pp. 450–461, 2014.

[23] P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 315–346, 2013.

[24] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[25] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[26] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[27] N.-D. Hoang, "NIDE: a novel improved differential evolution for construction project crashing optimization," *Journal of Construction Engineering*, vol. 2014, Article ID 136397, 7 pages, 2014.

[28] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 2010–2017, IEEE, Vancouver, Canada, July 2006.

[29] Y. Lu, J. Zhoun, H. Qin, Y. Wang, and Y. Zhang, "Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 378–387, 2011.

[30] M. Cheng and D. Tran, "Integrating chaotic initialized opposition multiple-objective differential evolution and stochastic simulation to optimize ready-mixed concrete truck dispatch schedule," *Journal of Management in Engineering*, 2015.

[31] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[32] S. E. Christodoulou, G. Ellinas, and A. Michaelidou-Kamenou, "Minimum moment method for resource leveling using entropy maximization," *Journal of Construction Engineering and Management*, vol. 136, no. 5, pp. 518–527, 2010.

[33] K. El-Rayes and D. H. Jun, "Optimizing resource leveling in construction projects," *Journal of Construction Engineering and Management*, vol. 135, no. 11, pp. 1172–1180, 2009.

[34] J. Martinez and P. Ioannou, "Resource leveling based on the modified minimum moment heuristic," in *Proceedings of the Conference of Computing in Civil and Building Engineering*, pp. 287–294, American Society of Civil Engineers, Anaheim, Calif, USA, June 1993.

[35] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer, 2005.