

Research Article

A Novel Resource-Leveling Approach for Construction Project Based on Differential Evolution

Hong-Hai Tran and Nhat-Duc Hoang

*Department of Technology and Construction Management, Faculty of Building and Industrial Construction,
National University of Civil Engineering, Room 307, A1 Building, No. 55 Giai Phong Road, Hanoi 10000, Vietnam*

Correspondence should be addressed to Nhat-Duc Hoang; ducxd85@yahoo.com

Received 30 November 2013; Revised 12 March 2014; Accepted 5 April 2014; Published 21 May 2014

Academic Editor: Manoj Jha

Copyright © 2014 H.-H. Tran and N.-D. Hoang. 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.

In construction engineering, project schedules are commonly established by the critical path method. Nevertheless, these schedules often lead to substantial fluctuations in the resource profile that are not only impractical but also costly for the contractors to execute. Therefore, in order to smooth out the resource profile, construction managers need to perform resource-leveling procedures. This paper proposes a novel approach for resource leveling, named as resource leveling based on differential evolution (RLDE). The performance of the RLDE is compared to that of Microsoft Project software, the genetic algorithm, and the particle swarm optimization algorithm. Experiments have proved that the newly developed method can deliver the most desirable resource-leveling result. Thus, the RLDE is an effective method and it can be a useful tool for assisting managers/planners in the field of project management.

1. Introduction

In today's market condition, the survivability of any construction contractor essentially depends on its capability of managing resources. Ineffective resource management escalates the operational expense or even gives rise to financial and scheduling problems [1, 2]. Without doubt, the excess requirement of resources in the construction site leads to the extension of project duration. As the construction contractor fails to deliver the project by the prespecified date, the owner may suffer from financial losses due to the nonavailability of the facility [3]. In addition, construction delays often bring about legal disputes among parties, higher overhead costs, and degraded reputation, and they occasionally result in project failures [4, 5]. As a consequence, resource management is an essential task that needs to be implemented thoroughly in the planning phase.

Basically, resources in construction projects consist of manpower, equipment, materials, money, and expertise. Needless to say, the proper management of these resources holds the key to the successful accomplishment of any project [3]. However, construction schedules, generated by network scheduling techniques, are often characterized by undesirable resource fluctuations that are impractical and costly for

the contractors to implement [6]. The reason is that it is expensive to hire and to lay off workers on a short-term basis according to the fluctuations in the resource profile. Additionally, if the resources cannot be managed efficiently, they may exceed the supply capability of the contractor and lead to schedule delay. Finally, the contractor must maintain a number of idle resources during low-demand periods. These facts undoubtedly cause profit damage for construction companies. Thus, construction managers mandatorily need to perform schedule-adjusting processes to alleviate fluctuations in the resource profile during the project execution.

The process of smoothing out resources is well known as the resource-leveling problem [6–8]. In resource leveling, the objective is to minimize the peak demand and fluctuations in the pattern of the resource usage [9]. This process aims to minimize variations in the resource profile by shifting noncritical activities within their available floats and keep the project duration unchanged.

Initially, the mathematical approaches were used to solve the resource-leveling problems because they can provide optimal solutions to the problem at hand. Nonetheless, these methods become impractical when the size of project network reaches a considerably large number. This is because

resource leveling belongs to the class of combinatorial problem. Hence, the increasing number of decision variables causes the problem solving to become infeasible. As a consequence, mathematical methods are not computationally tractable for real-life projects [10].

Other researchers attempted to utilize heuristic methods in solving the resource-leveling problem [9, 11]. Despite the simplicity of resource-leveling heuristics and their wide implementation on commercial project management software (e.g., Microsoft Project), the result oftentimes cannot satisfy project managers. It is because the heuristic approaches operate on the basis of prespecified rules. Thus, their performances are dependent on specific types of problem and on the set of implemented rules. Hence, they can only deliver feasible solutions and by no means guarantee an optimum solution [12].

Due to the limitations of the mathematic and heuristic methods, the application of evolutionary algorithms (EAs) for resource leveling has attracted more attention in recent years [13, 14]. EAs, inspired by the natural evolution of species, have been successfully utilized to tackle optimization problems in diverse fields [15]. Evolutionary computation is characterized by iterative progresses used to guide the randomly initiated population to the final optimal solution.

Recently, differential evolution (DE) [16–18] has drawn increasing interest of researchers who have explored its capability in a wide range of problems. DE is a population-based stochastic search engine, which is efficient and effective for global optimization in the continuous domain [19]. It uses mutation, crossover, and selection operators at each generation to move its population toward the global optimum. Superior performance of DE over other algorithms has been verified in many reported research works [15, 17, 20, 21]. Nevertheless, none of the previous works has investigated the potentiality of DE in solving resource-leveling problem.

Thus, this paper proposes a novel approach for resource leveling, named as resource leveling based on differential evolution (RLDE). RLDE is developed based on the DE algorithm as the search engine. The rest of the paper is organized as follows. In Section 2, the research methodology is presented. The structure of the proposed RLDE is shown in Section 3. The performance of the newly developed model is demonstrated in Section 4. Finally, some conclusions are mentioned in the last section of this paper.

2. Research Methodology

2.1. The Resource-Leveling Problem. In the resource-leveling problem, the objective is to reduce the peak resource demand and smooth out day-to-day resource consumptions within the required project duration and with the assumption of unlimited resource availability. Hence, the resource leveling can be expressed as an optimization problem within which the following cost function is minimized [9, 12]:

$$f = \sum_{i=1}^T (y_i - y_u)^2, \quad (1)$$

where T denotes the project duration, y_i represents the total resource requirements of the activities performed at time unit i , and y_u represents a uniform resource level given as follows:

$$y_u = \frac{\sum_{i=1}^T y_i}{T}. \quad (2)$$

Equation (1) means that if the value of the objective function f is small, the actual resource profile of the project will be close to the form of the ideal resource profile. In the ideal resource profile, the resource demand is constant throughout the project duration. Moreover, (1) can be rewritten in the following form:

$$f = \sum_{i=1}^T y_i^2 - 2y_u \sum_{i=1}^T y_i + y_u^2. \quad (3)$$

Because activity duration and rate of resource for each activity are fixed, y_u and $\sum_{i=1}^T y_i$ are constant. Thus, the cost function can be expressed as follows:

$$f = \sum_{i=1}^T y_i^2. \quad (4)$$

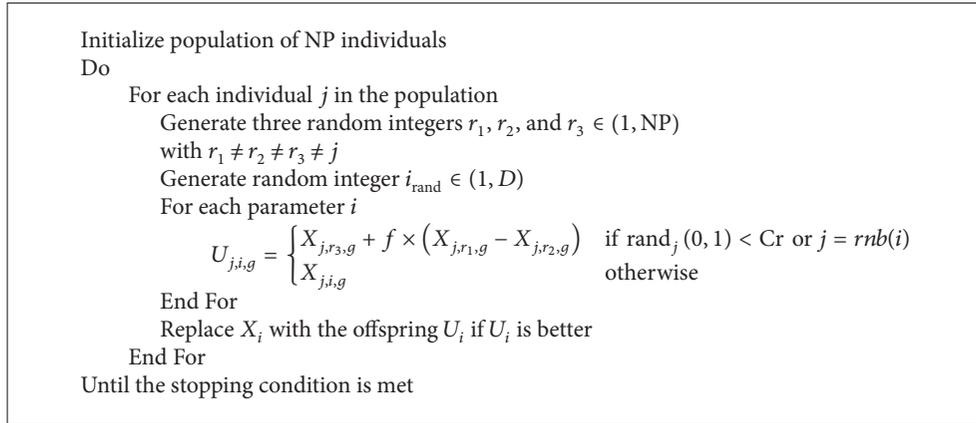
2.2. Differential Evolution. Differential evolution (DE) is an evolutionary algorithm proposed by Storn and Price [16, 17], which is designed for real parameter optimization problems. The algorithm of DE is depicted in Algorithm 1. In this figure, it is noted that NP represents the size of the population, $X_{j,i}$ is the j th decision variable of the i th individual in the population, g is the current generation, and D denotes the number of decision variables. $\text{rand}_j(0, 1)$ is a uniform random number lying between 0 and 1, and $\text{rnb}_{(i)}$ is a randomly chosen index of $(1, 2, \dots, \text{NP})$.

DE is based on the utilization of a novel crossover-mutation operator, based on the linear combination of three different individuals and one subject-to-replacement parent (or target vector) [20]. The crossover-mutation operator yields a trial vector (or child vector) which will compete with its parent in the selection operator. These two terms, trial vector and child vector, can be used interchangeably. The selection process is performed via selection between the parent and the corresponding offspring [22].

In the selection process, the trial vector is compared to the target vector (or the parent) using the greedy criterion [17]. These two terms, target vector and parent vector, can be used synonymously. Under the greedy criterion, the fitness value is the only information that determines which vector is allowed to survive in the next generation (see Figure 2). Thus, if the trial vector can yield a lower objective function value than its parent, then the trial vector supersedes the target vector. The selection operator is expressed as follows:

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{if } f(U_{i,g}) \leq f(X_{i,g}) \\ X_{i,g} & \text{if } f(U_{i,g}) > f(X_{i,g}), \end{cases} \quad (5)$$

where $X_{i,g}$ represents the parent vector at generation g . $U_{i,g}$ denotes the trial vector at generation g . $X_{i,g+1}$ is the chosen individual which survives to the next generation ($g + 1$).



ALGORITHM 1: The differential evolution algorithm.

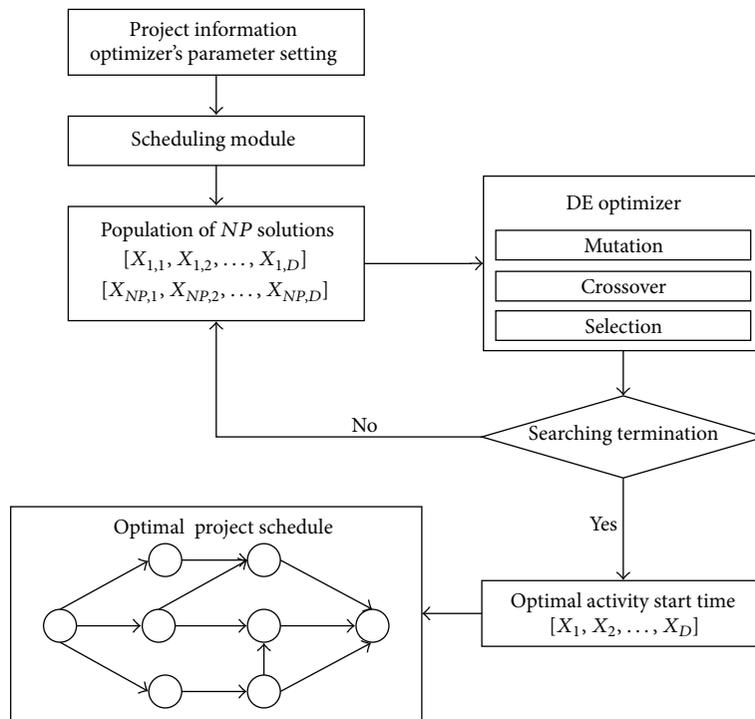


FIGURE 1: The resource leveling based on differential evolution (RLDE).

The optimization process can terminate when the stopping criterion is met. The user can set the type of this stopping condition. Commonly, the maximum generation (G_{max}) or the maximum number of function evaluations (NFE) can be used as the stopping condition. When the optimization process terminates, the final optimal solution will be readily presented to the user.

3. Resource Leveling Based on Differential Evolution (RLDE)

This section of the paper is dedicated to describing the RLDE resource-leveling approach (see Figure 1). It is noticed that the RLDE is developed based on DE as the optimization

technique. The objective of this optimization model is to minimize daily fluctuations in the project resource utilization. This goal is achieved by altering the schedules of noncritical activities in the project within their available floats. Thus, the schedules of critical activities are unchanged and the total project duration remains the same as the critical path method (CPM) based schedule.

Herein, the resource leveling is accomplished by minimizing the fluctuations of resource requirements and a desirable uniform resource level. The model requires inputs of project information including the activity relationship, the activity duration, and the resource demand. In addition, the user also needs to provide the parameter setting for the search engine, such as the maximum number of searching

TABLE 1: Metrics for performance measurement.

Performance measurement	Notation	Calculation
Fitness function	f	$f = \sum_{k=1}^T (y_k)^2$
Maximum resource demand	RD_{\max}	$RD_{\max} = y_{\max}$
Cumulative variation of the resource demand between consecutive periods	CRV	$CRV = \sum_{k=1}^{T-1} y_{k+1} - y_k $
Maximum variation of the resource demand between two consecutive periods	RV_{\max}	$RV_{\max} = \max_{k=1}^{T-1} \{ y_{k+1} - y_k \}$

Note: T is the total project duration. y_k represents the resource requirements at day k th. y_{\max} denotes the peak of the resource demand throughout the project execution.

generation (G_{\max}), the population size (NP), the mutation scale (F), and the crossover probability (Cr). With these inputs, the scheduling module can carry out the calculation process to obtain the CPM schedule, which specifies the early start and the late start of each activity. If all the necessary information is provided, the model is capable of operating automatically without users' interventions.

Before the searching process can commence, an initial population of feasible solutions is created by a uniform random generator. A solution for the resource-leveling problem is represented as a vector with D elements as follows:

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

where D is the number of decision variables of the problem at hand. It is obvious that D is also the number of noncritical activities in the project network. The index i denotes the i th individual in the population. The vector X represents the start time of D activities in the network. Since the original DE operates with real-value variables, a function is employed to convert those activities' start times from real values to integer values within the feasible domain. Consider

$$X_{i,j} = \text{Round}(\text{LB}(j) + \text{rand}[0, 1] \times (\text{UB}(j) - \text{LB}(j))), \quad (7)$$

where $X_{i,j}$ is the start time of activity j at the individual i th. $\text{rand}[0, 1]$ denotes a uniformly distributed random number between 0 and 1. $\text{LB}(j)$ and $\text{UB}(j)$ are early start and late start of the activity j .

The search engine DE takes into account the result obtained from the scheduling module and it shifts noncritical activities within their float values to seek for an optimal project schedule. In our research, the following objective function and constraints are employed:

$$f = \sum_{k=1}^T (y_k)^2 \quad (8)$$

subject to

$$ST_i - ES_i \leq TF_i; \quad ST_i \geq 0; \quad i = 1, 2, \dots, D, \quad (9)$$

where T denotes the project duration; y_k represents the total resource requirements of the activities performed at time unit k . ST_i is the start time of activity i . ES_i and TF_i represent early

start and total float of activity i , respectively. D is the number of activities in the network.

After the searching process terminates, an optimal solution is identified. The project schedule and its corresponding resource histogram are then constructed based on the optimal start times of all activities. The user can assess the quality of a project schedule using a set of metrics (see Table 1). It is worth noticing that the fitness function stated in (8) is the objective function for the search engine DE. Meanwhile, the set of metrics shown in Table 1 includes the fitness function (see (8)) and other criteria (the maximum resource demand, the cumulative variation of the resource demand between consecutive periods, and the maximum variation of the resource demand between two consecutive periods).

It is noted that the fitness function utilized in our research aims at obtaining an optimized resource profile that is as close to the ideal resource profile as possible. Thus, this fitness function is the most crucial criterion and it is very similar to that of the minimum moment method used in [9, 12]. However, the three additional criteria, illustrated in Table 1, can provide more insights about the optimized resource profile for users.

4. Experimental Result

In this section, a construction project adapted from Sears et al. [23] is used to demonstrate the capability of the newly developed RLDE. The project consists of 44 activities (see Table 2). In this experiment, R denotes a resource used in the project. The resource profile of project before resource-leveling process is shown in Figure 2.

RLDE is applied to reduce the resource fluctuation in this project. Parameter setting for the DE optimizer is shown in Table 3. The project's resource profile after being optimized by the RLDE is depicted in Figure 3. The optimal solutions, which specify the optimal start times of activities, are listed in Table 4. The optimal results obtained from the new model are listed as follows: $f = 13839$, $RD_{\max} = 19$, $RV_{\max} = 7$, and $CRV = 57$.

The results comparison between the RLDE and Microsoft Project 2007 (MP 07) is shown in Table 5. The fitness function of the proposed method (13839) is better than that of MP 07 (15245). This means that the resource profile produced by the RLDE is more stable than that obtained from the benchmark approach. Noticeably, the proposed RLDE can reduce the

TABLE 2: Project information.

Activity	Duration	Predecessor	Resource demand (R)
1	1		0
2	10	1	5
3	5	1	2
4	15	1	3
5	3	1	2
6	10	1	2
7	15	2	6
8	7	3	10
9	3	5	6
10	3	5	2
11	2	5	2
12	3	4, 10, 11	6
13	2	10	1
14	2	8, 12	5
15	3	12, 13	2
16	1	14	6
17	1	15	7
18	1	16	7
19	4	7, 9, 17, 18	13
20	2	15, 18	9
21	2	19	4
22	1	20	6
23	3	21	8
24	1	22	3
25	4	23, 24	8
26	2	25	7
27	25	6	5
28	3	23	6
29	3	23	2
30	3	26	9
31	3	30	10
32	3	30	3
33	2	27, 29, 30	4
34	1	32	0
35	4	33	1
36	3	34, 35	12
37	3	36	12
38	3	28, 31, 37	3
39	5	28, 31, 37	8
40	1	36	2
41	3	38, 39, 40	10
42	1	41	3
43	6	42	3
44	1	43	0

peak of resource demand much better than the commercial software. The maximum resource variation between two consecutive days of the RLDE is 7 which is lower than that of MP 07 ($RV_{max} = 12$). Moreover, in terms of the cumulative variation of the resource demand between two consecutive

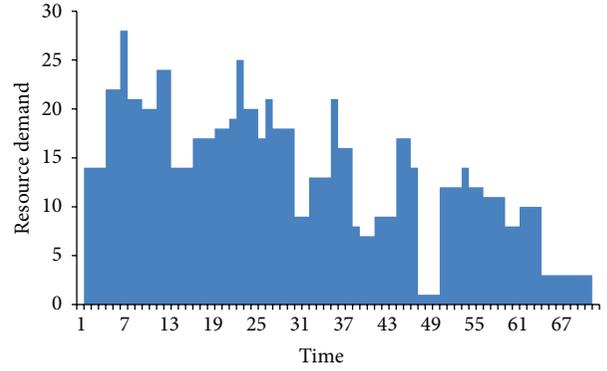


FIGURE 2: The original resource profile.

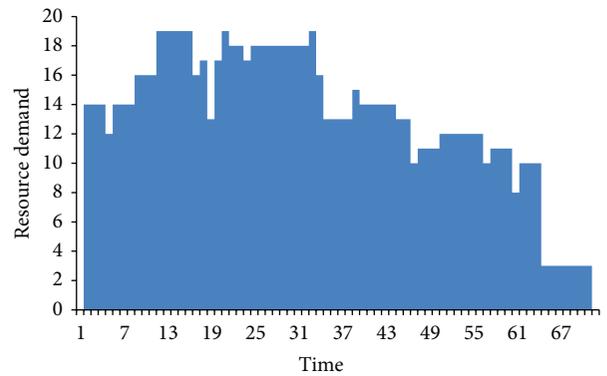


FIGURE 3: The optimal resource profile produced by the RLDE.

TABLE 3: The RLDE's parameter setting.

Parameters	Notation	Setting
Population size	NP	$8 \times D$
Maximum generation	G_{max}	1000
Mutation scale	F	0.5
Crossover probability	Cr	0.8

Note: D is the number of decision variables.

days, the proposed approach is also significantly better than MP 07.

To better verify the performance of the proposed RLDE, its performance is compared to that of the genetic algorithm (GA) [24] and the particle swarm optimization algorithm (PSO) [25]. The GA and PSO algorithms are both extensively utilized in the field of optimization and they have been applied to solve the resource-leveling problem. It is noted that, in this experiment, the population size and the maximum number of generations of the two benchmark algorithms are set similar to that of the RLDE. Thus, comparing the performance of the RLDE with that of the GA and PSO algorithms can point out the advantage of the proposed technique for tackling the problem at hand.

As can be seen from Table 5, the proposed resource-leveling approach has outperformed the GA and PSO algorithms. The fitness functions of the RLDE, GA, and PSO are

TABLE 4: The optimal solutions of the RLDE method.

Activity	Start time
1	1
2	2
3	2
4	2
5	2
6	2
7	12
8	12
9	9
10	6
11	7
12	19
13	18
14	22
15	21
16	24
17	26
18	25
19	27
20	31
21	31
22	33
23	33
24	34
25	36
26	40
27	20
28	45
29	39
30	42
31	48
32	45
33	45
34	50
35	47
36	51
37	54
38	58
39	57
40	57
41	62
42	65
43	66
44	72

13830, 13945, and 13965, respectively. This outcome indicates that the proposed method can produce a resource profile that is less fluctuating than those yielded by the GA and PSO algorithms. In addition, the newly proposed approach also yields the best performances in terms of the peak of

TABLE 5: Result comparison.

Approach	Performance			
	f	y_{\max}	RV_{\max}	CRV
Early start schedule	15339	28	13	151
Microsoft Project 2007	15245	24	12	139
PSO	13965	20	7	72
GA	13945	21	7	95
RLDE	13839	19	7	57

the resource demand and the cumulative variation of the resource demand between two consecutive days.

5. Conclusion

This study investigates the capability of the DE algorithm in solving resource-leveling problem and proposes a new resource-leveling approach, named as RLDE, to tackle the complexity of the problem at hand. The experiments and the result comparisons have demonstrated that the newly proposed RLDE can overcome the commonly used Microsoft Project software as well as the GA and PSO algorithms. This fact has strongly proved that the RLDE is a promising tool to assist project managers in dealing with the resource-leveling problem. The future works of this research include evaluating the performance of the new approach with different forms of fitness function reported in the literature and utilizing novel techniques to enhance the searching capability of the DE algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] F. A. Karaa and A. Y. Nasr, "Resource management in construction," *Journal of Construction Engineering and Management*, vol. 112, no. 3, pp. 346–357, 1986.
- [2] J. Wu and Q. An, "New approaches for resource allocation via dea models," *International Journal of Information Technology and Decision Making*, vol. 11, no. 1, pp. 103–117, 2012.
- [3] G. Maged, "Evolutionary resource scheduler for linear projects," *Automation in Construction*, vol. 17, no. 5, pp. 573–583, 2008.
- [4] S. A. Assaf and S. Al-Hejji, "Causes of delay in large construction projects," *International Journal of Project Management*, vol. 24, no. 4, pp. 349–357, 2006.
- [5] D. Arditi and T. Pattanakitchamroon, "Selecting a delay analysis method in resolving construction claims," *International Journal of Project Management*, vol. 24, no. 2, pp. 145–155, 2006.
- [6] 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.
- [7] 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.

- [8] S. H. H. Doulabi, A. Seifi, and S. Y. Shariat, "Efficient hybrid genetic algorithm for resource leveling via activity splitting," *Journal of Construction Engineering and Management*, vol. 137, no. 2, pp. 137–146, 2011.
- [9] J. Son and M. J. Skibniewski, "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach," *Journal of Construction Engineering and Management*, vol. 125, no. 1, pp. 23–31, 1999.
- [10] Y. Liu, S. Zhao, X. Du, and S. Li, "Optimization of resource allocation in construction using genetic algorithms," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '05)*, vol. 2, pp. 3428–3432, August 2005.
- [11] R. B. Harris, "Packing method for resource leveling (pack)," *Journal of Construction Engineering and Management*, vol. 116, no. 2, pp. 331–350, 1990.
- [12] 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.
- [13] J. Geng, L. Weng, and S. Liu, "An improved ant colony optimization algorithm for nonlinear resource-leveling problems," *Computers and Mathematics with Applications*, vol. 61, no. 8, pp. 2300–2305, 2011.
- [14] S. Leu, C. Yang, and J. Huang, "Resource leveling in construction by genetic algorithm-based optimization and its decision support system application," *Automation in Construction*, vol. 10, no. 1, pp. 27–41, 2000.
- [15] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [16] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution a Practical Approach to Global Optimization*, Springer, 2005.
- [17] 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.
- [18] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.
- [19] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On convergence of differential evolution over a class of continuous functions with unique global optimum," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 42, no. 1, pp. 107–124, 2012.
- [20] R. L. Becerra and C. A. C. Coello, "Cultured differential evolution for constrained optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 33–36, pp. 4303–4322, 2006.
- [21] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.
- [22] E. Mezura-Montes, C. A. C. Coello, E. I. Tun-Morales, S. D. Computación, and V. Tabasco, "Simple feasibility rules and differential evolution for constrained optimization," in *Proceedings of the 3rd Mexican International Conference on Artificial Intelligence (MICAI '04)*, April 2004.
- [23] K. Sears, G. Sears, and R. Clough, *Construction Project Management: A Practical Guide to Field Construction Management*, John Wiley and Son, Hoboken, NJ, USA, 5th edition, 2008.
- [24] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithm*, John Wiley and Son, Hoboken, NJ, USA, 2004.
- [25] M. Clerc, *Particle Swarm Optimization*, ISTE, 2006.

