JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT
ISSN 1392-3730 / eISSN 1822-3605
2017 Volume 23(1): 113–124
https://doi.org/10.3846/13923730.2014.982699

FUZZY CLUSTERING CHAOTIC-BASED DIFFERENTIAL EVOLUTION FOR RESOURCE LEVELING IN CONSTRUCTION PROJECTS

Min-Yuan CHENG[a], Duc-Hoc TRAN[a, b], Nhat-Duc HOANG[c]

[a]*Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan*
[b]*Faculty of Project Management, University of Science and Technology – The University of DaNang, #54, Nguyen Luong Bang Rd., DaNang, Vietnam*
[c]*Institute of Research and Development, Duy Tan University, Vietnam, P809 - K7/25 Quang Trung, Danang, Vietnam*

Received 07 Sep 2013; accepted 23 Jul 2014

Abstract. Project scheduling is an important part of project planning in many management companies. Resource leveling problem describes the process of reducing the fluctuations in resource usage over the project duration. The goal of resource leveling is to minimize the incremental demands that cause fluctuations of resources, and thus avoid undesirable cyclic hiring and firing during project execution. In this research, a novel optimization model, named as Fuzzy Clustering Chaotic-based Differential Evolution for solving Resource leveling (FCDE-RL), is introduced. Fuzzy Clustering Chaotic-based Differential Evolution (FCDE) is developed by integrating original Differential Evolution with fuzzy c-means clustering and chaotic techniques to tackle complex optimization problems. Chaotic was exploited to prevent the optimization algorithm from premature convergence. Meanwhile, fuzzy c-means clustering acts as several multi-parent crossover operators to utilize the information of the population efficiently to enhance the convergence. Experimental results revealed that the new optimization model is a promising alternative to assist project managers in dealing with construction project resource leveling.

Keywords: resource levelling, fuzzy clustering, chaotic, differential evolution, construction management.

Introduction

In today's market condition, the survivability of a construction company essentially depends on its capability of managing resources (Karaa, Nasr 1986). Poor resource management may unnecessarily escalate the operational expense or even gives rise to financial and scheduling problems. The excessive requirement of resource in the construction site may lead to the extension of project duration. As the contractor cannot accomplish the project by the pre-specified date, the owner may suffer from financial loss due to the non-availability of the facility (Georgy 2008). Moreover, construction delays often bring about disputes among parties, higher overhead costs, degradation of reputation, and occasionally result in project failure (Arditi, Pattanakitchamroon 2006; Assaf, Al-Hejji 2006). Hence, resource management is a crucial task that needs to be implemented thoroughly in the planning phase.

Construction resources basically consist of manpower, equipment, materials, money, and expertise; efficient management of these resources holds the key to the successful execution of any project (El-Rayes, Jun 2009). However, construction schedules, generated by network scheduling techniques, often bring about undesirable resource fluctuations that are impractical, inefficient, and costly for the contractors to implement (Martinez, Ioannou 1993). Thus, construction managers mandatorily need to perform schedule-adjusting process to reduce unnecessary fluctuations in resource utilization during the project execution.

Needless to say, the fluctuations of resource are troublesome for the contractor (Christodoulou *et al.* 2010). 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 the time of low demand. These facts undoubtedly cause profit decrease for construction companies.

The process of smoothing out resources is well known as resource leveling and has been studied extensively by many researchers (Savin *et al.* 1996; Son, Skibniewski 1999; Doulabi *et al.* 2011). In resource

Corresponding author: Duc-Hoc Tran
E-mail: *duchoc87@gmail.com*

leveling, the objective is to minimize the peak demand and fluctuations in the pattern of resource usage (Yan *et al.* 2005). This process aims to minimize variation in resource profile by shifting noncritical activities within their available floats and keep the project duration unchanged. Resource leveling of construction project can be solved by a variety of methods, ranging from mathematical methods, heuristics to evolutionary approaches (e.g. Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, etc.).

Initially, the mathematical approaches were used to solve the resource leveling problems because they can provide optimal solutions to the problem at hand. However, 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 (Savin *et al.* 1996). Consequently, mathematical methods are not computationally tractable for real-life projects (Yan *et al.* 2005).

Other researchers attempted to utilize heuristic methods in solving the resource leveling problem (Harris 1990; Son, Skibniewski 1999). 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 pre-specified rules. Thus, their performance is dependent on specific types of problem and on which rules are implemented. Hence, they can only deliver good feasible solutions and by no means guarantee an optimum solution (Hegazy 1999).

Due to the limitations of mathematic and heuristic methods, the application of Evolutionary Algorithms (EAs) for resource leveling has attracted more attention in recent years (Leu *et al.* 2000; Geng *et al.* 2011). EAs are stochastic optimization techniques based on the principles of natural evolution, have been successfully utilized to tackle optimization problems in diverse fields (Das, Suganthan 2011). Evolutionary computation is characterized by iterative progresses used to guide the randomly initiated population to the final optimal solution. Currently, evolutionary optimization algorithms, such as Genetic Algorithm (Haupt, R. L., Haupt, S. E. 2004), Particle Swarm Optimization (Clerc 2006), Ant Colony Optimization (Yin, Wang 2006), and Differential Evolution (Price *et al.* 2005; Foekstistov 2006), remain an active research area in the scientific community. Nevertheless, these algorithms still suffer from certain weakness. Geng *et al.* (2011) point out that premature convergence and poor exploitation are the main obstacles for the EAs in coping with complex optimization problems. Thus, it is a requisite to develop a more efficient algorithm to attain satisfactory solutions for resource leveling problem in practical construction projects.

Recently, Differential Evolution (DE) (Storn, Price 1997; Price *et al.* 2005) has increasingly drawn interest of researchers who have explored the capability of this algorithm 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. 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 (Storn, Price 1997; Becerra, Coello 2006; Zhang, Sanderson 2009).

Despite of aforementioned advantages, original DE or many of its variants still have to face some drawbacks. DE does not guarantee the convergence to the global optimum. It is easily trapped into local optima resulting in a low optimizing precision or even failure (Jia *et al.* 2011). In DE, population may not be distributed over search space, and individuals may be trapped in local solution. It may require more generations to converge toward optimal or near-optimal solution (Bedri Ozer 2010). DE has been shown to have certain weaknesses, especially if the global optimum should be located using a limited number of fitness function evaluations. It is good at exploring the search space and locating the region of global minimum but slow at exploitation of the solution (Noman, Iba 2008).

The inherent characteristics of chaotic systems provide an efficient approach for maintaining the population diversity in search algorithms. Chaos is apparently an irregular motion, seemingly unpredictable random behaviour exhibited by a deterministic nonlinear system under deterministic conditions. Chaotic systems are sensitive to small differences in initial condition may produce huge changes in outcomes. It is extremely sensitive to the initial conditions, and its property sometimes referred to as the instability in the so-called butterfly effect or Liapunove's sense (Kim, Stringer 1992). Some studies focus on hybridizing DE with chaotic algorithm, Jia *et al.* (2011) utilizes a chaotic local search (CLS) with a 'shrinking' strategy. The CLS helps to improve the optimizing performance of the canonical DE by exploring a huge search space in the early run phase to avoid premature convergence, and exploiting a small region in the later run phase to refine the final solutions. Bedri Ozer (2010) embeds seven chaotic maps to create the initial population of DE algorithm. It has been detected that coupling emergent results in different areas, like those of DE and complex dynamics, can improve the quality of results in some optimization problems.

Fuzzy c-means clustering is the process of dividing a set of objects into groups or clusters of similarities thereby speeding up the optimization search in DE. A successful clustering is able to reliably find true natural groupings in the data set. A soft clustering approach, fuzzy c-means clustering, is introduced to DE to help track the evolution of search algorithm by introducing cluster centers to the populations. In fuzzy clustering, data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Kwedlo (2011) proposed a new version of DE which uses k-means clustering to fine-tune each candidate solution obtained by mutation and crossover operators of DE. Wang *et al.* (2007) utilizes clustering technique to improve solution accuracy with less computational effort. Experiments showed that the new method is able to find near optimal solutions efficiently.

Hybridization with other different algorithms is an interesting direction for the improvement of DE (Cai *et al.* 2011). Although, there are many proposal for the improvement of DE, only a little work studied the hybridization of clustering and chaotic with the DE method (Cai *et al.* 2011). To the best of our knowledge, until now the fuzzy c-means clustering and chaotic is not used to enhance the performance of DE.

Therefore, the objective of this research is to utilize fuzzy c-means clustering and chaotic techniques to cope with the difficulties in original DE. Chaotic sequences have been adopted instead of random sequences and very interesting and good results have been exploited to prevent the new approach from premature convergence. Meanwhile, fuzzy c-means clustering acts as several multi-parent crossover operators to utilize the information of the population efficiently to make the algorithms converge faster. The remainder of this paper is organized as follows: Section 1 reviews briefly literature relevant to the establishment of the new optimization model. Sections 2 and 3 provide a detailed description of the proposed optimization model for resource leveling problem. Section 4 uses a numerical experiment to demonstrate model performance. The final section presents conclusions and suggests directions for future work.

1. Literature review

1.1. Resource leveling problem

In construction management, resource scheduling problems are being investigated intensively because of their practical importance. Resource leveling remains one of the top challenges due to its complexity (Hegazy 1999; Leu *et al.* 2000; Pang *et al.* 2008; Geng *et al.* 2011). In the resource leveling problem, the objective is to reduce the peak resource demand and smooth out day-to-day consumption within the required project duration, and with the assumption of unlimited resource availability. Thus, the resource leveling can be formulized as an optimization problem within which the following cost function is minimized (Hegazy 1999; Son, Skibniewski 1999; Ponz-Tienda *et al.* 2013):

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

where c_j, $0 \le j \le m$ is the weighting score of the k^{th} resource, 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 by:

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

According to Son and Skibniewski (1999), Eqn (1) can be rewritten as follows:

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

Since 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:

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

The Eqn (4) in essence, is equivalent to the minimum moment of resource histogram around the time axis as mentioned in previous works (Harris 1990; Hegazy 1999). Moreover, the objective function of the resource-leveling problem needs some modification. It is because the optimization process may yield several scheduling solutions or, in other words, resource profiles that have the same minimum moment of resource demand (Son, Skibniewski 1999). Although the cost function values are identical, their resource fluctuations can be different. Hence, to identify the most preferable resource profile, the deviations between resource consumption in consecutive time periods (Easa 1989) and the peak of resource demand should be taken into account (Son, Skibniewski 1999). In this research, a modified objective function for resource-leveling optimization model is presented later on.

1.2. Differential evolution

Differential evolution (DE) is a simple population-based, direct-search for solving global optimization problems (Storn, Price 1997; Price *et al.* 2005). The original DE algorithm is described briefly as follows.

Let $S \subset \Re^n$ be the search space of the problem under consideration. Then, DE utilizes NP, D-dimensional parameter vectors: $X_{i,G} = \{x_{i,G}^1, x_{i,G}^2, ..., x_{i,G}^D\}$, $i = 1, 2, ..., NP$ as a population for each generation of the algorithm. The initial population is generated randomly and should cover the entire parameter space. At each generation, DE applies two operators, namely mutation and crossover (recombination) to yield one trial vector $U_{i,G+1}$ for each target vector $X_{i,G}$. Then, a selection phase takes place to determine whether the trial vector enters the population of the next generation or not. For each target vector $X_{i,G}$, a mutant vector $V_{i,G+1}$ is determined by the following equation:

$$V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}), \quad (5)$$

where, $r_1, r_2, r_3 \in \{1, 2, ..., NP\}$ are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$, and F is a scaling factor such that $F \in [0,1]$.

Following the mutation phase, the crossover operator is applied to increase the diversity. For each mutant vector $V_{i,G+1}$, a trial vector $U_{i,G+1} = \{u^1_{i,G+1}, u^2_{i,G+1}, ..., u^D_{i,G+1}\}$ is generated, using the following scheme:

$$u^j_{i,G+1} = \begin{cases} v^j_{i,G+1} & if\,(rand_j[0,1) \leq CR \; or \; j = j_{rand} \\ x^j_{i,G} & \text{otherwise} \end{cases} \quad (6)$$

$$j = 1, 2, ..., D.$$

$CR \subset [0,1]$ is user-defined crossover constant; j_{rand} is a randomly chosen index from $\{1, 2, ..., D\}$, which can ensure that trail vector $U_{i,G+1}$ will differ from its target $X_{i,G}$ by at least one parameter.

To decide whether the trial vector $U_{i,G+1}$ should be a member of the population in next generation, it is compared to the corresponding target vector $X_{i,G}$ using the greedy criterion. The selection operator is expressed as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) < f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases}. \quad (7)$$

With the memberships of the next generation are selected, the evolutionary cycle of the DE iterates until a stopping condition is satisfied.

1.3. Chaos sequences

Chaos theory is a scientific theory that describes erratic behaviour in certain nonlinear dynamic systems. Chaotic mappings may be considered as particles traveling within a limited range in a deterministic nonlinear dynamic system with no definite regularity associated with their path of travel. Although movement is randomized, it is extremely sensitive to the initial condition (Cheng *et al.* 2012). Because chaotic sequences are easy and fast to generate and store, there is no need for storage over long sequences (Bedri Ozer 2010).

The one dimensional logistic map is one of the simplest systems with density of periodic orbits:

$$X_{n+1} = \mu X_n (1 - X_n). \quad (8)$$

In equation above, X_n is the n^{th} chaotic number where n denotes the iteration number. Obviously, X_n under conditions that initial $X_0 \in (0,1)$ and that $X_0 \notin \{0.0, 0.25, 0.5, 0.75, 1.0\}$. The variation of control parameter μ in Eqn (8) will directly impact the behaviour of X greatly. The domain area of control parameter μ has often been defined as $[0,4]$. In the experiments $\mu = 4$ has been used (Jiang 1998).

1.4. Fuzzy c-means clustering

Clustering is a process that aims at decomposing a given set of objects into subgroups or clusters based on similarity. The aim is to divide the data-set in such way that objects belonging to the same cluster are similar as possible, whereas objects belonging to different clusters are as dissimilar as possible. Clustering algorithms can be divided into main categories: crisp (or hard) clustering procedures where each data is assigned to exactly one cluster, fuzzy clustering techniques where every data point belongs to every cluster with a specific algorithm degree of membership (Jain *et al.* 1999). Many clustering algorithms are introduced in the literature. Fuzzy clustering presented the advantage of dealing efficiently with overlapping clusters. It delivered better and stable results compared with other clustering techniques (Alami *et al.* 2007). The fuzzy c-means (FCM) clustering (Bezdek *et al.* 1984) is employed in this study.

2. Fuzzy c-means clustering chaotic-based differential evolution (FCDE)

In this section, the newly proposed FCDE optimization algorithm is described in details. The FCDE is the core optimizer in the FCDE-RL model. It is noticed that our algorithm is developed based on standard Differential Evolution (Storn, Price 1997; Price *et al.* 2005) by integrating original DE with fuzzy c-means clustering and chaotic techniques. Chaos approach effectively exploits the whole search space and provides the necessary diversity in the DE population. Consequently, this operation incurs additional time and iteration to search the global optimum. On the contrary, fuzzy c-means clustering technique enhances the convergence speed of the algorithm by introducing the cluster centers. These moving centers provide direction for search of the global optimum improving the overall efficiency of the search algorithm. FCDE model, exploits the inherent characteristics of both chaos algorithm and fuzzy clustering and integrates it with differential evolution to improve the overall search capabilities of DE in finding the optimal solutions for a given search space. The overall picture of the proposed algorithm is illustrated in Figure 1.

2.1. Initialization

FCDE commences the search process by randomly generating population size NP, Maximum of generation $G_{\max}$ number of D-dimensional parameter vectors $X_{i,\text{G}}$ where $i = 1, 2, ..., NP$ and g indicates the current generation. In the original DE algorithm, NP does not change during the optimization process (Price *et al.* 2005). Moreover, the initial population (at $G = 0$) is expected to cover the entire search space uniformly. Hence, we can simply generate these individuals as follows:

$$X_{i,0} = LB + rand[0,1] * (UB - LB), \quad (9)$$

where $X_{i,0}$ is the decision variable i at the first generation. $rand[0,1]$ denotes a uniformly distributed random number between 0 and 1. LB and UB are two vectors of lower bound and upper bound for any decision variable.

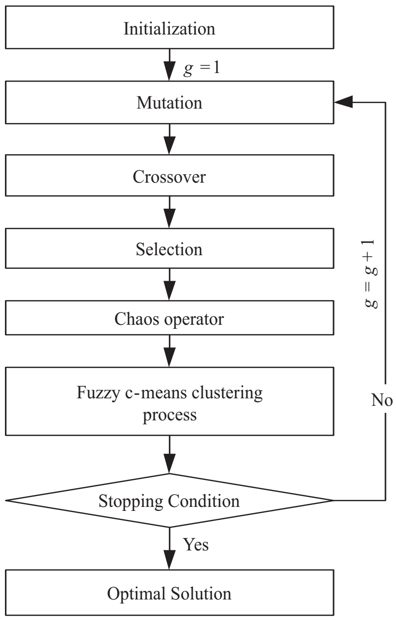

Fig. 1. Fuzzy clustering chaotic based differential evolution

2.2. Mutation

Once initialized, DE mutates the population to produce a set of mutant vectors. A mutated vector $V_{i,G+1}$ is generated corresponding to the target vector $X_{i,G}$ according to Eqn (5).

2.3. Crossover

The crossover operation is to diversity the current population by exchanging components of target vector and mutant vector. In this stage, a new vector, named as trial vector, is created according to Eqn (6).

2.4. Selection

In this stage, the trial vector is compared to the target vector (or the parent) using the greedy criterion (Price *et al.* 2005). 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; otherwise, the target vector retains its place in the population for at least one more generation. The selection operator is expressed as Eqn (7).

2.5. Chaotic differential evolution

The logistic map that generates chaotic sequences in DE, named CDE which ensures the individual in population to be spread in the search space as much as possible for population diversity used in experiments. Incorporating chaotic map into DE is proven to enhance the global convergence by escaping the suboptimal solution. Figure 2 shows the main steps of generating chaotically population.

In Figure 2, g and $G_{\max}$ is current and maximum generation, respectively. If the probability condition is satisfied, a percentage of the population (CF) is selected to do chaos. Then, mapping this population to chaotic

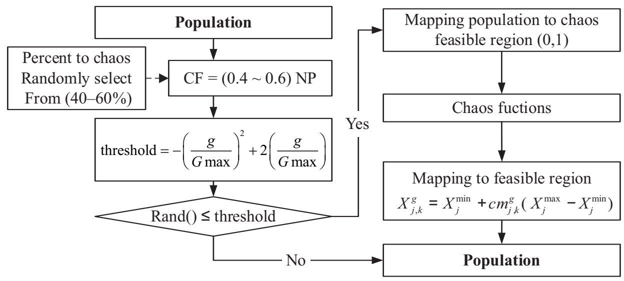

Fig. 2. Chaotic approach

feasible region in range (0,1) according to chaotic conditions and performs the logistic map following Eqn (8) to yield chaotic values $cm_{j,k}^g$. Afterwards, map the chaotic values to feasible region according to equation as follows:

$$X_{j,k}^g = X_j^{\min} + cm_{j,k}^g (X_j^{\max} - X_j^{\min}), \tag{10}$$

where $j = 1, 2, ..., D$; $k = 1, 2, ..., CF$, $X_{j,k}^g$ is the j^{th} decision variable of k^{th} individual in CF-population at generation g; $X_j^{\max}$, $X_j^{\min}$ is upper and lower bound of j^{th} decision variable, respectively.

2.6. Fuzzy clustering differential evolution

The FCM clustering technique adopted in DE, named FDE, could easily conduct an efficient convergence of DE. FCM introduced in this study was intended to track the main stream of population movement during DE evolution. Each cluster centers could be treated approximately as one of the items in the main stream of evolution, and replaced for population as candidate individuals. The FDE algorithm is illustrated in Figure 3. Where m is clustering period, NP is the population size, and k, the number of centroid (Cai *et al.* 2011), is an integer number from $[2, \sqrt{NP}]$.

In order to exploit the search space efficiently, the clustering is performed periodically in the FDE. It is similar to the method used in Weiguo *et al.* (2005) and Cai *et al.* (2011). The reason for performing the clustering periodically is that DE needs time to explore the search place and form clusters. An attempt to perform the clustering very early will lead to a false identifica-

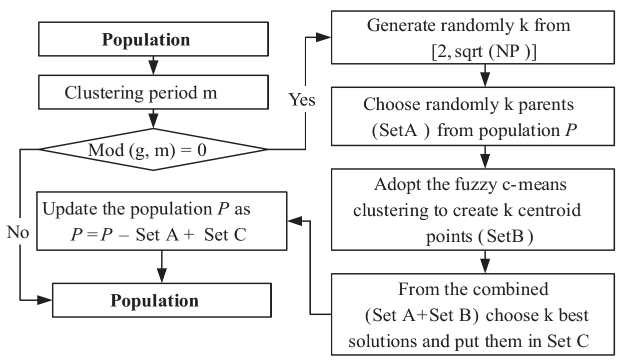

Fig. 3. Fuzzy c-means clustering algorithm

tion of clusters. Consequently, it is important to choose a clustering period that is large enough so that DE has time to completely form stable clusters. In this approach a parameter m is adopted to control the clustering period.

Initially, the period of the clustering operator specified in the algorithm is 10. When the clustering condition is satisfied, the fuzzy c-means clustering will create *k* offspring and the population needs to be updated by them. Deb (2005) proposed a generic population-based algorithm-generator for real-parameter optimization, where the optimization task is divided into four independent plans: (i) selection plan; (ii) generation plan; (iii) replacement plan; and (iv) update plan. In the flowchart of the algorithm above can also be described with the population update-algorithm proposed in Deb (2005).

- Selection plan: Choose *k* individuals from current population randomly (the set A);
- Generation plan: Create *k* offspring (the set B) using the fuzzy c-means clustering;
- Replacement plan: Choose *k* best solutions (the set C) from combined (set A + set B) for replacement;
- Update plan: Update the population as P = P – Set A + Set C.

In the update plan, the k best solutions are chosen from the combined set A + Set B, thereby the elite-preservation is ensured.

2.7. Stopping condition

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

3. Fuzzy c-means clustering chaotic-based differential evolution for resource leveling

This section describes the FCDE-RL optimization model (see Fig. 4). It is noticed that the FCDE-RL is developed based on the FCDE as the searching engine. The objective of this optimization model is to minimize daily fluctuations in resource utilization without altering the total project duration.

In this study, we consider the case that the resource leveling is accomplished by minimizing the fluctuations between resource requirements and a desirable uniform resource level. The model requires inputs of project information including activity relationship, activity duration and resource demand. In addition, the user also needs to provide parameter setting for the optimizer, such as maximum number of searching generation (G_{max}), the population size (*NP*), the chaotic percentage (*CF*), and the period clustering (*m*). With these inputs, the scheduling module can carry out calculation process to obtain critical path method (CPM) based schedule, early start and late start of each activity. With all the necessary information provided, the model is capable of operating automatically.

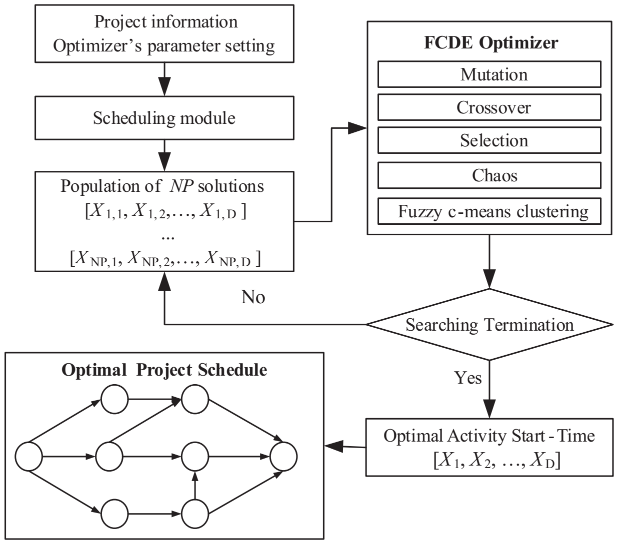

Fig. 4. Flowchart of FCDE for RL problem (FCDE-RL)

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

$$X = [X_{i,1}, X_{i,2}, ..., X_{i,j}, ..., X_{i,D}], \qquad (11)$$

where *D* is the number of decision variable of the problem at hand. It is obvious that *D* is also the number of 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 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:

$$X_{i,j} = Round(LB(j) + rand[0,1] \times (UB(j) - LB(j))), \qquad (12)$$

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

The optimizer (FCDE) takes into account the result obtained from scheduling module and shifts noncritical activities within their float times to seek for an optimal project schedule. In the research, the following objective function and constraints are employed:

$$f = \sum_{j=1}^{m} c_j \left(\alpha \sum_{k=1}^{T} (y_k)^2 + \beta \sum_{k=1}^{T-1} \left| y_{k+1} - y_k \right| + \gamma \times y_{max} \right), \qquad (13)$$

Subject to:

$$ST_i - ES_i \le FF_i;\ ST_i \ge 0;\ i = 1, 2, ..., D, \qquad (14)$$

where *T* denotes the project duration; y_k epresents the total resource requirements of the activities performed at time unit *k*. $(y_{k+1} - y_k)$ measures the different of resource usage between two consecutive time periods. y_{max} denotes the peak of resource demand throughout the project execution. α, β, and γ are weighting coefficients. ST_i is the start time of activity *i*. ES_i and FF_i represent early start and free 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 activities start time. The user can assess the quality of a project schedule using a set of metrics (Table 1).

Table 1. Metrics for performance measurement

Notation	Calculation
Fitness	$Fitness = \alpha \frac{1}{2}\sum_{k=1}^{T}(y_k)^2 + \beta\sum_{k=1}^{T-1}\lvert y_{k+1} - y_k\rvert + \gamma \times y_{\max}$ where α, β, and γ are weighting coefficient
M_x	$M_x = \frac{1}{2}\sum_{k=1}^{T}(y_k)^2$
$RD_{\max}$	$RD_{\max} = y_{\max}$
CRV	$CRV = \sum_{k=1}^{T-1}\lvert y_{k+1} - y_k\rvert$
$RV_{\max}$	$RV_{\max} = \max[\lvert y_2 - y_1\rvert, \lvert y_3 - y_2\rvert, ..., \lvert y_T - y_{T-1}\rvert]$ where *T* is the total project duration

4. Experimental results

In this section, a construction project adapted from Sears *et al.* (2008) is used to demonstrate the capability of the newly developed FCDE-RL model. The project consists of 44 activities and the total project duration is determined to be 70 days (see Table 2). In this study, the resource of interest is manpower. The resource profile of project before resource-leveling process is shown in the Figure 5.

Table 2. Project information

Act. ID	Dur.	Predecessors	Daily resource demand	Early start (ES)	Late start (LS)
1	*2*	*3*	*4*	*5*	*6*
1	0	–	0	0	0
2	10	1	5	0	0
3	5	1	2	0	9
4	15	1	3	0	3
5	3	1	2	0	12
6	10	1	2	0	8
7	15	2	6	10	10
8	7	3	10	5	14

End of Table 2

1	*2*	*3*	*4*	*5*	*6*
9	3	5	6	3	22
10	3	5	2	3	15
11	2	5	2	3	16
12	3	4, 10, 11	6	15	18
13	2	10	1	6	19
14	2	8, 12	5	18	21
15	3	12, 13	2	18	21
16	1	14	6	20	23
17	1	15	7	21	24
18	1	16	7	21	24
19	4	7, 9, 17, 18	13	25	25
20	2	15, 18	9	22	30
21	2	19	4	29	29
22	1	20	6	24	32
23	3	21	8	31	31
24	1	22	3	25	33
25	4	23, 24	8	34	34
26	2	25	7	38	38
27	25	6	10	10	18
28	3	23	6	34	52
29	3	23	2	34	40
30	3	26	9	40	40
31	3	30	10	43	52
32	3	30	3	43	46
33	2	27, 29, 30	4	43	43
34	0	32	0	46	49
35	4	33	1	45	45
36	3	34, 35	12	49	49
37	3	36	12	52	52
38	3	28, 31, 37	3	55	57
39	5	28, 31, 37	8	55	55
40	1	36	2	52	59
41	3	38, 39, 40	10	60	60
42	1	41	3	63	63
43	6	42	3	64	64
44	0	43	0	70	70

Table 3. FCDE-RL's parameter setting

Input parameters	Notation	Setting
Number of decision variables	*D*	44
Population size	*NP*	8x*D*
The crossover probability	*CR*	0.9
Percentage of population to chaos	*CF*	40–60%
Period clustering	*m*	10
Number of centroid in clustering	*k*	$[2, \sqrt{NP}]$
Maximum generation	$G_{\max}$	1000
Weighting coefficient 1	α	1
Weighting coefficient 2	β	1
Weighting coefficient 3	γ	10

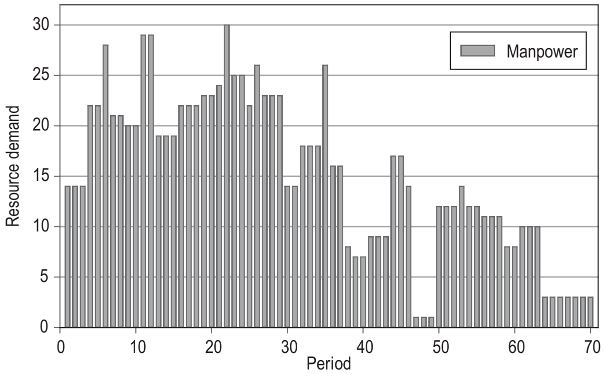

Fig. 5. Project resource profile before resource-leveling process

4.1. Optimization result of FCDE-RL

In this section, FCDE-RL model is applied to reduce significant resource fluctuations. Based on proposed values from the literature and several experiments (Storn, Price 1997; Cai *et al.* 2011), we set parameters for FCDE optimizer as shown in the Table 3. The project's resource profile after being optimized by FCDE-RL is depicted in the Figure 6. The optimal solution or optimal activities' start times are listed in the Table 4. The optimal results obtained from the new model are listed as followed:

Fitness = 9522, M_x = 9215, $RD_{\max}$ = 24, $RV_{\max}$ = 7, and CRV = 49.

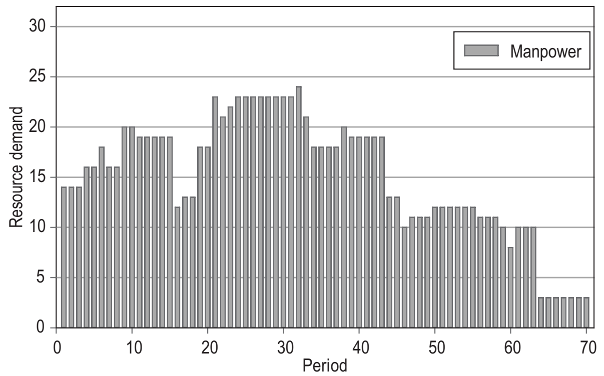

Fig. 6. Project resource profile after being optimized by FCDE-RL

Table 4. Optimal start time (ST) for all activities found by FCDE-RL

Act. ID	Opt. ST	Act. ID	Opt. ST	Act. ID	Opt. ST	Act. ID	Opt. ST
1	0	**12**	15	**23**	31	**34**	48
2	0	**13**	16	**24**	32	**35**	45
3	0	**14**	20	**25**	34	**36**	49
4	0	**15**	18	**26**	38	**37**	52
5	0	**16**	22	**27**	18	**38**	55
6	0	**17**	23	**28**	43	**39**	55
7	10	**18**	24	**29**	37	**40**	58
8	8	**19**	25	**30**	40	**41**	60
9	5	**20**	29	**31**	46	**42**	63
10	3	**21**	29	**32**	43	**43**	64
11	3	**22**	31	**33**	43	**44**	70

4.2. Result comparisons

The results comparison between FCDE-RL and the project management software, Microsoft Project 2010, is shown in the Table 5. Obviously, the performance of the new model is significantly better than that of the commercial software in terms of M_x, $RD_{\max}$, $RV_{\max}$ and CRV. This means that the new model has reduced the resource fluctuation considerably.

Table 5. Result comparison between FCDE-RL and Microsoft Project 2010

Methods	M_x	$RD_{\max}$	$RV_{\max}$	CRV
FCDE-RL	9215	24	7	49
Microsoft Project 2010	9717	24	11	125

To better verify the performance of the proposed model (FCDE-RL), three different algorithms are used for performance comparison: standard DE (DE) (Price *et al.* 2005), CDE, FDE, Genetic Algorithm (GA) (Haupt, R. L., Haupt, S. E. 2004), and Particle Swarm Optimization (PSO) (Clerc 2006). To evaluate the stability and accuracy of each algorithm, the optimization performance is expressed in terms of best result found (best), average result (avg), standard deviation (std), and worst result (worst) after 25 times of running (see Table 6). The best and worst results demonstrate the capacity of the algorithms in finding the optimal solution in all of the metrics for performance measurement. Average and standard deviation are two additional characteristics that describe solution quality. The standard deviation occurs in cases when the algorithms are not able to generate optimal solutions in all executions.

Observing from Table 6, the performance of the newly developed model is competitive in terms of accuracy and stability. It is clearly shown that the FCDE-RL and variants of DE are able to find optimal solutions in overall fitness function. Moreover, in terms of the average results, FCDE-RL performed the best as it generated the lowest average fitness solution with a value of 9522.4 with 0.84 deviation value.

The proposed algorithm achieves the best results in all of evaluation functions. FCDE-RL successfully diminishes the moment of resource histogram, maximum resource demand, and deviation of resource between consecutive periods. In terms of the moment of resource histogram, among algorithms only variants of DE are capable of finding the best results. FCDE-RL demonstrated the stability and accuracy since the average and standard deviation of results obtained from the new model are always smaller than that of other algorithms. As clearly shown in the table, all algorithms succeeded in obtaining the best solutions in term of maximum resource demand with a value of 24 and in terms of deviation of resource between consecutive periods with a value of 7. However, the FCDE-RL outperformed other benchmarked algorithms with respect to stability since it always gains the best value in all executions.

Table 6. Result comparison between FCDE-RL and benchmarked algorithms

Performance Measurement		PSO	GA	DE	CDE	FDE	FCDE-RL
Fitness	Best	9534.0	9538.0	**9522.0**	**9522.0**	**9522.0**	**9522.0**
	Avg.	9579.5	9566.5	9529.0	9526.6	9524.0	**9522.4**
	Std.	23.46	15.40	3.68	3.53	2.49	**0.84**
	Worst	9618.0	9596.0	9532.0	9532.0	9528.0	**9524.0**
M_x	Best	9223.0	9231.0	9217.0	**9215.0**	**9215.0**	**9215.0**
	Avg.	9262.9	9253.3	9222.0	9219.0	9216.6	**9215.4**
	Std.	23.73	14.48	5.52	5.66	3.50	**0.84**
	Worst	9303.0	9281.0	9231.0	9229.0	9225.0	**9217.0**
$RD_{\max}$	Best	**24.0**	**24.0**	**24.0**	**24.0**	**24.0**	**24.0**
	Avg.	24.3	24.0	24.0	24.0	24.0	**24.0**
	Std.	0.95	0.00	0.00	0.00	0.00	**0.00**
	Worst	27.0	24.0	24.0	24.0	24.0	**24.0**
$RV_{\max}$	Best	7.0	7.0	7.0	7.0	7.0	**7.0**
	Avg.	7.6	7.4	7.2	7.0	7.0	**7.0**
	Std.	1.35	0.84	0.63	0.00	0.00	**0.00**
	Worst	11.0	9.0	9.0	7.0	7.0	**7.0**
CRV	Best	63.0	67.0	51.0	51.0	**49.0**	**49.0**
	Avg.	76.6	73.2	65.8	63.2	63.2	**52.2**
	Std.	9.74	5.12	6.88	7.21	7.27	**3.43**
	Worst	93.0	85.0	73.0	71.0	69.0	**57.0**

4.3. Resource leveling on different objectives

In order to investigate the impact of using different resource leveling objective functions on the resource utilization histograms, nine different objective functions (Damci, Polat 2014) were optimized by the proposed model FCDE-RL. Table 7 presents the initial values of nine objective functions calculated using the earliest start times of all activities and the optimal values of these objectives, which were determined after resource leveling for each of these objective functions. It is shown in Table 7 that almost the values of nine objective functions were improved after resource leveling when compared to their initial values. Figure 7 provides the project's resource profile after being optimized by proposed model. It can be seen that each of the objective functions generates different resource utilization histograms after resource leveling. In other words, nine different objective functions bring about nine different solutions since each of the objective functions aims to minimize different parameters (i.e. (1) the sum of the absolute deviations in daily resource usage; (2) the sum of only the increases in daily resource usage from one day to the next; (3) the sum of the absolute deviations between daily resource usage and the average resource usage; (4) the maximum daily resource usage; (5) the maximum deviation in daily resource usage; (6) the maximum absolute deviation between daily resource usage and the average resource usage; (7) the sum of the square of daily resource usage; (8) the sum of the square of the deviations in daily resource usage; (9) the sum of the square of the deviations between daily resource usage and the average resource usage). In order to determine the improvement levels achieved by resource leveling using nine different objective functions, the improvement percentage in each objectives and the average improvement percentage for each objective function were calculated in Table 8. The important level of each objectives considered in this study is equal. As shown in Table 8, the objective function 2 provided the best average improvement percentage (50.5%) when compared to the other objective functions. Although the objective function 2 yields the best average improvement percentage in the case study, another objective function(s) may produce the highest value of improvement in different projects. Hence, the project managers should run the model for different objective functions in order to decide the objective function that provides the best improvement. Also, project managers may assign different weights to these objective depending on their special demands and determine the levelled resource utilization histogram that best fit their demands.

Table 7. Result for different objective functions optimized by FCDE-RL

No	Before Leveling	After Leveling by FCDE-RL								
		$\sum_{i=1}^{T}\lvert Rdev_i\rvert$	$\sum_{i=1}^{T}\lvert Rinc_i\rvert$	$\sum_{i=1}^{T}\lvert R_i - A_{rr}\rvert$	$\left[\max(R_i)\right]$	$\left[\max(Rdev_i)\right]$	$\left[\max(\lvert R_i - A_{rr}\rvert)\right]$	$\sum_{i=1}^{T}(R_i)^2$	$\sum_{i=1}^{T}(Rdev_i)^2$	$\sum_{i=1}^{T}(R_i - A_{rr})^2$
1	161	49	49	89	130	97	118	77	54	67
2	75	21	20	39	64	46	58	33	25	28
3	468.23	384.00	380.00	349.77	431.54	432.69	422.23	351.54	392	351.54
4	30	24	24	24	24	30	27	24	24	24
5	13	7	7	9	11	7	12	7	7	7
6	14.89	12.11	12.11	12.11	12.11	14.89	12.11	12.11	12.11	12.11
7	20274	18674	18646	18458	19334	19432	19198	18430	18842	18430
8	1201	183	187	421	804	435	660	329	160	269
9	4283.09	2683.09	2655.09	2467.09	3343.09	3441.09	3207.09	2439.09	2851.09	2439.09

Note: i = day under consideration; T = total project duration; $Rdev_i$ = deviation between resources require on day i and i+1; $Rinc_i$ increase in between resources require on day i and i+1; A_{rr} average resource use; R_i resources requires on day i.

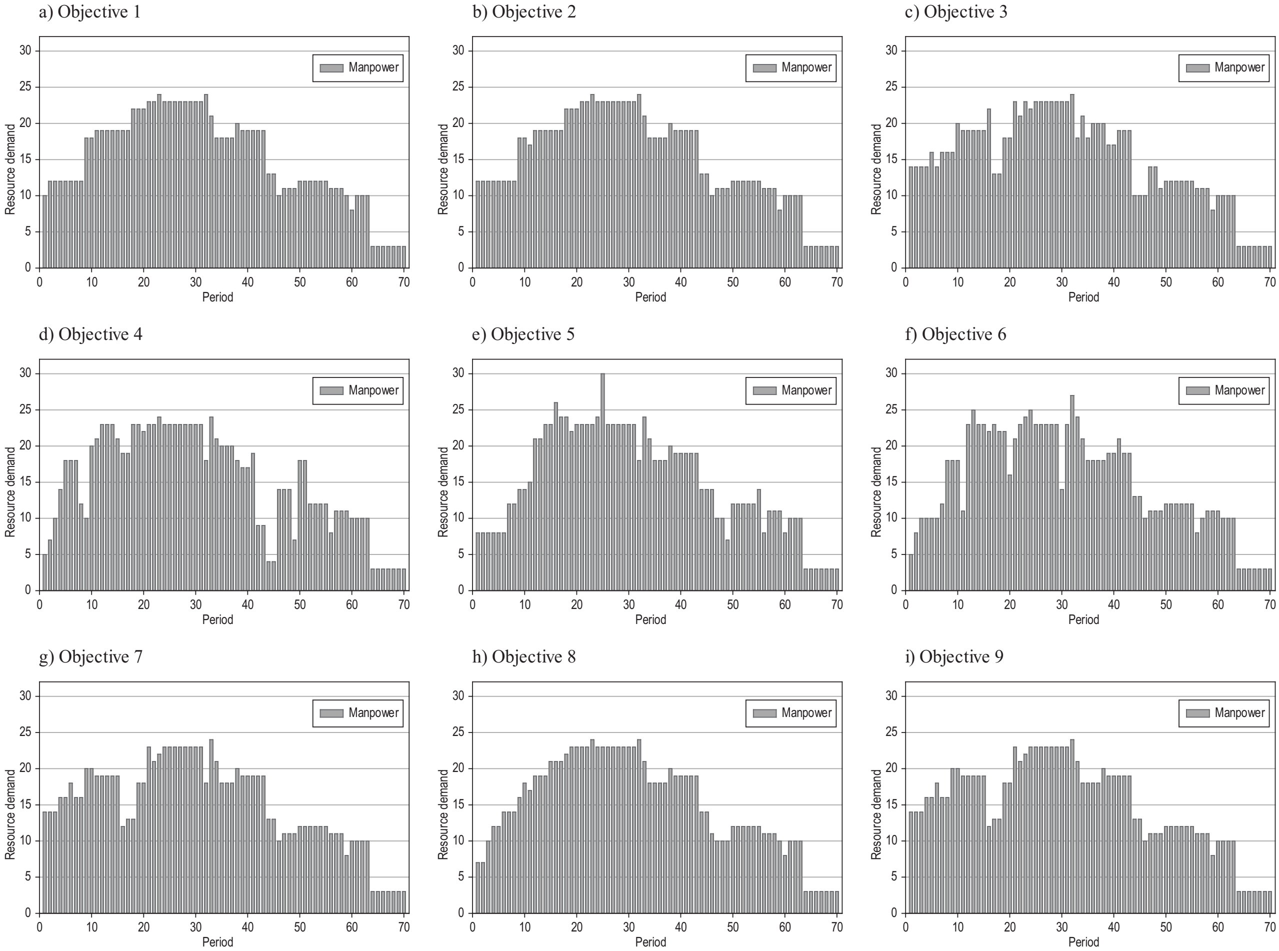

Fig. 7. Resource histogram after levelling in different objective functions optimized by FCDE-RL

Table 8. Percentages of improvement in different objective functions optimized by FCDE-RL

No	The objective functions and the improvement percentages									
	1	2	3	4	5	6	7	8	9	Average
1	69.6%	69.6%	44.7%	19.3%	39.8%	26.7%	52.2%	66.5%	58.4%	49.6%
2	72.0%	73.3%	48.0%	14.7%	38.7%	22.7%	56.0%	66.7%	62.7%	50.5%
3	18.0%	18.8%	25.3%	7.8%	7.6%	9.8%	24.9%	16.3%	24.9%	17.1%
4	20.0%	20.0%	20.0%	20.0%	–	10.0%	20.0%	20.0%	20.0%	16.7%
5	46.2%	46.2%	30.8%	15.4%	46.2%	7.7%	46.2%	46.2%	46.2%	36.8%
6	18.7%	18.7%	18.7%	18.7%	–	18.7%	18.7%	18.7%	18.7%	16.6%
7	7.9%	8.0%	9.0%	4.6%	4.2%	5.3%	9.1%	7.1%	9.1%	7.1%
8	84.8%	84.4%	64.9%	33.1%	63.8%	45.0%	72.6%	86.7%	77.6%	68.1%
9	37.4%	38.0%	42.4%	21.9%	19.7%	25.1%	43.1%	33.4%	43.1%	33.8%

Conclusions

This paper uses FCDE to solve the resource leveling problem. Integrating the fuzzy clustering and chaos algorithms into the DE effectively eliminated the drawbacks of the original DE. Chaos algorithm's randomness enhanced the population diversity and avoided being trapped at local optima, while fuzzy c-means clustering improved the convergence speed of the search algorithm provided by the moving cluster centers. The real construction project is used to validate the proposed model. Experimental results and results comparisons indicate that the FCDE-RL is able to effectively and efficiently improve the performance of the original DE beyond the levels of performance attainable by other benchmark algorithms.

Furthermore, this article has investigated nine objective functions for the resource leveling. Obviously, each of these objective functions produces different resource histogram. The improvement percentage index was used to evaluate the improvement degrees obtained by the nine objective functions. In this case study, the objective function 2 that involves the sum of only the increases in the daily resource usage from two consecutive days provides the best improvement on average. Since the objective function(s) which yields the best result on a certain project may be not consistent, the project managers should appraise all of the functions to identify the desirable one.

The FCDE-RL has broad application potential because the model is easily modifiable for solving many other classes of single-objective optimization problems in the construction management field such as resource allocation and resource constrained. Moreover, resource leveling problems in the realm of total project cost minimization are frequently encountered in construction management. Trade-offs between time and cost are necessary to improve overall construction project benefits. Further work is necessary to address these issues in order to apply FCDE to the resolution of complicated RLs that consider multi-objective optimizations. Extending the current model FCDE from a single-objective to multi-objective format using multiple objective differential evolution theory represents an interesting direction for further research.

References

Alami, J.; Imrani, A. E.; Bouroumi, A. 2007. A multipopulation cultural algorithm using fuzzy clustering, *Applied Soft Computing* 7(2): 506–519. http://dx.doi.org/10.1016/j.asoc.2006.10.010

Arditi, D.; Pattanakitchamroon, T. 2006. Selecting a delay analysis method in resolving construction claims, *International Journal of Project Management* 24(2): 145–155. http://dx.doi.org/10.1016/j.ijproman.2005.08.005

Assaf, S. A.; Al-Hejji, S. 2006. Causes of delay in large construction projects, *International Journal of Project Management* 24(4): 349–357. http://dx.doi.org/10.1016/j.ijproman.2005.11.010

Becerra, R. L.; Coello, C. A. C. 2006. Cultured differential evolution for constrained optimization, *Computer Methods in Applied Mechanics and Engineering* 195(33–36): 4303–4322. http://dx.doi.org/10.1016/j.cma.2005.09.006

Bedri Ozer, A. 2010. CIDE: Chaotically initialized differential evolution, *Expert Systems with Applications* 37(6): 4632–4641. http://dx.doi.org/10.1016/j.eswa.2009.12.045

Bezdek, J. C.; Ehrlich, R.; Full, W. 1984. FCM: The fuzzy c-means clustering algorithm, *Computers & Geosciences* 10(2–3): 191–203. http://dx.doi.org/10.1016/0098-3004(84)90020-7

Jiang, B. L. W. 1998. Optimizing complex functions by chaos search, *Cybernetics and Systems* 29(4): 409–419. http://dx.doi.org/10.1080/019697298125678

Cai, Z.; Gong, W.; Ling, C. X.; Zhang, H. 2011. A clustering-based differential evolution for global optimization, *Applied Soft Computing* 11(1): 1363–1379. http://dx.doi.org/10.1016/j.asoc.2010.04.008

Cheng, M.-Y.; Huang, K.-Y.; Chen, H.-M. 2012. Dynamic guiding particle swarm optimization with embedded chaotic search for solving multidimensional problems, *Optimization Letters* 6(4): 719–729. http://dx.doi.org/10.1007/s11590-011-0297-z

Christodoulou, S.; Ellinas, G.; Michaelidou-Kamenou, A. 2010. Minimum moment method for resource leveling using entropy maximization, *Journal of Construction Engineering and Management* 136(5): 518–527. http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000149

Clerc, M. 2006. *Particle swarm optimization*. Wiley-ISTE. 243 p. http://dx.doi.org/10.1002/9780470612163

Damci, A.; Polat, G. 2014. Impacts of different objective functions on resource leveling in construction projects: a case study, *Journal of Civil Engineering and Management* 20(4): 1–11. http://dx.doi.org/10.3846/13923730.2013.801909

Das, S.; Suganthan, P. N. 2011. Differential evolution: a survey of the state-of-the-art, *IEEE Transactions on Evolutionary Computation* 15(1): 4–31. http://dx.doi.org/10.1109/TEVC.2010.2059031

Deb, K. 2005. A population-based algorithm-generator for real-parameter optimization, *Soft Computing* 9(4): 236–253. http://dx.doi.org/10.1007/s00500-004-0377-4

Doulabi, S. H. H.; Seifi, A.; Shariat, S. Y. 2011. Efficient hybrid genetic algorithm for resource leveling via activity splitting, *Journal of Construction Engineering and Management* 137(2): 137–146. http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000261

Easa, S. M. 1989. Resource leveling in construction by optimization, *Journal of Construction Engineering and Management* 115(2): 302–316. http://dx.doi.org/10.1061/(ASCE)0733-9364(1989)115:2(302)

El-Rayes, K.; Jun, D. 2009. Optimizing resource leveling in construction projects, *Journal of Construction Engineering and Management* 135(11): 1172–1180. http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000097

Foekstistov, V. 2006. *Differential evolution in search of solutions*. Springer Science+Business Media, LLC. 196 p. http://dx.doi.org/10.1007/978-0-387-36896-2_1

Geng, J.-Q.; Weng, L.-P.; Liu, S.-H. 2011. An improved ant colony optimization algorithm for nonlinear resource-leveling problems, *Computers & Mathematics with Applications* 61(8): 2300–2305. http://dx.doi.org/10.1016/j.camwa.2010.09.058

Georgy, M. E. 2008. Evolutionary resource scheduler for linear projects, *Automation in Construction* 17(5): 573–583. http://dx.doi.org/10.1016/j.autcon.2007.10.005

Harris, R. B. 1990. Packing method for resource leveling (Pack), *Journal of Construction Engineering and Management* 116(2): 331–350. http://dx.doi.org/10.1061/(ASCE)0733-9364(1990)116:2(331)

Haupt, R. L.; Haupt, S. E. 2004. *Practical genetic algorithm*. John Wiley & Sons, Inc. 272 p. http://dx.doi.org/10.1002/0471671746.fmatter

Hegazy, T. 1999. Optimization of resource allocation and leveling using genetic algorithms, *Journal of Construction Engineering and Management* 125(3): 167–175. http://dx.doi.org/10.1061/(ASCE)0733-9364(1999)125:3(167)

Jain, A. K.; Murty, M. N.; Flynn, P. J. 1999. Data clustering: a review, *ACM Computing Surveys* 31(3): 264–323. http://dx.doi.org/10.1145/331499.331504

Jia, D.; Zheng, G.; Khurram Khan, M. 2011. An effective memetic differential evolution algorithm based on chaotic local search, *Information Sciences* 181(15): 3175–3187. http://dx.doi.org/10.1016/j.ins.2011.03.018

Sears, S. K.; Sears, G. A.; Clough, R. H. 2008. *Construction project management: a practical guide to field construction management*. 5th ed. New Jersey: John Wiley and Sons, Inc. 408 p.

Karaa, F. A.; Nasr, A. Y. 1986. Resource management in construction, *Journal of Construction Engineering and Management* 112(3): 346–357. http://dx.doi.org/10.1061/(ASCE)0733-9364(1986)112:3(346)

Kim, J. H.; Stringer, J. 1992. *Applied chaos*. Electric Power Research Institute, Palo Alto, California, USA. 576 p.

Kwedlo, W. 2011. A clustering method combining differential evolution with the K-means algorithm, *Pattern Recognition Letters* 32(12): 1613–1621. http://dx.doi.org/10.1016/j.patrec.2011.05.010

Leu, S.-S.; Yang, C.-H.; Huang J.-C. 2000. Resource leveling in construction by genetic algorithm-based optimization and its decision support system application, *Automation in Construction* 10(1): 27–41. http://dx.doi.org/10.1016/S0926-5805(99)00011-4

Martinez, J.; Ioannou, P. 1993. Resource leveling based on the modified minimum moment heuristic, in *Proceedings of Computing in Civil and Building Engineering*, 1993, ASCE, New York, USA, 287–294.

Noman, N.; Iba, H. 2008. Accelerating differential evolution using an adaptive local search, *IEEE Transactions on Evolutionary Computation* 12(1): 107–125. http://dx.doi.org/10.1109/TEVC.2007.895272

Pang, N.; Shi, Y.; You, Y. 2008. Resource leveling optimization of network schedule based on particle swarm optimization with constriction factor, in *International Conference on Advanced Computer Theory and Engineering (ICACTE'08)*, 20–22 December, 2008, Phuket, Thailand, 652–656. http://dx.doi.org/10.1109/ICACTE.2008.124

Ponz-Tienda, J. L.; Yepes, V.; Pellicer, E.; Moreno-Flores, J. 2013. The resource leveling problem with multiple resources using an adaptive genetic algorithm, *Automation in Construction* 29: 161–172. http://dx.doi.org/10.1016/j.autcon.2012.10.003

Price, K. V.; Storn, R. M.; Lampinen, J. A. 2005. *Differential evolution – a practical approach to global optimization*. Berlin: Springer-Verlag. 292 p. http://dx.doi.org/10.1007/3-540-31306-0

Savin, D.; Alkass, S.; Fazio, P. 1996. Construction resouce leveling based using neural networks, *Canadian Journal of Civil Engineering* 23(3): 917–925. http://dx.doi.org/10.1139/l96-898

Son, J.; Skibniewski, M. 1999. Multiheuristic approach for resource leveling problem in construction engineering: hybrid approach, *Journal of Construction Engineering and Management* 125(1): 23–31. http://dx.doi.org/10.1061/(ASCE)0733-9364(1999)125:1(23)

Storn, R.; Price, K. 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* 11: 341–359. http://dx.doi.org/10.1023/A:1008202821328

Wang, Y.-J.; Zhang, J.-S.; Zhang, G.-Y. 2007. A dynamic clustering based differential evolution algorithm for global optimization, *European Journal of Operational Research* 183(1): 56–73. http://dx.doi.org/10.1016/j.ejor.2006.10.053

Weiguo, S.; Swift, S.; Leishi, Z.; Xiaohui, L. 2005. A weighted sum validity function for clustering with a hybrid niching genetic algorithm, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 35(6): 1156–1167. http://dx.doi.org/10.1109/TSMCB.2005.850173

Yan, L.; Sheng-Li, Z.; Xi-Kai, D.; Shu-Quan, L. 2005. Optimization of resource allocation in construction using genetic algorithms, in *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 18–21 August, 2005, Guangzhou, China, 6: 3428–3432. http://dx.doi.org/10.1109/ICMLC.2005.1527534

Yin, P.-Y.; Wang, J.-Y. 2006. Ant colony optimization for the nonlinear resource allocation problem, *Applied Mathematics and Computation* 174(2): 1438–1453. http://dx.doi.org/10.1016/j.amc.2005.05.042

Zhang, J.; Sanderson, A. C. 2009. JADE: Adaptive differential evolution with optional external archive, *IEEE Transactions on Evolutionary Computation* 13(5): 945–958. http://dx.doi.org/10.1109/TEVC.2009.2014613

Min-Yuan CHENG. Distinguished Professor of Department of Civil and Construction Engineering at the National Taiwan University of Science and Technology. An Editorial Board member of Automation in Construction Journal. His research interests include construction business process re-engineering, geographic information system, construction automation and e-business for construction management, project management information system, artificial intelligence applications in construction management.

Duc-Hoc TRAN. He is currently a lecturer at Faculty of Project Management, University of Science and Technology-The University of Da Nang. He got the MSc. and PhD degrees at National Taiwan University of Science and Technology. Research interests: project scheduling, scheduling theory, intelligent optimization method; multiple objective optimization; and artificial intelligence applications in construction management.

Nhat-Duc HOANG. He is currently a researcher and lecturer at Institute of Research and Development, Duy Tan University. He got the MSc and PhD degrees at National Taiwan University of Science and Technology. He has published papers in various international journals such as Journal of Civil Engineering and Management, Automation in Construction, and Journal of Computing in Civil Engineering. His research focuses on applications of Artificial Intelligence in construction engineering and management.