KSCE Journal of Civil Engineering (2016) 20(1):90-100

DOI 10.1007/s12205-015-0414-0

Construction Management

pISSN 1226-7988, eISSN 1976-3808
www.springer.com/12205

An Efficient Hybrid Differential Evolution based Serial Method for Multimode Resource-Constrained Project Scheduling

Min-Yuan Cheng* and Duc-Hoc Tran**

Received July 27, 2014/Revised January 14, 2015/Accepted February 24, 2015/Published Online April 3, 2015

Abstract

The Multimode Resource-Constrained (MRC) problem aims at finding the start times and execution modes for the activities of a project that minimizes project duration under current precedence constraints and resource limitations. This study integrates the fuzzy c-means clustering technique and the chaotic technique into the Differential Evolution to develop the Fuzzy Clustering Chaotic-based Differential Evolution (FCDE) algorithm, an efficient hybrid approach to solving MRC and other related problems. Within the FCDE, the chaos prevents the optimization algorithm from premature convergence and the fuzzy c-means clustering acts as several multi-parent crossover operators for utilizing population information efficiently and enhance convergence efficiency. Further, this study applies a serial method to reflect individual-user priorities into the active schedule and the project duration calculations. Experiments run indicate that the proposed FCDE-MRC obtains optimal results more reliably and efficiently than the benchmark algorithms considered. The FCDE-MRC is a promising alternative methodology to handling resource-constrained problems.

Keywords: *resource-constrained, fuzzy clustering, chaotic, differential evolution, construction management*

1. Introduction

In today's competitive business environment, companies cannot survive without efficiently planning and scheduling. Resource-constrained project scheduling is a key challenge for many industrial problems. Project managers widely use the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) to generate schedules that are used in planning and managing large-scale construction projects. These two approaches involve logical dependencies because they assume infinite resource availability. However, assuming infinite resource availability is not reasonable for most real-world construction projects (Zhang *et al.,* 2006a).

A classical Resource Constrained Project Scheduling Problem (RCPSP) involves the scheduling of project activities subject to resource and precedence constraints, under the objective of minimizing the project duration. Each activity has a single execution mode, that is, the resource requirements and associated duration for an activity are fixed (Zhang *et al.*, 2006b). However, in the real project when limited resources are shared by multiple activities and the real-time amounts of available resources are thus flexible, some activities may have multiple execution modes to be selected, each mode having different resource amounts corresponding to different durations for each activity (Rostami *et al.*, 2014; Zhang, 2012). Multiple selections of execution modes are considered to reduce delay of activities and idle times of resources in consideration of real-time amounts of resources (Li and Zhang, 2013). It has been shown that the research on MRC project scheduling is crucial and has been increasingly paid attention in the construction filed.

Using Evolutionary Algorithms (EAs) to analyze multimode resource-constrained problems has attracted increasing attention in recent years (Elloumi and Fortemps, 2010). Inspired by the process of natural evolution, EAs have been used successfully to resolve optimization problems in diverse fields (Liao *et al.*, 2011). Genetic Algorithms (GAs) are an evolutionary approach used widely to solve the MRC (Alcaraz and Maroto, 2001; Alcaraz *et al*., 2003; Lova *et al*., 2009; Peteghem and Vanhoucke, 2010). The GA searches for the optima in multiple chromosome generations that represent schedules reproduced by crossover and mutation. The internal updating mechanism of chromosomes enables GA to search for the global optima. However, deficiencies in GA performance such as premature convergence and slow convergence have been identified. Particle Swarm Optimization (PSO), an algorithm that simulates bird flocking behavior, has been applied to solve the MRC problems (Jarboui *et al*., 2008; Zhang *et al.*, 2006c; Zhang and Xu, 2014). Like GA, PSO first initializes a population of random solutions and then updates

*Professor, Dept. of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan (E-mail: myc@mail.ntust.edu.tw)

**Ph.D. Student, Dept. of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan (Corresponding Author, E-mail: duchoc87@gmail.com)

generations to search for the optima. PSO advantages over GA include relative ease of implementation, faster search process, and more effective performance (Elbeltagi *et al.*, 2005). Nevertheless, similar to other stochastic search methods, PSO may become trapped in a local minimum and thus may resolve upon a local rather than a global minimum.

The Differential Evolution (DE) (Price *et al.*, 2005; Storn and Price, 1997) algorithm is an evolutionary computation technique. DE has been drawn increasing interest from researchers, who have explored the capabilities of this algorithm in a wide range of problems. DE is an effective population-based stochastic search engine for global optimization in the continuous domain. DE uses mutation operators, crossover operators, and selection operators at each generation to move its population toward the global optimum. The superior performance of DE over competing algorithms has been verified in many reported research works (Cheng and Tran, 2014; Damak *et al*., 2009; Wu *et al*., 2010).

Despite the aforementioned advantages, DE in both its original form and many later variants has several drawbacks. DE does not guarantee convergence to the global optimum. It is easily trapped into the local optima, resulting in low optimization precision or even failure (Jia *et al.*, 2011). Further, because a population may not be distributed over the search space, individuals may be trapped in a local solution. Therefore, DE may require more generations to converge toward the optimal or near-optimal solution (Ozer, 2010). DE is also subject to various weaknesses, especially if the global optimum is identified in a small number of fitness evaluations. Moreover, although DE is good at exploring the search space and locating the region of the global minimum, it is slow at exploiting the solution (Noman and Iba, 2008).

The inherent characteristics of chaotic systems provide an efficient approach to maintaining search algorithm population diversity. Chaos is defined as behavior that is apparently unpredictable and random as exhibited by a deterministic nonlinear system under deterministic conditions. Chaotic systems that are sensitive to small differences in initial system conditions may produce large variances in outcomes. This is a property of instability sometimes referred to as the butterfly effect or Liapunove's sense (Kim and Stringer, 1992). Some studies have focused on hybridizing DE with a chaotic algorithm. For example, Jia *et al.* (2011) used a Chaotic Local Search (CLS) with a 'shrinking' strategy. The CLS improves 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 late-run phase to refine the final solutions. Ozer (2010) embedded seven chaotic maps to create the initial population of the DE algorithm. Findings of these studies indicate that coupling emergent results in different areas such as those of DE and complex dynamics may improve the quality of results in some optimization problems.

Fuzzy c-means clustering is a soft clustering approach that divides a set of objects into groups or clusters of similarities to accelerate the optimization search in DE. Successful clustering identifies the true natural groupings in the dataset. Fuzzy c-means clustering helps track the evolution of the search algorithm in DE through cluster centers introduced into the populations. In fuzzy clustering, data elements may belong to more than one cluster, with a set of membership levels associated with each element. Membership levels indicate the strength of the association between a data element and a particular cluster. Kwedlo (2011) proposed a new version of DE that uses k-means clustering to fine-tune each candidate solution obtained by DE mutation and crossover operators. Wang *et al.* (2007) utilized a clustering technique to improve solution accuracy with less computational effort. Experiments indicate that the new method is able to find near-optimal solutions efficiently.

Hybridization using other algorithms is an interesting direction to further improve DE (Cai *et al.,* 2011). Although many methods have been proposed to improve DE, few researchers have studied DE hybridization with clustering techniques and chaotic techniques (Cai *et al.*, 2011). An extensive review of the literature done for this study found that fuzzy c-means clustering and chaotic techniques have not yet been used to enhance the performance of DE.

This study uses a hybridization strategy to improve the DE optimizer. This hybridization strategy incorporates the fuzzy c-means clustering technique and the chaotic technique to overcome performance problems inherent to the original DE. Chaotic sequences are adopted instead of random sequences, with good results exploited to prevent premature convergence. Further, fuzzy c-means clustering introduces multi-parent crossover operators to population information in order to accelerate algorithm convergence. The remainder of this paper is organized as follows. Section 2 provides a brief overview of the literature related to the new optimization model. In the section 3, the detailed descriptions of the new model are presented; subsequently, the performance of the new model is demonstrated using numerical experiments, result comparisons and sensitivity analyses in section 4. Finally, conclusions and further studies are given in section 5.

2. Literature Review

2.1 Formulating the Multimode Resource-constrained

The problem considered in this research is the non-preemptive MRC, in which the objective is to minimize total project duration. This problem can be formulated as follows:

A project consists of a set J of N non-dummy activities, $J = \{1, 2, \ldots, N\}$. Each activity $j \in J$ can be executed in one of M_j modes. Each execution mode represents an alternative combination of resource requirements and activity duration. The duration of activity *j*, when performed in mode m $(m = 1, \ldots, M_j)$ is d_{jm}. The activities in progress are not allowed to be interrupted. Activity *j*, executed in mode *m*, requires a certain amount of the renewable resource $k \in R \in \{1, \ldots, K\}$ for each period executed. The amount per period is given by r_{jml}. For each resource $k \in R$, its availability per period is constant and given by AR_k. In addition, activity *j*, executed in mode *m* uses a total amount of nr_{jml} units of the non-renewable resource $l \in NR = \{1, \ldots, L\}$.

For each non-renewable resource l, the overall availability for the entire project is represented by ANR_l. Precedence relations can be found among some of the activities. Activity j cannot start before all its predecessors, given by the set P_j, are completely finished. The values of AR_k, ANR_l, d_{jm}, r_{jmk} and nr_{jmk} (resource availability, duration of activities and resource requirements per activity) are integer and non-negative.

The aim is to determine the execution mode and starting time of each activity, so that the schedule is feasible with respect to precedence and the resource constraints, and the total project duration is minimized. A solution can be presented as a vector $So = \{s_1, s_2, \ldots, s_N;\ e_1, e_2, \ldots, e_N\}$, where s_j, e_j denote the start time, execution mode of activity, respectively.

The scope of this study is limited to a deterministic environment in which decisions are made with deterministic input data from the case study. The assumptions of the case study are that the mean values of the resources are used and the logical relationship amongst activities is 'finish-to-start' since it is the most popular dependency used by project teams. The proposed scheduling tool in this paper focused mainly on deterministic and static situations. It differs from the simulation based-resource constrained scheduling methods which is used for stochastic and dynamical natures of construction processes (Lu *et al.*, 2008; Zhang *et al.*, 2002). Nonetheless, the simulation based-resource constrained scheduling methods lies beyond the scope of the present paper.

2.2 The DE optimization Algorithm

Differential Evolution (DE) is a simple population-based, direct-search for solving global optimization problems (Price *et al.*, 2005; Storn and Price, 1997). 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: $X_{i,G} = \{x^1_{i,G}, x^2_{i,G}, \ldots x^D_{i,G}\},\ i = 1, 2, \ldots, NP$ 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.

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

where, $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$ 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 $U_{i,G+1} = \{u_{i,G+1}, u_{i,G+1}, \ldots, u_{i,G+1}\}$ is generated, using the following scheme.

$$u^j_{i,G+1} = \begin{cases} v^j_{i,G+1} & if(rand_j[0,1) \leq CR \;\; or \;\; j = j_{rand} \\ x^j_{i,G} & otherwise \end{cases} \quad j = 1, 2, \ldots, D \tag{2}$$

$CR \subset [0, 1]$ is user-defined crossover constant; j_{rand} is a randomly chosen index from $\{1, 2, \ldots, D\}$, 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:

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & if \ f(U_{i,G+1}) < f(X_{i,G}) \\ X_{i,G} & otherwise \end{cases} \tag{3}$$

Once all the individuals of the next generation are selected, the evolutionary cycle of the DE iterates until a stopping condition is satisfied.

3. Fuzzy c-means Clustering Chaotic-based Differential Evolution for the Multimode Resource Constrained (FCDE-MRC)

This section describes the newly proposed FCDE optimization algorithm for solving the MRC problem in detail. The FCDE is the core optimizer in the FCDE-MRC model. Our algorithm was developed based on standard DE (Price *et al*., 2005; Storn and Price, 1997) by integrating the original DE with fuzzy c-means clustering and chaotic techniques. The chaos approach effectively exploits the whole search space and provides the diversity necessary in the DE population. Consequently, this operation incurs additional time and iteration to search for the global optimum. On the contrary, the fuzzy c-means clustering technique enhances the convergence speed of the algorithm by introducing cluster centers. These moving centers provide direction to the global optimum search, improving overall search algorithm efficiency. The FCDE solutions are transformed into a feasible schedules through serial schedule generation scheme. Fig. 1 illustrates the proposed algorithm.

3.1 Initialization

The FCDE-MRC optimization model requires project information inputs including activity relationship; activity duration, resources demands of each execution mode. In addition, the user must provide search optimizer parameter settings such as maximum number of search generations G_{max} and population size (NP).

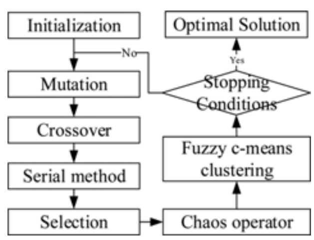


Fig. 1. Fuzzy Clustering Chaotic-based Differential Evolution for Multimode Resource Constrained (FCDE-MRC)

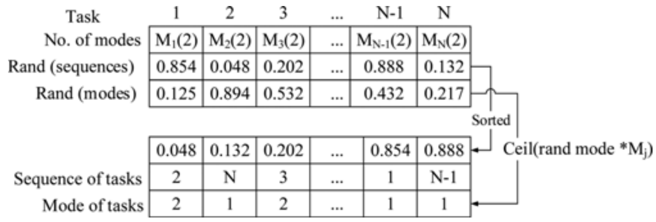


Fig. 2. Structure of FCDE Individual

With these inputs, the scheduling can carry out the calculation process to obtain the project duration and amount of resources required for each activity.

FCDE-MRC commences the search process by randomly generating population size *NP* and the maximum generation $G_{\max}$ number for the *D*-dimensional parameter vectors $X_{i,g}$, where $i = 1, 2, \ldots, NP$ 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, individuals may be simply generated as follows:

$$X_{i,0} = LB + rand[0,1]*(UB - LB) \tag{4}$$

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 representing the lower and upper bounds, respectively, for any decision variable.

A candidate solution to the MRC problem can be represented as two vectors with *2N* elements in total: the first one is a position vector which consists of the position of each task in the sequence, whereas the second is devoted to the modes assignment, as shown in Fig. 2.

It is obvious that N is also the number of activities in the project network. Since original DE operates with real-value variables, a function (ceil) is employed to convert those activities' execution modes from real values to integer values within the feasible domain.

3.2 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 Eq. (1) vector.

3.3 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 Eq. (2).

3.4 Modified Serial Method for MRC

Once the FCDE individual is created, its fitness function is calculated through serial method. The second vector of FCDE individual defines the execution mode of each activity and then determines the corresponding durations and resource requirements of all activities. The first vector of FCDE individual carries out the sequence of activities.

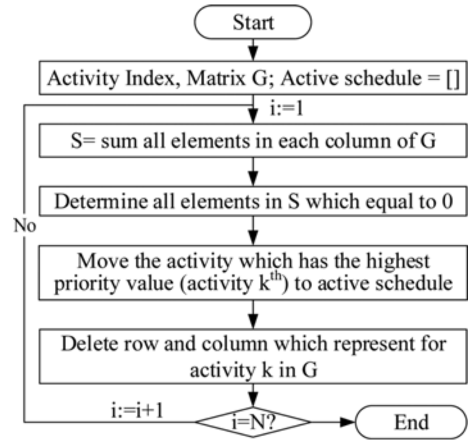


Fig. 3. Transfer to Feasible Active Schedule

The serial method was proposed by Kelly (1963) (Kolisch, 1996). It consists of n = 1,…, D stages, in which one activity is selected and scheduled in each stage. When an activity has been checked and currently available amounts of resources are adequate, this activity is scheduled at the earliest precedence time (e.g., the earliest completion times of its predecessors) and resource-feasible time. We revised the serial schedule schema for easier comprehension and implementation using the following two steps:

Step1: Transfer FCDE individual sequence of tasks priorities to an active schedule based on precedence constraints.

Denote a set of tasks in project $J = \{1, 2, \ldots, N\}$. We can define priority relations in set *J* as a set of pair $C = \{(i,j)|i$ that must be executed before $j\}$. We introduce the binary relation matrix $V = (v_{ij}, 1 \le i, j \le n)$, $v_{ij} = 1$, if $(i,j) \in C$, $v_{ij} = 0$, if $(i,j) \notin C$, related with a set of priority constraints and define a full-priority relation matrix $G = (g_{ij}, 1 \le i,\ j \le n)$. This matrix describes all priority relation chains. So, $g_{kj} = 1$ if it is possible to find such a sequence of index pairs that $(k, k_1) \in C$, $(k_1, k_2) \in C$,…, $(k_1, k_2) \in C$. The matrix V has the following property: $v_{ij} = 1 \Rightarrow v_{ji} = 1$. The G matrix shares this feature as well (Sakalauskas and Felinskas, 2006).

Step 2: Calculate project duration based on active schedule

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

Figure 5(a) is an example of an activity on node project with six activities and two dummy activities where index numbers within circle nodes represent activities, numbers above and below the circles indicate activity required amount resources, duration of two execution modes, respectively. Fig. 5(b) shows

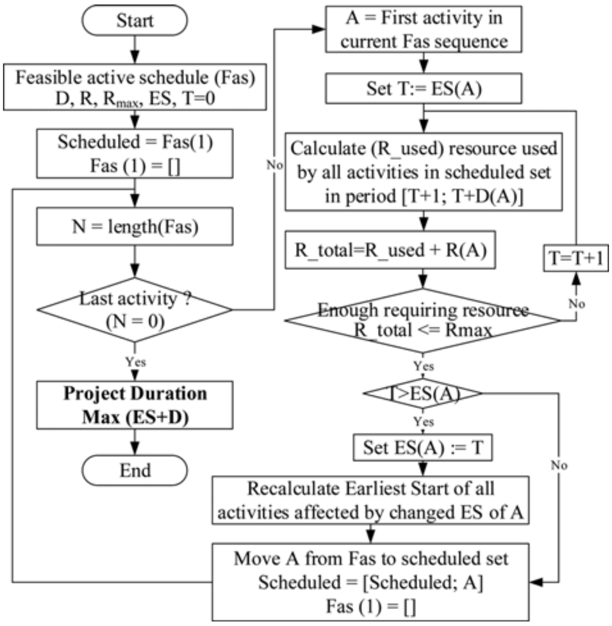
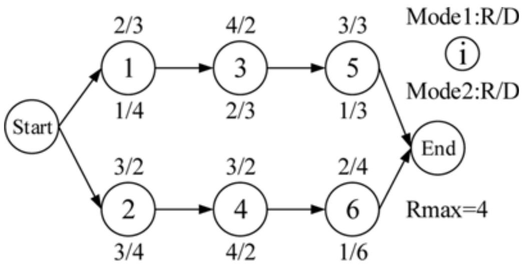


Fig. 4. Serial Method



(a) Project network

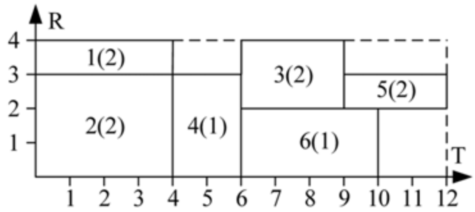
1	2	3	4	5	6	Activity index	
2	2	2	2	2	2	No. of mode	
0.96	0.73	0.80	0.14	0.32	0.92	Priority value	Individual
0.63	0.55	0.71	0.16	0.86	0.27	Mode value	
4	5	2	3	6	1	Activity seq. (sorted)	
2	2	2	1	2	1	Mode assignment	

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Individual-represented and matrix G

2(2)	4(1)	6(1)	1(2)	3(2)	5(2)
Activity(execution mode)					

(c) Feasible individual-represented sequence and mode



(d) Feasible schedule

Fig. 5. Individual and Corresponding Schedule of Project Example

the FCDE individual represented and matrix G. The activity sequence is transformed into a feasible activity schedule in Fig. 5(c) based on precedence constraints represented in matrix G. For example, task 1 will compete with task 2 to determine which task will be transferred first. Since the sum of all elements in the first and second column of G, representing activity 1 and 2, respectively, equals 0. Following the activity sequence results in the selection of task 2. The second column and row of matrix G are then deleted. In competition between task 4 and task 1, task 4 will be selected. This process continues until the feasible active individual-represented sequence is generated.

The corresponding schedule is thus determined using the serial method. Based on selected mode of all activities, at the beginning, the Earliest Start (ES) vector for all activities without considering resource constraints is calculated as [0, 0, 4, 4, 7, 6], which corresponds to activity list [1, 2, 3, 4, 5, 6]. The first activity's start time in feasible activity sequence is set equal to T (currently T = 0). The currently occupied resource in period [1; 4] is zero, so the requiring resource is satisfied. The next step, check condition (T > ES(2)), is not satisfied. Therefore, activity 2 is moved to the scheduled set and deleted from the feasible active sequence (Fas). The first activity in the Fas thus becomes activity 4. Following the same procedure as activity 2, activity 4 is moved to the scheduled set. So on for activity 6 and 1. We next consider activity 3. Set T equals 4 because ES(3) is currently equal to 4. Checking resource availability in period [4; 6] reveals this condition as unsatisfied. T is then increased to 6 to satisfy the resource-constrained condition. Hence, ES(3) is set to 6 and the ES values of all activities affected by this change are recalculated. The ES vector is computed as [0, 0, 6, 4, 9, 6]. The above calculation process continues until all activities in the Fas are moved into the scheduled set.

3.5 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 Eq. (3).

3.6 Chaotic Differential Evolution

The logistic map that generates chaotic sequences in DE, named CDE, ensures that individuals in a population are spread as evenly across the search space as possible to achieve population diversity (Cheng *et al*., 2012; Ozer, 2010). Incorporating the chaotic map into the DE has been shown to enhance global convergence by escaping the suboptimal solution. Fig. 6 shows the main steps to generate chaotic populations.

The criterion is the improvement made in the past iterations and current iteration. With a specified threshold (10 iterations), if there is no improvement on the objective function value, a percentage of the population (CF) out of total population (NP) is

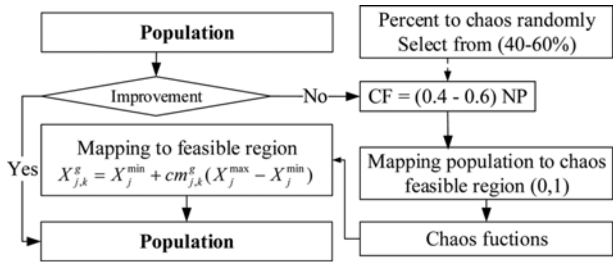


Fig. 6. Chaotic Approach

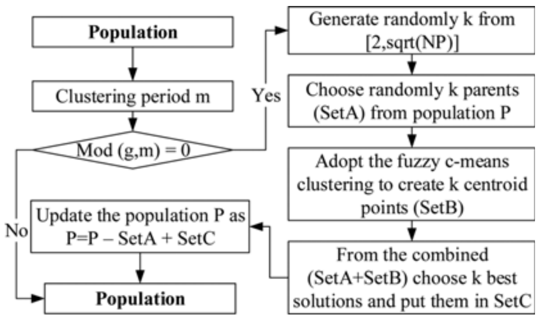


Fig. 7. Fuzzy c-means Clustering Process

selected for chaos. The population is then mapped to a chaotic-feasible region in range (0, 1) based on chaotic conditions and performs the logistic map in accordance with Eq. (5) to yield chaotic values $cm^g_{j,k}$.

$$X_{n+1} = \mu X_n(1-X_n) \tag{5}$$

In this equation, X_n is the n^{th} chaotic number, where n denotes the iteration number. Obviously, $X_n \in (0,1)$ 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 in control parameter μ in Eq. (5) directly and significantly impacts the behavior of X. [0, 4] has usually been defined as the domain area of control parameter of μ. In this study, $\mu = 4$ has been used due to the advantages of diversity during evolution (Jiang, 1998; Ohya, 1998).

Afterwards, the chaotic values are mapped to the feasible region based on Eq. (6):

$$X^g_{j,k} = X^{min}_j + cm^g_{j,k}(X^{max}_j - X^{min}_j) \quad j = 1,2,\ldots,D;\ k = 1,2,\ldots,CF \tag{6}$$

where $X^g_{j,k}$ is the j^{th} decision variable of the k^{th} individual in the CF-population at generation g and X^{max}_j and X^{min}_j are the upper bounds and lower bounds, respectively, of the j^{th} decision variable.

3.7 Fuzzy clustering Differential Evolution

The FCM clustering technique adopted in DE, named FDE, is able to efficiently converge DE. The FCM introduced in this study was intended to track the main stream of population movement during DE evolution. Each cluster center was treated approximately as one item in the main stream of evolution and replaced in the population as candidate individuals. Fig. 7 illustrates the FDE algorithm, where m is the clustering period; NP is the population size; and k is the centroid number (Cai *et al.*, 2011), an integer in the range of $[1, \sqrt{NP}]$.

Clustering is performed periodically in the FDE to efficiently exploit the search space. This clustering is similar to the method used in (Cai *et al.*, 2011). Performing clustering too early will lead to false identification of clusters. Consequently, it is important to choose a clustering period large enough to give the DE adequate time to form complete, stable clusters. In this approach, parameter m is adopted to control the clustering period.

The initial period for the clustering operator specified in the algorithm is 10. When the clustering condition is satisfied, fuzzy c-means clustering will create k offspring, which will update the population. Deb (2005) proposed a generic population-based algorithm-generator for real-parameter optimization, where the optimization task is divided into 4 independent plans: (i) selection plan, (ii) generation plan, (iii) replacement plan, and (iv) update plan. The algorithm flowchart above may also be described using the population-update algorithm proposed in Deb (2005).

- Selection plan: Choose k individuals at random from the current population (set A).
- Generation plan: Use fuzzy c-means clustering to create k offspring (set B).
- Replacement plan: Choose k best solutions (set C) from the combined (set A + set B) for replacement.
- Update plan: Update the population as P = P-Set A + Set C.

In the update plan, selecting the k best solutions from the combined set A + Set B preserves the elite members of the population.

3.8 Stopping Condition

The optimization process terminates when the user-designated stopping criterion is met. The stopping condition is often designated as the maximum generation G_{max} or maximum Number of Function Evaluations (NFE). This study used the maximum number of generation as stopping condition for the proposed algorithm. Termination of the optimization process presents the final optimal solution to the user.

3.9 Case Study

A real construction project case study named Sky Park Residence project in Viet Nam was used to demonstrate the capability of the FCDE-MRC model. All the detailed data of the project are obtained from the Licogi 16 Joint Stock Co. The project is located at the northwest gate of the capital of Hanoi. The total land area of the project is 9262 m^2, of which construction area is 3342 m^2, account for 36%. Scale the project includes two blocks, office block which is 25 stories tall, apartment block has 35 floors tall, 3 basements for car and the first five floors which are used as commercial centers. All the activities considered are extracted from the foundation and basement construction phase of apartment block building project. The execution modes and duration data for the activities are obtained based on project data and experts' experience at planning and designing steps. The

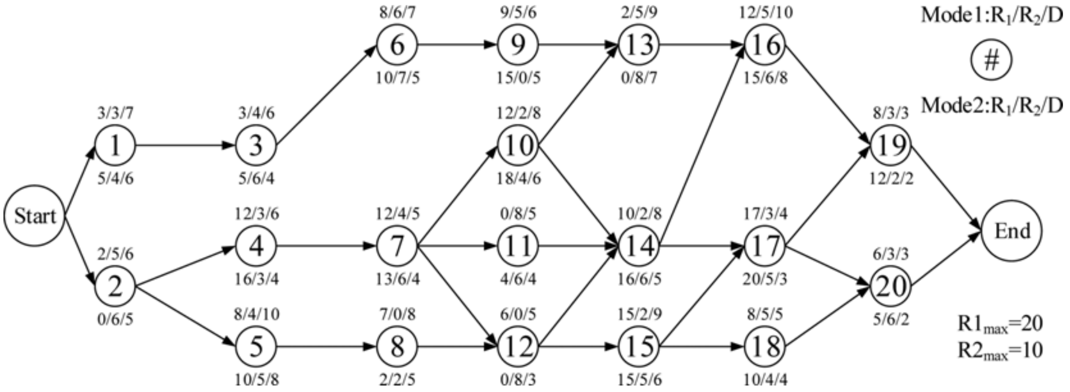


Fig. 8. Network of Project

characteristics and types of data are accuracy and quantitative, respectively. All data are collected by measuring directly from project schedules and interviewing with project planners. Test network was slightly modified to adapt to model requirements. Fig. 8 shows the precedence relationships of the project network. The case consists of 20 activities with two of selection modes for each activity. Each execution mode corresponds to resource requirements and duration for each activity is noted above and below circle node, respectively. The project uses day as the measure of duration, i.e., one day per unit. The project considers two types of renewable resources (crew and equipment) and a maximum availability amounts of two types of resources are 20 and 10. The precedence constraints among activities are described using arrow lines.

4.1 Optimization Result of FCDE-MRC

The FCDE-MRC model is applied to minimize project duration while satisfying both the precedence constraints and the resource constraints. Based on proposed values from the literature and several experiments (Cai *et al.,* 2011; Storn and Price, 1997; Wu, Wu *et al.*, 2010), this study set parameters for the FCDE optimizer in the case study as shown in Table 1. Fig. 9 shows the corresponding optimal solution schedule obtained using an FCDE-MRC based approach for construction project. In addition to sequences and start times or finish times of the all activities of projects in which dummy activities are not included, the schedule describes the resources allocation profiles of two types of resources (R_1, R_2) are reflectively described in the schedule.

4.2 Result Comparisons

To verify the performance of the proposed FCDE-MRC model, three different algorithms were used to compare performance. These algorithms include the original DE (Price *et al.,* 2005), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO)(Zhang *et al*., 2006c). For comparison purposes, all four algorithms used an equal number of function evaluations, had population sizes of 400, and used a maximum of 100 generations. In GA, the constant mutant and crossover probability factors were set at 0.5 and 0.9, respectively. In PSO, the two learning factors, c_1, c_2, were both chosen at 2.05, and the inertia factor w is set in

Table 1. Parameter Settings for the FCDE-MRC

Input parameters	Notation	
Number of decision variables	D	40
Population size	NP	10*D
Scaling Factor	F	0.5-0.8
Crossover probability	CR	0.8
Population to chaos ratio	CF	0.4-0.6
Period clustering	m	10
Number of centroids in clustering	G_{max}	$[2, \sqrt{NP}]$
Maximum generations		100

the range of 0.3-0.7. DE and FCDE control parameters remained the same, as stated previously in Table 1. To evaluate the stability and accuracy of each algorithm, optimization performance is expressed in terms of success rate, best result found (best), average result (avg.), standard deviation (std.), and worst result (worst) after 30 runs (see Table 2). Success rate equals to number of runs that the algorithm found optimal solution over all experimental runs. The best results and worst results demonstrate the capacity of the algorithm to find the optimal solution. Average and standard deviation are two additional characteristics used to describe solution quality. The standard deviation occurs in cases in which the algorithms are not able to generate optimal solutions in all executions. These criteria (best, worst, average, standard deviation) are calculated according to statistical theory after obtaining all results from experimental runs. In Fig. 10 shows the best project duration corresponding to the number of iterations of case study. Fig. 10 shows that the FCDE-MRC model found the optimal solution in fewer iterations than the original DE and other benchmark algorithms. The proposed FCDE is able to obtain the optimal solution better than DE only two more iterations. However, this comparison only is done on the best run of all algorithms.

Table 2 shows that the performance of the newly developed model in the case study was competitive in terms of both accuracy and stability. The FCDE-MRC found optimal solutions with highest success rate 60% in all executions of cases study compared to DE, the second best algorithm considered in the

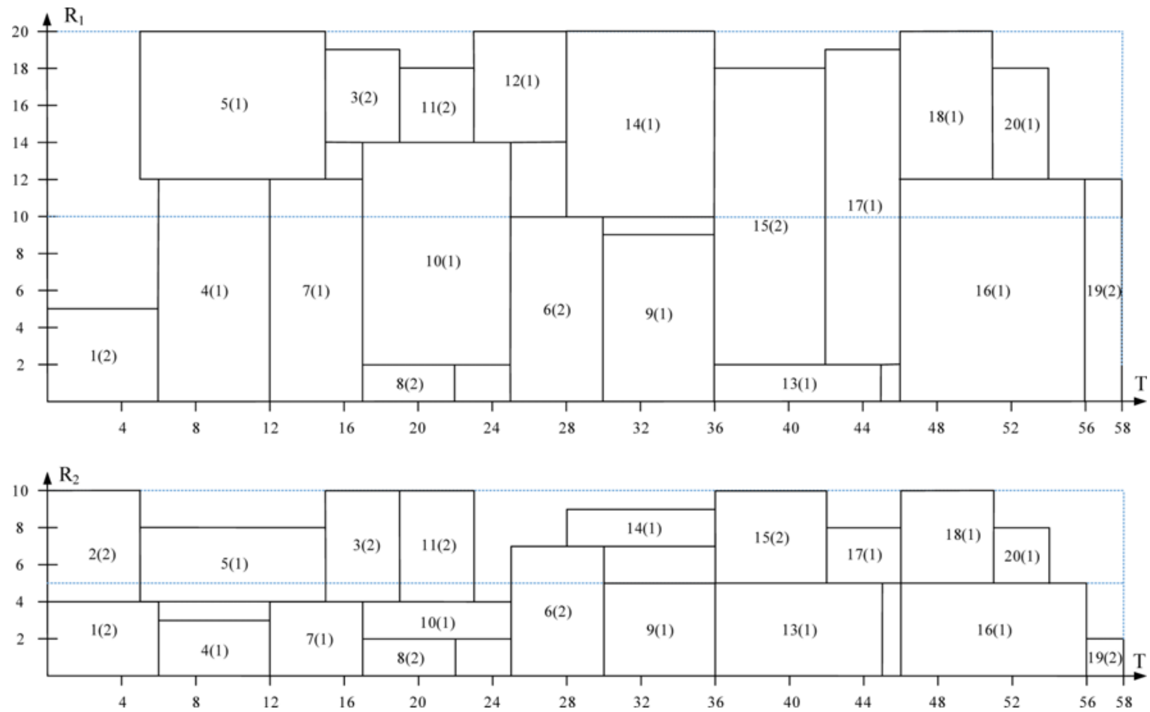


Fig. 9. Schedule with Execution Mode and Resource Allocation Profiles

Table 2. Result Comparison between FCDE-MRC and Benchmarked Algorithms

Performance Measurement			GA	PSO	DE	FCDE-MRC
Case study	Success rate (%)		10.00	16.67	33.33	60.00
	Optimal value	Best	58	58	58	58
		Avg.	60.133	60.033	59.967	58.933
		Std.	1.106	0.964	0.850	0.827
		Worst	63	62	61	60

Note: Avg. is average, std. is standard deviation

case study, with 33.33% in overall. Moreover, in terms of average results, FCDE-MRC performed the best, as it generated the lowest fitness solution with a value of 58.933 and a deviation value of 0.827. Therefore, the FCDE-MRC is a competitive model based on the relatively high accuracy of its obtained results and reliability in finding the optimal solution in significantly fewer computational times.

4.3 Sensitivity Analyses

Sensitivity analyses on some tuning parameters are conducted

Table 6. Sensititive Analyses Results

Measurement	Tuning parameters	Values	Success rate (%)	Best	Avg.	Std.	Worst
FCDE performance	Maximum generations (G_{max})	50	26.67	58	59.40	1.00	61
		100	60.00	58	58.93	0.83	60
		150	60.00	58	58.57	0.77	61
		200	60.00	58	58.57	0.77	60
	Population to chaos ratio (CF)	0.2	16.67	58	59.67	0.99	61
		0.4	30.00	58	59.03	0.81	60
		0.6	43.33	58	59.20	1.13	61
		0.8	36.67	58	59.03	0.89	60
		1.0	20.00	58	59.70	0.92	61
	Crossover Probability (CR)	0.2	33.33	58	59.03	0.85	60
		0.4	33.33	58	59.20	0.92	60
		0.6	20.00	58	59.60	1.13	61
		0.8	60.00	58	58.93	0.83	60
	Period clustering (m)	5	33.33	58	59.27	0.94	60
		10	60.00	58	58.93	0.83	60
		15	40.00	58	59.10	0.96	61
		20	36.67	58	59.07	0.91	61

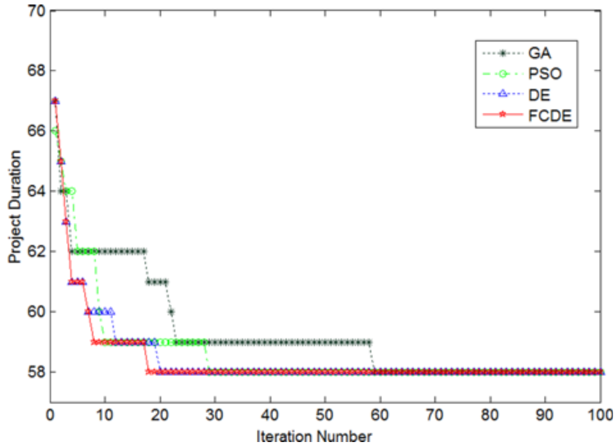


Fig. 10. Best Duration Curves for Project

for the proposed algorithm FCDE to clarify the variations of the results due to variations of tuning parameters. Table 3 shows the sensitivity analyses results on the tuning parameters.

4.3.1 Influence of Maximum Generation

In order to test the influence of maximum generation (stopping condition) on the performance of the proposed FCDE, a set of experimental run is conducted with changing the number of iterations (G_{max} = 50; 100; 150; 200) and all other parameters are fixed. For each G_{max}, we perform 30 independent run to obtain the results. As can be seen from the data in Table 3 that the higher number of iteration is, the stable and accuracy of the proposal model are. Nevertheless, there are only minor improvements in stable of model since the success rate is the same 60% for G_{max} = 100; 150; 200 and the average fitness values are slightly different on each values of G_{max}. Moreover, we need sacrifice the computational time double. Therefore, G_{max} = 100 is appropriate choice in this case study to balance the stable, accuracy and computational time of the proposal model.

4.3.2 Influence of Population to Chaos Ratio (CF)

In the chaos process, increasing the percentage of the population to chaos will increase the diversity of possible movements, promoting the exploration of the search space. Nevertheless, the probability to find the right search direction reduce significantly. The influence of population to chaos ratio (CF) is investigated in this subsection. All the parameters are the same as mentioned in the Table 1 only except for CF. The results for CF = 0.2; 0.4; 0.6; 0.8 and 1.0 are shown in Table 3. For CF = 0.6 and 0.8, FCDE is significantly better than the rest of other settings on success rate and the stable, respectively. In summary, according to the results in Table 3, we can conclude that FCDE can provide the best results using the ratio of population to chaos in range 0.6-0.8.

4.3.3 Effect of Crossover Probability (CR)

In DE, the choice of the control parameters scaling factor (F) and crossover probability (CR) is sensitive for different problem (Zielinski *et al.,* 2006). Price *et al*. (2005) suggested an initial value of 0.5 for F and an initial value of 0.8 for Cr, depending on specific problem characteristics. Different settings are suitable for different problems. We run experiments for each value of CR = 0.2; 0.4; 0.6; 0.8 to show the performance of FCDE. From Table 3, it can be seen that for the CR = 0.8, the proposed model yielded the best outcome compared to the other settings. It indicates that CR = 0.8 is the most suitable setting for FCDE in the case study.

4.3.4 Effect of Period Clustering (m)

In order to investigate the effect of period clustering m on the performance of FCDE, a set of experimental runs has been performed. All other parameters are kept unchanged as mentioned in the Table 1 and we only modify the period clustering parameter m as follows: m = 5, 10, 15, 20. For each m, we perform 30 independent run to obtain the results. As shown in Table 3 that a lower period clustering can achieve faster convergence, however this may lead to get stuck in local optimum. The higher period clustering makes the model more robust but lowers convergence. According to the experimental results in Table 3, the parameter m can be suggested in the interval [5, 10] for the case study. In this paper, m = 10 is the most appropriate selection.

4.3.5 Conclusions and Further Study

The FCDE based methodology for solving the multimode resource-constrained project-scheduling problem with the objective of minimizing project duration is proposed in this study. Integrating the fuzzy-clustering algorithm and chaos algorithm into the DE effectively eliminated the drawbacks of the original DE. The randomness of the chaos algorithm enhanced population diversity and avoided entanglement in the local optima. Further, the moving-cluster centers inherent to the fuzzy c-means clustering technique improved the convergence speed of the search algorithm. Moreover, this study used the serial method to incorporate individual-user priorities into the active schedule to calculate project duration. The proposed method is easy to understand, convenient to implement, and generates accurate results quickly. In addition, the comparisons with other methods demonstrate that the FCDE method is able to solve the MRC problem efficiently more than the existing methods.

The construction project was used to validate the comparative effectiveness of the FCDE-MRC model in handling resource-constrained problems. FCDE-MRC improved the performance of the original DE significantly more than the benchmark algorithms considered. The proposed model achieved the highest success rates of 60% with the smallest average optimal value and standard deviation value to obtain optimal solutions within the generation limit of 100. The proposed model thus provides stable, highly accurate results that project managers may use to make optimal decisions during construction project implementation.

The proposed FCDE is simple, robust, and efficient. Further minor modifications of the proposed FCDE algorithm hold interesting potential to resolve other classes of single-objective optimization problems not only in the construction management but also in construction industry related fields such as resource-allocation and resource leveling problems. Moreover, application of FCDE to a more complicated MRC problem with multiple objectives (e.g., leveling of resource usage and minimizing of project duration) and consideration of uncertain activity duration is under study at present. Extending the current model FCDE from a single-objective to multi-objective format using multiple objective differential evolution and fuzzy theory represents an interesting direction for future research.

The scheduling tool only considered deterministic and static input data and assumed that the logical relationship amongst activities is 'finish-to-start'. Therefore, further study is required to build an optimization model to stochastic data. Other possible dependencies in activity relationship including 'finish-to-finish', 'start-to-start', 'start-to-finish' could be involved in the proposed model. Integrated current model with other simulation techniques such as CYCLONE and Monte Carlo are interesting directions for future works. Further, regarding the activities that are not capable of exact quantification of input values, one can use methods such as Analytic Hierarchy Process (AHP), Analytic Network Process (ANP) and Fuzzy Preference Relations (FPR) to quantity the qualitative variables.

References

Alcaraz, J. and Maroto, C. (2001). "A robust genetic algorithm for resource allocation in project scheduling." *Annals of Operations Research*, Vol. 102, Nos. 1-4, pp. 83-109, DOI: 10.1023/A:1010949931021.

Alcaraz, J., Maroto, C., and Ruiz, R. (2003). "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms." *Journal of the Operational Research Society,* Vol. 54, No. 6, pp. 614-626, DOI: 10.1057/palgrave.jors.2601563.

Ozer, A. B. (2010). "CIDE: Chaotically initialized differential evolution." *Expert Systems with Applications,* Vol. 37, No. 6, pp. 4632-4641, DOI: 10.1016/j.eswa.2009.12.045.

Cai, Z., Gong, W., Ling, C. X., and Zhang, H. (2011). "A clustering-based differential evolution for global optimization." *Applied Soft Computing*, Vol. 11, No. 1, pp. 1363-1379, DOI: 10.1016/ j.asoc.2010. 04.008.

Cheng, M. Y., and Tran, D. H. (2014). "Two-phase differential evolution for the multiobjective optimization of time–cost tradeoffs in Resource-constrained construction projects." *Engineering Management, IEEE Transactions on,* Vol. 61, No. 3, pp. 450-461, DOI: 10.1109/TEM. 2014.2327512.

Cheng, M.-Y., Huang, K.-Y., and Chen, H.-M. (2012). "Dynamic guiding particle swarm optimization with embedded chaotic search for solving multidimensional problems." *Optimization Letters,* Vol. 6, No. 4, pp. 719-729, DOI: 10.1007/s11590-011-0297-z.

Damak, N., Jarboui, B., Siarry, P., and Loukil, T. (2009). "Differential evolution for solving multi-mode resource-constrained project scheduling problems." *Computers & Operations Research,* Vol. 36, No. 9, pp. 2653-2659, DOI: 10.1016/j.cor.2008.11.010.

Deb, K. (2005). "A population-based algorithm-generator for real-parameter optimization." *Soft Comput,* Vol. 9, No. 4, pp. 236-253, DOI: 10.1007/s00500-004-0377-4.

Elbeltagi, E., Hegazy, T., and Grierson, D. (2005). "Comparison among five evolutionary-based optimization algorithms." *Advanced Engineering Informatics,* Vol. 19, No. 1, pp. 43-53, DOI: 10.1016/j.aei.2005. 01.004.

Elloumi, S. and Fortemps, P. (2010). "A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem." *European Journal of Operational Research,* Vol. 205, No. 1, pp. 31-41, DOI: 10.1016/j.ejor.2009.12.014.

Jarboui, B., Damak, N., Siarry, P., and Rebai, A. (2008). "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems." *Applied Mathematics and Computation,* Vol. 195, No. 1, pp. 299-308, DOI: 10.1016/j.amc. 2007.04.096.

Jia, D., Zheng, G., and Khurram Khan, M. (2011). "An effective memetic differential evolution algorithm based on chaotic local search." *Information Sciences*, Vol. 181, No. 15, pp. 3175-3187, DOI: 10.1016/j.ins.2011.03.018.

Jiang, B. L. W. (1998). "Optimizing complex functions by chaos search." *Cybernetics and Systems*, Vol. 29, No. 4, pp. 409-419, DOI: 10.1080/019697298125678.

Kim, J. H. and Stringer, J. (1992). *Applied chaos*, *Electric Power Research Institute, Palo Alto, California*.

Kolisch, R. (1996). "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation." *European Journal of Operational Research,* Vol. 90, No. 2, pp. 320-333, DOI: 10.1016/0377-2217(95)00357-6.

Kwedlo, W. (2011). "A clustering method combining differential evolution with the K-means algorithm." *Pattern Recognition Letters,* Vol. 32, No. 12, pp. 1613-1621, DOI: 10.1016/j.patrec.2011.05.010.

Li, H. and Zhang, H. (2013). "Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints." *Automation in Construction,* Vol. 35, pp. 431-438, DOI: 10.1016/j.autcon.2013.05.030.

Liao, T., Warren, P. J., Egbelu, B. R., Sarker, and Leu, S. S. (2011). "Metaheuristics for project and construction management – A state-of-the-art review." *Automation in Construction,* Vo. 20, No. 5, pp. 491-505, DOI: 10.1016/j.autcon.2010.12.006.

Lova, A., Tormos, P., Cervantes, M., and Barber, F. (2009). "An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes." *International Journal of Production Economics*, Vol. 117, No. 2, pp. 302-316, DOI: 10.1016/ j.ijpe.2008.11.002.

Lu, M., Lam, H.-C., and Dai, F. (2008). "Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization." *Automation in Construction,* Vol. 17, No. 6, pp. 670-681, DOI: 10.1016/j.autcon.2007.11.004.

Noman, N. and Iba, H. (2008). "Accelerating differential evolution using an adaptive local search." *Evolutionary Computation, IEEE Transactions on,* Vol. 12, No. 1, pp. 107-125, DOI: 10.1109/TEVC. 2007.895272.

Ohya, M. (1998). "Complexities and their applications to characterization of chaos." *International Journal of Theoretical Physics,* Vol. 37, No. 1, pp. 495-505, DOI: 10.1023/A:1026620313483.

Peteghem, V. V. and Vanhoucke, M. (2010). "A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem." *European Journal of Operational Research,* Vol. 201, No. 2, pp. 409-418, DOI: 10.1016/j.ejor.2009. 03.034.

Price, K. V., Storn, R. M., and Lampinen, J. A. (2005). *Differential evolution a practical approach to global optimization*, Springer-Verlag Berlin Heidelberg 2005.

Rostami, M., Moradinezhad, D., and Soufipour, A. (2014). "Improved and competitive algorithms for large scale multiple resource-constrained project-scheduling problems." *KSCE Journal of Civil Engineering,* Vol. 18, No. 5, pp. 1261-1269, DOI: 10.1007/s12205-014-0401-x.

Sakalauskas, L. and Felinskas, G. (2006). "Optimization of resource constrained project schedules by genetic algorithm based on the job priority list." *Information Technology and Control,* Vol. 35, No. 4, 412-418, DOI: 10.1.1.103.7996.

Storn, R. M. and Price, K. (1997). "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces." *Journal of Global Optimization,* Vol. 11, pp. 341-359, DOI: 10.1023/A:1008202821328.

Wang, Y.-J., Zhang, J.-S., and Zhang, G.-Y. (2007). "A dynamic clustering based differential evolution algorithm for global optimization." *European Journal of Operational Research,* Vol. 183, No. 1, pp. 56-73, DOI: 10.1016/j.ejor.2006.10.053.

Wu, L., Wang, Y., and Zhou, S. (2010). "Improved differential evolution algorithm for resource-constrained project scheduling problem." *Systems Engineering and Electronics, Journal of,* Vol. 21, No. 5, pp. 798-805, DOI: 10.3969/j.issn.1004-4132.2010. 05.013.

Zhang, H. (2012). "Ant colony optimization for multimode resource-constrained project scheduling." *Journal of Management in Engineering,* Vol. 28, No. 2, pp. 150-159, DOI: 10.1061/(ASCE)ME.1943-5479.0000089.

Zhang, H., Tam, C. M., and Shi, J. J. (2002). "Simulation-based methodology for project scheduling." *Construction Management and Economics,* Vol. 20, No. 8, pp. 667-678, DOI: 10.1080/01446190220000014088.

Zhang, H., Li, H., and Tam, C. (2006a). "Particle swarm optimization for resource-constrained project scheduling." *International Journal of Project Management,* Vol. 24, No. 1, pp. 83-92, DOI: 10.1016/j.ijproman.2005.06.006.

Zhang, H., Li, H., and Tam, C. (2006b). "Permutation-based particle swarm optimization for resource-constrained project scheduling." *Journal of Computing in Civil Engineering*, Vol. 20, No. 2, pp. 141-49, DOI: 10.1061/(ASCE)0887-3801(2006)20:2(141).

Zhang, H., Tam, C. M., and Li, H. (2006c). "Multimode project scheduling based on particle swarm optimization." *Computer-Aided Civil and Infrastructure Engineering,* Vol. 21, No. 2, 93-103, DOI: 10.1111/j.1467-8667.2005.00420.x.

Zhang, Z. and Xu, J. (2014). "Applying rough random MODM model to resource-constrained project scheduling problem: A case study of Pubugou Hydropower Project in China." *KSCE Journal of Civil Engineering,* Vol. 18, No. 5, pp. 1279-1291, DOI: 10.1007/s12205-014-0426-1.

Zielinski, K., Weitkemper, P., Laur, R., and Kammeyer, K. D. (2006). "Parameter study for differential evolution using a power allocation problem including interference cancellation." In *Evolutionary Computation, 2006, CEC 2006, IEEE Congress on*., pp. 1857-1864, DOI: 10.1109/cec.2006.1688533.