IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 06, Issue 06 (June. 2016), ||V2|| PP 46-53

Hybrid Artificial Intelligence Based on Evolutionary Approach in Optimizing Multiple Resources of Projects

Thi Minh-Truc Huynh[1], Duc-Hoc Tran[1*], Thi Thao-Nguyen Nguyen[1], Quynh-Chau Truong[1]

[1]*(Faculty of Project Management, The University of Da Nang-University of Science and Technology, 54 Nguyen Luong Bang Rd, DaNang, VietNam)*
**Email:* tdhoc@dut.udn.vn

***Abstract:* -** Resource leveling problems are attractive and important part of construction project planning. Resource leveling is the process used within project scheduling to reduce fluctuations in resource usage over the period of project implementation. This study presents a new hybrid intelligence model, named as Artificial Bee Colony with Differential Evolution, to handle the multiple resources leveling in multiple projects problems (ABCDE-MRLMP). The proposed algorithm integrates crossover operation from differential evolution (DE) with original artificial bee colony (ABC) to balance exploration and exploitation phase of the optimization process. The ABCDE-MRLMP algorithm is compared with benchmark algorithms considered and previous findings using two construction case studies. The experimental results demonstrate the efficiency and effectiveness of the proposed model. The ABCDE-MRLMP is a promising alternative approach to handling resource leveling project scheduling problems.

***Keywords:* -** *Multiple resources levelling, Scheduling, Artificial Bee Colony, Optimization, Differential Evolution, Construction management.*

I. INTRODUCTION

The process of smoothing out resources, known as resource levelling, has been studied extensively [1-4]. Resource levelling attempts to minimize both the demand peak and the fluctuations in the pattern of resource usage [5] by optimizing noncritical activities within their available floats while keeping the project duration unchanged. Research on resource levelling has focused mainly on three aspects: (1) single-resource levelling in single-project scheduling [6], (2) multiple-resource levelling in single-project scheduling [7], and (3) single-resource levelling in multiple-project scheduling. However, multiple-resource levelling in multiple-project scheduling (MRLMP) is the most typical scenario in the construction and manufacturing industries, a situation that is relatively more complex and difficult to solve and that lacks a standard handling procedure [8, 9]. Thus, developing a more efficient optimization algorithm for MRLMP problems and to attain better resource levelling problem solutions are essential to improving the management of construction project resources.

The application of Evolutionary Algorithms (EAs) to resource levelling has attracted increasing attention in recent years [10, 11]. Based on the principles of natural evolution, EAs are stochastic optimization techniques that have successfully resolved optimization problems in diverse fields [12]. However, EAs suffer from certain weaknesses. Geng, Weng [10] identified premature convergence and poor exploitation as the main obstacles preventing EAs from coping effectively with complex optimization problems. Thus, developing a methodology for multiple-resource levelling in multiple-project problems and a more efficient algorithm to attain better resource levelling problem solutions are essential to improving the management of construction project resources.

The Differential Evolution (DE) [13] is currently one of the most popular evolutionary algorithms. DE may be used in a wide variety of highly nonlinear and complex optimization problems. This algorithm is simply structured and easy to use, while demonstrating great robustness and fast convergence in solving single-objective global optimization problems [13]. The ability of DE to provide efficient solutions for complex problems with relatively simple operations has encouraged many researchers to apply DE-based techniques [12, 14, 15]. The Artificial Bee Colony (ABC) algorithm is one of the most recently introduced evolutionary methods [16]. With its few parameters, ABC is simple to implement and relatively efficient and robust in comparison to other algorithms. It has been applied successful to solve complex multi-model optimization problems [17]. The ABC algorithm distributes information on food location throughout the entire population of bees. This characteristic makes ABC good at exploration but poor at exploitation and thus inadequate for problems that must apply existing information to find a better solution. Additionally, ABC converges relatively slowly for certain complex issues [18].

Although meta-heuristic methods have been proven to have superior features than other traditional methods, they also suffer some limitations. In addition, numerous researchers have found that a skilled combination of two meta-heuristics may be beneficial and perform significantly better than single pure meta-heuristic algorithm in handling real-world and large-scale problems [19-22]. Therefore, hybridization with other algorithms offers the potential to further improve the performances of ABC and DE. The superior performance of hybridized ABC and DE over other algorithms in engineering problems have been widely reported and verified [18]. An extensive review of the literature done for this study found that many reports of impressive hybridized ABC-DE performance in benchmark functions and practical applications, however, this algorithm has yet to be applied to solving the resource levelling problem. Therefore, this paper applies the hybridized ABC-DE algorithm in a model that is designed to solve the MRLMP problems problem.

The remainder of this paper is organized as follows: Section 2 describes the multiple resources levelling in the multiple projects problem. Section 3 provides a detailed description of the proposed algorithm for the resource levelling problem. Section 4 uses two numerical case studies to demonstrate model performance. The final section presents conclusions and suggests directions for future work.

II. PROBLEM FORMULATION

An enterprise consists of total of *n* projects. Each project includes multiple activities and each activity uses *p* resources. Symbols used in related formulas include: the set of activities in the project k is $\{(i_k, j_k)\} = \{A_k, \ldots, Z_k\}$; $R_{m(t)}$ is the demand for resource *m* by all *n* projects on day *t*; $R_{mt}(i_k, j_k)$ is the demand for resource m by activity (i_k, j_k) on day *t*; $R_m(i_k, j_k)$ is the demand for resource m by activity (i_k, j_k) on one day. $TE(i_k, j_k)$, $TL(i_k, j_k)$, $TS(i_k, j_k)$, $TF(i_k, j_k)$, $T(i_k, j_k)$, $S(i_k, j_k)$ represent early start time, late start time, actual start time, actual finish time, duration, and slack time of (i_k, j_k), respectively. The precedence set of activity (i_k, j_k) is $\{(pset_k, i_k)\}$. Multiple-resource levelling in multiple-project scheduling differs from conventional resource levelling techniques primarily as follows [8]:

Firstly, due to differing levels of resource demand, assimilation must transform absolute demand into relative demand in order to enable all the p resources to be comparable in terms of quantity. The relative demand of resource m in all n projects on day t may be expressed as:

$$SR_m(t) = \lambda R_m(t) / R_{\max_m} \tag{1}$$

where $Rmax_m = \max\{R_m(t)\}$ denotes the maximum demand for resource m in total n projects on one day and λ is an amplifying coefficient within [1,100] used to increase simulation accuracy. The above formula limits the relative demand for each resource in a total of n projects on every single day to between 0 and λ.

Secondly, the weight score w_m measures the degree of importance for each resource. This paper uses the analytical hierarchy process (AHP) to obtain the weights of different resources. Larger weight scores correlate to greater priority.

The mathematical formulation of the objective function for multiple-resource levelling in multiple-projects scheduling is:

$$Min\ RI = \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{p}\left[w_m (SR_m(t) - \overline{SR_m})^2 \right] \tag{2}$$

subject to:

$$T_E(i_k, j_k) \le T_s(i_k, j_k) \le T_L(i_k, j_k) \tag{3}$$

$$\max\{T_s(pset_k, i_k) + T_s(pset_k, i_k)\} \le T_s(i_k, j_k) \le T_L(i_k, j_k) \tag{4}$$

$$R_m(t) = \sum_{k=1}^{n}\sum_{i_k, j_k} R_{mt}(i_k, j_k);\ \ \overline{SR_m} = \frac{1}{T}\sum_{t=1}^{T} SR_m(t) \tag{5}$$

$$R_{mt}(i_k, j_k) = \begin{cases} R_m(i_k, j_k)\ if: T_s(i_k, j_k) < t \le T_f(i_k, j_k) \\ 0\ if: t \le T_s(i_k, j_k)\ or\ t > T_f(i_k, j_k) \end{cases} \tag{6}$$

$$S(i_k, j_k) = T_L(i_k, j_k) - T_E(i_k, j_k) \tag{7}$$

where *T* is equal to the difference between the maximum of the latest finish time and the minimum of the earliest start time for all n projects.

III. THE HYBRID ARTIFICIAL BEE COLONY WITH DIFFERENTIAL EVOLUTION FOR MULTIPLE RESOURCES LEVELING

This section describes the newly proposed ABCDE optimization algorithm in detail. The ABCDE is the core optimizer in the ABCDE-MRLMP model. Fig. 1 shows the overall operational architecture of the proposed algorithm.

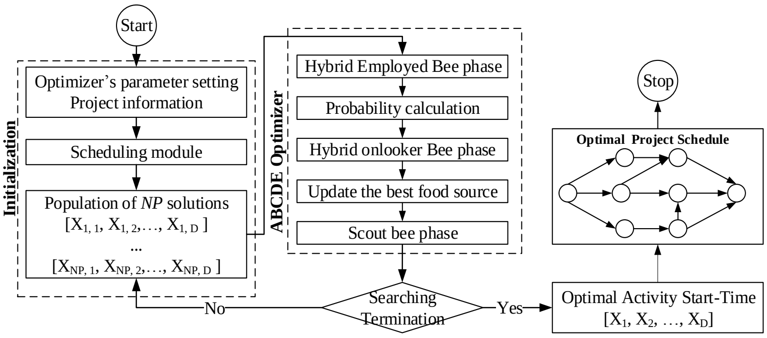


Fig. 1 Flowchart for the ABCDE-MRLMP

Inputs required by the ABCDE-MRLMP optimization model include activity relationship, activity duration, and resource demand. In addition, the user must provide search engine parameter settings such as maximum number of search generations G_{max} and population size (*NP*). The scheduling procedure uses these inputs in the calculation process to obtain the project duration and resource amount required for each activity. The model operates automatically.

Before starting the search process, a uniform random generator creates an initial population of feasible solutions. A solution for the resource levelling problem is represented as a vector with D elements as follows:

$$X = [X_{i,1}, X_{i,2}, \ldots, X_{i,D}] \tag{8}$$

where *D* is the number of decision variables in the problem at hand. *D* is also the number of non-critical activities in the project network. The index i denotes the ith individual in the population. The vector *X* represents the start time of *D* non-critical activities in the network. Because the original DE operates with real-value variables, a function is employed to convert the start times of those activities from real values to integer values that are constrained within the feasible domain.

$$X_{i,j} = Round(LB(j) + rand[0,1] \times (UB(j) - LB(j))) \tag{9}$$

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 the early start time and late start time for activity *j*. In multiple resources levelling in the multiple projects scheduling problem, two constraint conditions limit the actual start time of all activities: (1) actual start time must be between the early and late start times and (2) actual start time is limited by the actual start time of its predecessor activities. The first constraint is simple to handle because limits are fixed prior to calculation. However, the minimum limit of the second constraint is unknown prior to calculation and thus more difficult to elicit. For the decision variables of ABCDE on each dimension is determined in turn, when calculating the actual start time of one activity, actual start time of all activities in its predecessor set $TS(pset_k, i_k)$ have been computed, the max{$TS(pset_k, i_k)$+ $T(pset_k, i_k)$} has been confirmed simultaneously.

3.1 Hybrid employed bee phase

The DE crossover-mutation operators mutates the population to produce a set of mutant vectors. A mutated vector V_i^{G+1} is created using equation $V_i^{G+1} = X_{r1}^G + F(X_{r2}^G + X_{r3}^G)$ that corresponds to the target vector V_i^G. The crossover operation exchanges components of the target vector and the mutant vector to diversify the current population. In this stage, a new vector called the trial vector, is created using Eq. (10).

$$u_{i,j}^{G+1} = \begin{cases} v_{i,j}^{G+1} & if\ (rand_j[0,1) \le CR\ or\ j = j_{rand} \\ x_{i,j}^{G} & otherwise \end{cases} \quad j = 1, 2, \ldots, D \tag{10}$$

3.2 Probability calculation

Employed bees return to their hive and share food source information with onlooker bees. The information sharing stage of the ABCDE algorithm generates collective intelligence. The probability value influences the behaviour of onlooker bees, which select food sources based on probability. The probability value is calculated as:

$$P_i = 0.9 * Fit(X_i) / \max(Fit(X_i)) + 0.1 \tag{11}$$

where Fit(Xi) is the fitness value of the i^{th} solution (food source).

3.3 Hybrid onlooker bee phase

An onlooker bee selects a food source depending on the probability value P_i associated with that food source. ABCDE produces an onlooker bee by using Eq. (10). After evaluating the nectar amount of possible new position, the greedy selection is applied and the onlooker bee either updates the new position by removing or retaining the old solution.

3.4 Update the best food source position

The best food source position is updated after termination of the onlooker bee phase. A new best food source position replaces the old if the former provides an equal or better amount of nectar. Otherwise, the old remains valid.

3.5 Scout bee phase

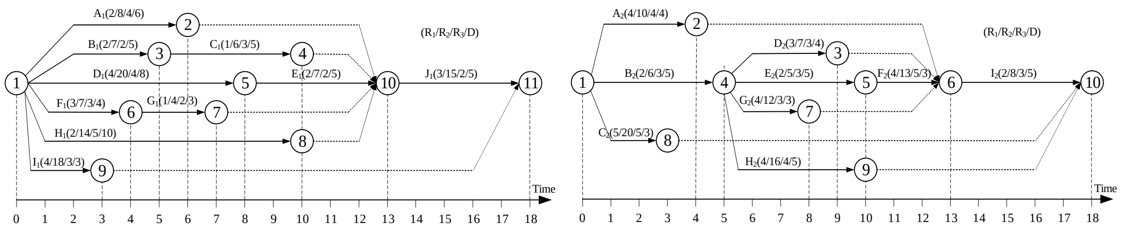
If food source $X_{i,j}$ (solution $X_{i,j}$) shows no improvement further through a continuous pre-determined number of cycles, then the food source abandoned by its bee is replaced with a new food source discovered by the scout bee, Eq. (9).

3.6 Stopping condition

The optimization process terminates when the user-designated stopping criterion is met. Termination of the optimization process presents the final optimal solution to the user.

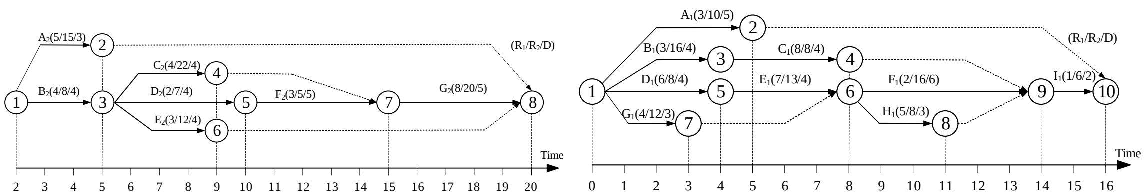
IV. CASE STUDIES

Two case studies adapted from Guo, Li [8] was used to demonstrate the capability of the newly developed ABCDE-MRLMP model. The first case consists of two projects with same duration needing to be started simultaneously in an enterprise. Each activity in both projects uses three resources (R1 human, R2 fund, R3 equipment), and the importance of each resource is set as w_1=0.637, w_2=0.258, w_3=0.105, respectively. The second case includes two projects with different durations and starting day. Every activity in example 2 uses two resources (R1 human, R2 fund), and the importance of R1 is the same as the importance of R2, w_1=w_2=0.5. In both case studies, each activity has a certain duration D that is indicated above the arrow line. Fig. 2 and Fig. 3 show the precedence relationships of the network projects in both cases, respectively.



a) Network of Project 1, Case 1 b) Network of Project 2, Case 1

Fig. 2 Networks of two projects-Case 1



b) Network of Project 2, Case 2 a) Network of Project 1, Case 2

Fig. 1 Networks of two projects-Case 2

Consequently, the objective functions for both case studies are calculated as follows:

Case1:

$$\min RI = \frac{1}{18}\sum_{t=1}^{18}\begin{bmatrix} 0.637(SR_1(t)-\overline{SR_1(t)})^2 \\ +0.258(SR_2(t)-\overline{SR_2(t)})^2+0.105(SR_3(t)-\overline{SR_3(t)})^2 \end{bmatrix} \text{s.t} \begin{cases} 0 \le T_s(A_1) \le 7 & 0 \le T_s(F_1) \le 6 \\ 0 \le T_s(B_1) \le 3 & T_s(B_1)+4 \le T_s(G_1) \le 10 \\ T_s(B_1)+5 \le T_s(C_1) \le 8 & 0 \le T_s(H_1) \le 3 \\ 0 \le T_s(I_1) \le 15 & 5 \le T_s(D_2) \le 9 \\ 0 \le T_s(A_2) \le 9 & 5 \le T_s(G_2) \le 7 \\ 0 \le T_s(C_2) \le 15 & 5 \le T_s(H_2) \le 13 \end{cases}$$

Case 2:

$$\min RI = \frac{1}{20}\sum_{t=1}^{20}\left[0.5(SR_1(t)-\overline{SR_1(t)})^2+0.5(SR_2(t)-\overline{SR_2(t)})^2\right] \text{s.t} \begin{cases} 0 \le T_s(A_1) \le 11 & T_s(B_1)+4 \le T_s(C_1) \le 10 \\ 0 \le T_s(B_1) \le 6 & 0 \le T_s(G_1) \le 5 \\ 8 \le T_s(H_1) \le 13 & 5 \le T_s(C_2) \le 11 \\ 2 \le T_s(A_2) \le 17 & 5 \le T_s(E_2) \le 16 \end{cases}$$

4.1 Optimization result of ABCDE-MRLMP

Application of the ABCDE -MRLMP model significantly reduces fluctuation in resource use. This study used parameters for the ABCDE optimizer based on proposed values from the literature and several experiments as shown in Table 1. Fig. 4 shows the network resource profile for the projects at initialization and after levelling using ABCDE-MRLMP optimization.

Table 1 Parameter settings for ABCDE-MRLMP

Input parameters	Notation	Setting	
		Case 1	Case 2
No of decision variables	*D*	12	8
Population size	*NP*	150	100
limit	*l*	15	15
Crossover probability	*CR*	0.5~0.9	0.5~0.9
Scaling factor	*F*	0.5	0.5
Amplification coefficient	λ	30	30
Maximum generation	G_{max}	200	150

Table 2 and Table 3 lists the optimal results, i.e. optimal non-critical-activity start times obtained from proposed model and other benchmark algorithms on case 1 and case 2, respectively. RI_m in Table 2 and Table 3 is the resource intensity for single resource m.

In the first case, the optimal resource intensity (RI) obtained by ABCDE-MRLMP was 94.9%, 2.0%, 2.0%, 6.9%, 14.4% less than the initial schedule, ABC, DE, PSO, and GA, respectively as shown in Table 2. In the second case, the RI obtained by proposed model was 84.3% better than the initial schedule.

To evaluate the stability and accuracy of each algorithm, optimization performance was expressed in terms of best result found (best), average result (avg), standard deviation (std), and worst result (worst) after 30 runs (Table 4). The best and worst results demonstrate the capacity of each algorithm to find the optimal solution for all of the performance measurement metrics. Average and standard deviation are two additional characteristics that describe solution quality. Standard deviation occurs in cases when algorithms are not able to generate optimal solutions in all executions.

As shown in Table 4, the performance of the ABCDE-MRLMP is competitive in terms of accuracy and stability. It is clearly shown that the proposed model is able to find optimal solutions in fitness function. Furthermore, in terms of average results, ABCDE-MRLMP performed the best of the considered algorithms in both case studies because it generated the lowest average fitness solution with a value of 4.715 and a deviation value of 0.206 in case 1 and obtained the lowest fitness solution with a value of 11.451 and the smallest deviation value of 0.461 in case 2.

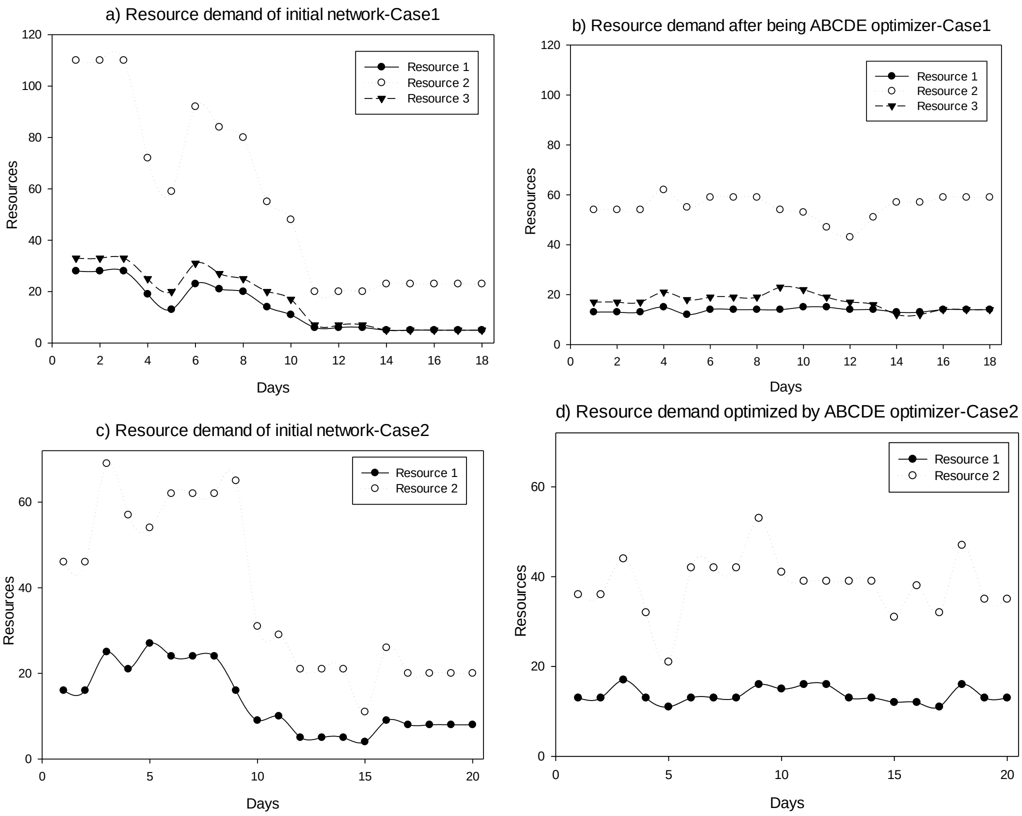


Fig. 4 Resource profiles before and after levelling

Table 2 Comparison of best performance for algorithms on case 1

Items	RI	RI_1	RI_2	RI_3	Actual start time of non-critical activities											
					A_1	B_1	C_1	F_1	G_1	H_1	I_1	A_2	C_2	D_2	G_2	H_2
Initial	89.46	76.95	1169	123.8	0	0	5	0	4	0	0	0	0	5	5	5
GA	5.327	1.06	13.76	9.17	0	3	8	0	9	0	15	8	12	9	6	13
PSO	4.897	0.84	26.65	7.95	0	0	8	6	10	3	12	0	15	8	5	13
DE	4.652	0.84	21.97	5.73	0	3	9	0	10	0	12	8	15	9	6	13
ABC	4.652	0.84	21.97	5.73	0	3	9	0	10	0	12	8	15	9	6	13
ABCDE-MRLMP	**4.558**	**0.62**	**21.31**	**9.51**	3	0	8	0	8	0	12	8	15	9	5	13

Note that RI is resource intensity.

Table 3 Comparison of best performance for algorithms on case 2

Items	RI	RI_1	RI_2	Actual start time of non-critical activities							
				A_1	B_1	C_1	G_1	H_1	A_2	C_2	E_2
Initial	71.70	60.04	366.43	0	0	5	0	4	0	0	0
GA, PSO, DE, ABC, ABCDE-MRLMP	**11.29**	**2.94**	**41.93**	9	0	8	0	12	17	5	14

Table 4 Comparison of results for the ABCDE-MRLMP and benchmarked algorithms

Performance Measurement			GA	PSO	ABC	DE	ABCDE-MRLMP
Fitness value	Case 1	Best	5.327	4.897	4.652	4.652	4.558
		Avg.	6.832	6.154	5.074	5.346	4.715
		Std.	1.979	0.864	0.341	0.501	0.206
		Worst	13.385	10.038	6.133	6.506	5.364
	Case 2	Best	11.295	11.295	11.295	11.295	11.295
		Avg.	14.009	13.613	12.074	12.279	11.451
		Std.	2.074	1.496	0.801	0.998	0.461
		Worst	18.880	17.060	13.557	14.917	13.557

Note that Avg. is average, Std. is standard deviation.

V. CONCLUSION

This paper uses ABCDE to solve the problem of multiple-resource levelling in the context of multiple-projects scheduling. The proposed algorithm integrates crossover operations from differential evolution (DE) with the original artificial bee colony (ABC) in order to balance the exploration and exploitation phases of the optimization process. Hybrid algorithm ABCDE has better global search ability and local search ability than original ABC and DE algorithms. Two application cases are analyzed to illustrate the effectiveness of the proposed model and to demonstrate the capabilities of the model in generating an optimal schedule that eliminates undesirable resource fluctuations and resource idle times. Experimental results and a comparison of results indicate that the ABCDE-MRLMP effectively improves the performance of the original ABC and DE beyond the levels of performance attained by other benchmark algorithms.

The ABCDE 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 problems.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the University of Science and Technology-The University of Da Nang (DUT) for financially supporting this work. Appreciation is also expressed to Faculty of Project Management's research group at DUT for assistance provided during this investigation. Finally, the authors would like to thank anonymous referees for their valuable comments.

REFERENCES

[1] Savin, D., S. Alkass, and P. Fazio, Construction resouce leveling based using neural networks. Can. J. Civ. Engrg., 23(3), 917-925, 1996.

[2] Son, J. and M.J. Skibniewski, Multiheuristic Approach for Resource Leveling Problem in Construction Engineering: Hybrid Approach. Vol. 125. 1999: ASCE. 23-31.

[3] Doulabi, S.H.H., A. Seifi, and S.Y. Shariat, Efficient Hybrid Genetic Algorithm for Resource Leveling via Activity Splitting. Vol. 137. 2011: ASCE. 137-146.

[4] Gather, T., J. Zimmermann, and J.-H. Bartels, Exact methods for the resource levelling problem. Journal of Scheduling, 2011. 14(6): p. 557-569.

[5] Yan, L., et al. Optimization of resource allocation in construction using genetic algorithms. in Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on. 2005.

[6] Damci, A. and G. Polat, Impacts of different objective functions on resource leveling in construction projects: a case study. Journal of Civil Engineering and Management, 2014. 20(4): p. 1-11.

[7] Ponz-Tienda, J.L., et al., The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm. Automation in Construction, 2013. 29(0): p. 161-172.

[8] Guo, Y., N. Li, and T. Ye. Multiple Resources Leveling in Multiple Projects Scheduling Problem Using Particle Swarm Optimization. in Natural Computation, 2009. ICNC '09. Fifth International Conference on. 2009.

[9] Woodworth, B.M. and C.J. Willie, A Heuristic Algorithm For Resource Leveling in Multi-Project, Multi-Resource Scheduling. Decision Sciences, 1975. 6(3): p. 525-540.
[10] Geng, J.-q., L.-p. Weng, and S.-h. Liu, An improved ant colony optimization algorithm for nonlinear resource-leveling problems. Computers & Mathematics with Applications, 2011. 61(8): p. 2300-2305.
[11] Leu, S.-S., C.-H. Yang, and J.-C. Huang, Resource leveling in construction by genetic algorithm-based optimization and its decision support system application. Automation in Construction, 2000. 10(1): p. 27-41.
[12] Das, S. and P.N. Suganthan, Differential Evolution: A Survey of the State-of-the-Art. IEEE Transactions on Evolutionary Computation, 2011. 15(1): p. 4-31.
[13] Storn, R.M. and K. Price, Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, 1997. 11: p. 341–359.
[14] Mohamed, A.W., RDEL: Restart Differential Evolution algorithm with Local Search Mutation for global numerical optimization. Egyptian Informatics Journal, 2014. 15(3): p. 175-188.
[15] Mohamed, A.W., H.Z. Sabry, and T. Abd-Elaziz, Real parameter optimization by an effective differential evolution algorithm. Egyptian Informatics Journal, 2013. 14(1): p. 37-53.
[16] Karaboga, D. and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 2007. 39(3): p. 459-471.
[17] Karaboga, D. and B. Akay, A comparative study of Artificial Bee Colony algorithm. Applied Mathematics and Computation, 2009. 214(1): p. 108-132.
[18] Li, Y., Y. Wang, and B. Li, A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow. International Journal of Electrical Power & Energy Systems, 2013. 52(0): p. 25-33.
[19] Li, X. and M. Yin, Hybrid differential evolution with artificial bee colony and its application for design of a reconfigurable antenna array with discrete phase shifters. Microwaves, Antennas & Propagation, IET, 2012. 6(14): p. 1573-1582.
[20] Bettemir, Ö. and R. Sonmez, Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling. Journal of Management in Engineering, 2014: p. 04014082.
[21] Mohammadi, H.R. and A. Akhavan, Parameter Estimation of Three-Phase Induction Motor Using Hybrid of Genetic Algorithm and Particle Swarm Optimization. Journal of Engineering, 2014. 2014: p. 6.
[22] J.Jithendranath and K.H. Reddy, Differential Evolution approach to Optimal Reactive Power Dispatch with Voltage Stability enhancement by Modal Analysis. International Journal of Engineering Research and Applications, 2013. 3(4): p. 66-70.
[23] Price, K.V., R.M. Storn, and J.A. Lampinen, Differential Evolution A Practical Approach to Global Optimization. 2005: Springer-Verlag Berlin Heidelberg.
[24] Karaboga, D. and B. Basturk, On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing, 2008. 8(1): p. 687-697.
[25] Haupt, R.L. and S.E. Haupt, Practical Genetic Algorithms. John Wiley & Sons, Inc. NJ., 2004.
[26] Clerc, M., Particle Swarm Optimization. ISTE Ltd, London., 2006.