Research for HCODEQ Method

Ji-Pyng Chiou[1*], Chung-Fu Chang[2], and Yung-Lin Wang[3]
Department of Electrical Engineering, Ming Chi University of Technology, Taiwan[1,3]
Department of Electrical Engineering, WuFeng University, Taiwan[2]
jipyng@mail.mcut.edu.tw[1], cfchang@mail.wfc.edu.tw[2], n36956789@gmail.com[3]
**Corresponding Author*

Received 16 Aug. 2016; received in revised form 28 Oct. 2016; accepted 9 Nov. 2016

Abstract

This paper presents a comparison of the convergence properties between the HCODEQ, CODEQ, and differential evolution (DE) methods. The concepts of chaotic search, opposition-based learning, and quantum mechanics are used in the CODEQ method to overcome the drawback of selecting the crossover factor and scaling factor used in the DE method. However, a larger population size must be used in the CODEQ method. That is a drawback for all evolutionary algorithms (EAs). To overcome this drawback, acceleration operation and migrating operation are embedded into the CODEQ method, i.e. HCODEQ method. The migrating operation can be used to maintain the population diversity, which guarantees a high probability of obtaining the global optimum. And the aim of the accelerated operation is to speed up the convergence. However, this faster convergence also leads to a higher probability of obtaining a local optimum because the diversity of the population descends faster during the solution process. So, these two operations can act as a trade-off operation for the population diversity and convergence to accelerate the search of the global solution. To prove the convergence property of the HCODEQ method, four benchmark functions from the literature are used to compare the performance of the HCODEQ, CODEQ, and DE methods. Numerical results show that the HCODEQ method outperformed other methods.

Keywords: HCODEQ, CODEQ, DE, migrating operation, accelerated operation

1. Introduction

CODEQ (Omran & Salman, 2009; Omran & Salman, 2009; Omran & Salman, 2010; Omran, 2010) is a population-based, parameter-free meta-heuristic algorithm integrating concepts from chaotic search, opposition-based learning, Differential Evolution (DE) and quantum mechanics. DE as developed by Stron and Price (1996). It has proved to be a promising candidate in solving real-valued optimization problems (Amjad, Salam & Saif, 2015; Zamuda & Brest, 2014; Havangi, Nekoui, Teshnehlab & Taghirad, 2014; Reddy & Sahoo, 2014; Liang, Qu, Mao, Niu & Wang, 2014; Chiou & Chang, 2010; Chiou & Chang, 2009; Price, 1997). DE is a stochastic search and optimization method. The fittest in an offspring competes one-on-one with that of the corresponding parent, which is different from the other EAs. This type of competition will lead to a faster convergence rate. However, this faster convergence also leads to a higher probability of obtaining a local optimum because the diversity of the population descends faster during the solution process. To maintain the diversity of the population, a larger population size must be used like the other evolutionary algorithms (EAs) use. So the selection of parameters is very important for the DE method because some parameters are more sensitive to the problem. For example, a fixed scaling factor is used in DE. Using a smaller scaling factor, DE becomes increasingly robust. However, much computational time should be spent to evaluate the objective function. DE with a larger scaling factor result generally falls

into local solution or non-convergence. Two parameters including the scaling factor and mutation operator are more difficult to set in DE. So, the concept of quantum mechanics is needed in CODEQ to overcome these two parameters selection problem. At the same time, the concepts of opposition-based learning and the chaotic search can be combined as an excluded operation used to speed up the convergence. The basic concept of opposition-based learning is the consideration of an estimate and its corresponding opposite estimate simultaneously to approximate the current candidate solution (Omran & Salman, 2009). And chaotic sequences can be used to test the searching ability of heuristic optimization method (Omran & Salman, 2009). Due to the need to execute the crossover operation, DE is not rotationally invariant (Omran & Salman, 2009). To avoid the problem, the crossover operation of DE was removed in CODEQ. However, a larger population size is still used in CODEQ method. That's a drawback for all evolutionary algorithms (EAs).

To overcome the problem associated with a larger population size used in CODEQ algorithm, two operations including acceleration operation and migrating operation are embedded into original CODEQ method called HCODEQ method. The use of these two operations act as a trade-off operator which can increase the convergence speed without decreasing the diversity among individuals. Migrating operation maintains the diversity of population, which guarantees a high probability of obtaining the global optimum. And the accelerated operation is used to accelerate the convergence. To illustrate the convergence property of the proposed HCODEQ method, four benchmark functions from the literature are used to compare the performance of the proposed method with the HCODEQ, CODEQ, and DE methods in this study. From the computation results, it is observed that the convergence property of the HCODEQ method is better than the other methods.

2. HCODEQ Method

The main idea of the HCODEQ method is to use two operations, migrating operation and acceleration operation, to act as a trade-off operator to overcome the drawback associated with the use of a larger population size in the CODEQ method. The use of the acceleration operation can speed up the convergence of the HCODEQ. And the population diversity can be maintained by the migrating operation. The process of the HCODEQ method is briefly described in the following.

Step 1. Initialization

Input system data and generate the initial population. The initial population is chosen randomly and would attempt to cover the entire parameter space uniformly. The uniform probability distribution for all random variables as following is assumed as:

$$Z_i^0 = Z_{min} + \sigma_i \cdot (Z_{max} - Z_{min}), i = 1, \dots, N_p \quad (1)$$

where $\sigma_i \in (0, 1]$ is a random number. The initial process can produce N_p individuals of Z_i^0randomly.

Step 2. Mutation operation

The essential ingredient in the mutation operation is the difference vector. Different from the DE algorithm, the concept of the quantum mechanics (Omran & Salman, 2009; Omran & Salman, 2009; Omran & Salman, 2010; Omran, 2010) is used to generate the noise replica from the individual parent in HCODEQ algorithm which is expressed as follows:

$$\hat{Z}_i^{G+1} = Z_i^G + (Z_{i1}^G - Z_{i2}^G) \cdot \ln(1/u), i = 1, \dots, N_p, i1 \neq i2 \neq i \quad (2)$$

where $\mathrm{u} \in (0, 1]$ is a random number.

Step 3. Estimation and selection

$$Z_i^{G+1} = argmin\{f(Z_i^G), f(\hat{Z}_i^{G+1})\} \quad (3)$$

$$Z_b^{G+1} = argmin\{f(Z_i^G)\} \quad (4)$$

where $\arg\min$ means the argument of the minimum.

Step 4. Exclude operation if necessary

To increase the convergence of the HCODEQ algorithm, the excluded operation is considered. First, a new individual is created as follows:

$$Z_w^{G+1} = \begin{cases} Z_{min} + Z_{max} - \gamma \cdot Z_{worst}^{G+1}, & if\ \delta \le 0.5 \\ Z_{best}^{G+1} + \left| Z_{i1}^{G+1} - Z_{i2}^{G+1} \right| \cdot (2 \cdot c^{G+1} - 1), & otherwise \end{cases} \quad (5)$$

where γ and δ are randomly generated numbers uniformly distributed in the range of (0,1). Z_{worst}^{G+1} and Z_{best}^{G+1} are the worst and best individuals in the (G+1)th generation. c^{G+1} is the chaotic variable defined as follow:

$$c^{G+1} = \begin{cases} c^G/p, & if\ c^G \in (0,p) \\ (1-c^G)/(1-p), & if\ c^G \in [p,1) \end{cases} \quad (6)$$

where c^0 and p are initialized randomly within the interval (0,1).

The worst individual in the G-th generation is replaced by the generated individual if the fitness of the generated individual is better than that of the worst individual in the G-th generation.

Step 5. Migrating operation if necessary

In order to effectively enhance the investigation of the search space and reduce the choice pressure of a small population, a migrating operation is introduced to regenerate a new diverse population of individuals. The new population is yielded based on the best individual Z_b^{G+1}. The g-th gene of the i-th individual is as follows:

$$Z_{ig}^{G+1} = \begin{cases} Z_{bg}^{G+1} + \mu_i \cdot (Z_{g\,min} - Z_{bg}^{G+1}), if\ \beta < (Z_{bg}^{G+1} - Z_{g\,min})/(Z_{g\,max} - Z_{g\,min}) \\ Z_{bg}^{G+1} + \mu_i \cdot (Z_{g\,max} - Z_{bg}^{G+1}), otherwise \end{cases} \quad (7)$$

where μ_i and β are randomly generated numbers uniformly distributed in the range of $[0,1]$; $i = 1, \ldots N_p$; and $g = 1, \ldots, n$.

The migrating operation is executed only if a measure fails to match the desired tolerance of population diversity. The measure is defined as follows:

$$\varepsilon = \sum_{\substack{i=1 \\ i \neq b}}^{N_p} \sum_{g=1}^{n} \eta_Z / \left(n \cdot (N_p - 1) \right) < \varepsilon_1 \quad (8)$$

where

$$\eta_Z = \begin{cases} 0, if\ \varepsilon_2 < \left| \frac{Z_{gi}^{G+1} - Z_{bi}^{G+1}}{Z_{bi}^{G+1}} \right| \\ 1, otherwise \end{cases} \quad (9)$$

Parameters $\varepsilon_1, \varepsilon_2 \in [0,1]$ express the desired tolerance for the population diversity and the gene diversity with respect to the best individual. η_Z is the scale index. From (8) and (9), it can be seen that the value ε is in the range of $[0,1]$. If ε is smaller than ε_1, then the migrating operation is executed to generate a new population to escape the local point; otherwise, the migrating operation is turned off.

Step 6. Acceleration operation if necessary

When the best individual in the present generation cannot be improved any longer by the mutation operation, a decent method is then employed to push the present best individual towards attaining a better point. The accelerated phase is expressed as follows:

$$Z_b^{G+1} = \begin{cases} Z_b^{G+1}, if\ J(Z_b^{G+1}) < J(Z_b^G) \\ Z_b^{G+1} - \alpha \nabla J, otherwise \end{cases} \quad (10)$$

where Z_b^G denotes the best individual as obtained from equation (4). The gradient of the objective function, ∇J, can be approximately calculated by finite difference. The step size $\alpha \in (0,1]$ in (10) is determined by the descent property. Initially, α is set to one to obtain the new individual.

Step 7. Repeat step 2 to 6 until the terminal conditions are achieved.

The computational process of the HCODEQ is stated using a flowchart as shown in Figure 1.

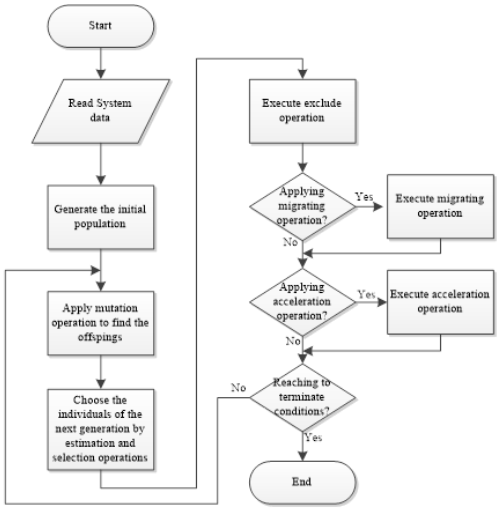

Figure 1: Main Calculation Procedures of the HCODEQ Method

3. Examples

The standard benchmark functions from the literature are frequently used to achieve the testing for reliability, efficiency and validation of optimization algorithms. To validate and compare the performance of optimization algorithms, the benchmark functions should have diverse properties, including modality, separability, and valley landscape so that they can be truly useful to test new algorithms in an unbiased way (Jamil & Yang, 2013; Molga & Smutnicki, 2005). Four benchmark functions are investigated and the computation results are used to compare the performance of the HCODEQ method with that of the CODEQ and DE methods.

Example 1: Let us consider the minimization problem which is described by:

$$\min_{Z_1,Z_2} J(Z_1,Z_2) = 100(Z_1^2 - Z_2)^2 + (1 - Z_1)^2 \quad (11)$$

where $-2.048 \leq Z_1 \leq 2.048$ and $-2.048 \leq Z_2 \leq 2.048$.

The first benchmark function is a Rosenbrock's Valley Function. This is a continuous, differentiable, non-separable, scalable, and unimodal function. Rosenbrock's valley is a classic optimization problem, also known as the banana function or the second function of De Jong. The global optimum lies inside a long, narrow, parabolic shaped flat valley. To find if the valley is trivial, however, convergence to the global optimum is difficult and hence, this problem has been frequently used to test the performance of optimization algorithms. This function has a global minimum value of 0 at $(Z_1,Z_2) = (1,1)$. To verify the performance of the HCODEQ method, the convergence property of the HCODEQ method, CODEQ method and DE method are compared via this example. The setting-factors were used in the HCODEQ method to solve this example. The population size is set to 5. The maximum generation is 300. The tolerances of the gene diversity and population diversity are set to 0.01 and 0.1, respectively. The setting-factors used in the CODEQ method to solve this example as follows. The population size is set to 5. The maximum generation is 300. These initial-setting factors for the DE method are the same as that for the CODEQ except that DE uses the scaling factor fixed to 0.1 and the crossover factor fixed to 0.5. For comparison, the six strategies of mutation operation of DE method are respectively used to solve this example. The solution for this example is repeatedly solved one hundred times. The best and worst values among the best solutions of the one hundred runs are respectively expressed in Table 1. The average for the best solutions of the one hundred runs and the standard deviation with respect to the average are also shown in this table. A smaller standard deviation implies that almost all the best solutions are close to the average best solution. That is, it has low sensitivity with respect to the different initial population. From the Table 1, the standard deviation for the HCODEQ method is smaller than all mutation strategies of DE method and CODEQ method. And the average best value of the HCODEQ method is smaller than DE and CODEQ methods. So, the parameter selection problem is alleviated. That implies that the HCODEQ method is a robust method compared with DE and CODEQ methods. Table 2 lists the computational results when the population size is reassigned to 10 to solve this example one hundred times again. From the computation results, the convergent properties of the HCODEQ method are better than the DE method and CODEQ method. The numbers of the parameter used in the HCODQ, DE, and CODEQ methods are 4, 5, and 2, respectively. Although the number of the parameters used in HCODEQ method is greater than that of the CODEQ method, the parameter selection problem in HCODEQ method is alleviated than in the CODEQ method. The number of times that these best solutions were smaller than 0.00001 are also shown in Tables 1 and 2. From Table 1, the number of the successful runs for the best solutions that were smaller than 0.00001 is 40, 29, 44, 73, 70, and 19

for six different strategies of mutation operations. The number of successful runs for the best solutions that were smaller than 0.00001 is 79 and 98 in the CODEQ and HCODEQ methods, respectively. From Table 2, the number of successful runs for the best solutions that were smaller than 0.00001 is 89, 98, 100, 100, 99, and 96 for six different strategies of mutation operation. The number of the successful runs for the best solutions that were smaller than 0.00001 is 100 for both the CODEQ and HCODEQ methods. Based on the computational results, the convergence property of the HCODEQ method is outperformed than the DE and CODEQ methods.

Table 1: Computation Results for One Hundred Runs of Example 1, population size = 5

Mutation Strategy	**1**	**2**	**3**	**4**	**5**	**6**	**CODEQ**	**HCODEQ**
Best	3.10e-17	4.35e-23	1.52e-24	1.98e-12	2.47e-14	1.60e-19	0	1.77e-10
Worst	9.290304	149.9470	8.871615	1.927184	8.136053	3665.115	6.304407	0.0042637
Average	0.371408	2.21562	0.558783	0.028183	0.093700	37.36613	0.103410	4.31e-05
STD	1.270686	15.00289	1.470137	0.195969	0.815074	366.4436	0.638052	4.26e-04
Count	40	29	44	73	70	19	79	98

Table 2: Computation Results for One Hundred Runs of Example 1, population size = 10

Mutation Strategy	1	2	3	4	5	6	CODEQ	HCODEQ
Best	3.99e-19	1.99e-26	0	2.70e-15	5.04e-16	7.78e-20	0	1.66e-10
Worst	2.873036	0.457369	3.02e-09	2.48e-06	1.32e-05	1.087578	1.61e-27	5.79e-07
Average	0.039089	0.008802	3.02e-11	6.24e-08	1.92e-07	0.010948	1.67e-29	8.94e-08
STD	0.290182	0.061975	3.02e-10	3.29e-07	1.43e-06	0.108753	1.61e-28	1.24e-07
Count	89	98	100	100	99	96	100	100

Example 2. Let us consider the minimization problem is described by

$$\min_{Z_1 - Z_2} J(Z_1, Z_2) = \frac{1}{\frac{1}{K} + \sum_{j=1}^{25} f_j^{-1}(Z_1, Z_2)} \tag{12}$$

where $f_j(Z_1, Z_2) = c_j + \sum_{i=1}^{2} (Z_i - a_{ij})^6$, $-65.536 \le Z_1, Z_2 \le 65.536$, $K = 500, c_j = j$ and

$$[a_{ij}] = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$$

The second benchmark function is the Fifth function of De Jong. This is a multimodal test function. This function has a global minimum value of 0.998 at $(Z_1, Z_2) = (-32, -32)$ as also shown by Michalewicz (1999).

In Example 2, the parameters for the HCODEQ, CODEQ, and DE methods are selected as those of Example 1. The solution for this example is repeatedly solved one hundred times. The best and worst values among the best solutions of the one hundred runs are respectively expressed in Table 3. The average for the best solutions of the one hundred runs and the standard deviation with respect to the average are also shown in this table. From Table 3, the standard deviation for the HCODEQ method is smaller than all mutation strategies of DE method and CODEQ method. And the average best value of the HCODEQ method is smaller than DE and CODEQ methods. That implies that the HCODEQ method is a robust method compared with DE and CODEQ methods again. From the computation results, the convergent properties of the HCODEQ method are better than that of the DE method and CODEQ method. Table 4 lists the computational results when the population size is reassigned to 10 to solve this example one hundred times again. From Table 4, the standard deviation for the HCODEQ methods is smaller than that of all mutation strategies of DE and CODEQ methods. And the average best value of the HCODEQ method is less than that of DE and CODEQ methods. A smaller standard deviation also implies that the method has a low sensitivity with respect to the different initial population. So the parameter selec-

tion problem of the HCODEQ method is alleviated.

Example 3: Let us consider the minimization problem as described by

$$\min_{z_1,z_2} J(z_1,z_2) = [1+(z_1+z_2+1)^2(19-14z_1+3z_1^2-14z_2+6z_1z_2+3z_2^2)]\times[30+(2z_1-3z_2)^2(18-32z_1+12z_1^2+48z_2-36z_1z_2+27z_2^2)] \quad (13)$$

where

$$-2 \le z_1, z_2 \le 2$$

The third benchmark function is the Goldstein Price function which is a continuous, differentiable, non-separable, non-scalable, and multimodal function. This function has a global minimum value of 3.00 as also shown by Michalewicz (1999). In Example 3, the parameters for the HCODEQ, CODEQ, and DE methods are selected as those of Examples 1 and 2. The solution for this example is repeatedly solved one hundred times. The best and worst values among the best solutions of one hundred runs are expressed in Table 5. The average for the best solutions of one hundred runs and the standard deviation with respect to the average are also shown in this table. Table 6 lists the computational results when the population size is re-assigned to 10 to solve this example one hundred times again. From the computational results in Tables 5 and 6, the convergence property of the HCODEQ has outperformed other methods.

Example 4: Let us consider the minimization problem as described by:

$$\min_{z_1,z_2} J(z_1,z_2) = \left(4-2.1z_1^2+\frac{z_1^4}{3}\right)z_1^2 + z_1z_2 + (-4+4z_2^2)z_2^2 \quad (14)$$

where $-3 \le z_1 \le 3$ and $-2 \le z_2 \le 2$

The fourth benchmark function is the Six-Hump Camel Back Function. The Six-Hump Camel Back Function is a global optimization test function. Within the bounded region of it owns six local minima, two of them are global ones (Molga & Smutnicki, 2005). Like the Goldstein Price function, the Six-Hump Camel Back Function is also a continuous, differentiable, non-separable, non-scalable, and multimodal function. This function has a global minimum value of -1.0316 as also shown by Michalewicz (1999). In Example 4, the parameters for HCODEQ, CODEQ, and DE methods are selected as those of Examples 1, 2, and 3. The solution for this example is repeatedly solved one hundred times. The best and worst values among the best solutions of the one hundred runs are respectively expressed in Table 7. The average for the best solutions of the one hundred runs and the standard deviation with respect to the average are also shown in this table. Table 8 lists the computational results when the population size is reassigned to 10 to solve this example one hundred times again. From the computational results in Tables 7 and 8, the convergence property of the HCODEQ has outperformed that of other methods.

Table 3: Computation Results for One Hundred Runs of Example 2, population size = 5

Mutation Strategy	1	2	3	4	5	6	CODEQ	HCODEQ
Best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004
Worst	23.80943	20.15349	23.80943	21.07269	20.15349	29.46829	12.67051	10.76318
Average	6.638564	7.660642	9.558399	3.534609	3.467216	7.254276	5.137431	1.514822
STD	5.958980	5.351153	6.847977	4.415153	4.328542	5.999762	3.963695	1.490098
Count	23	9	12	50	58	18	22	75

Table 4: Computation Results for One Hundred Runs of Example 2, population size = 10

Mutation Strategy	1	2	3	4	5	6	CODEQ	HCODEQ
Best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004
Worst	12.67051	16.44091	21.98841	10.76318	15.50382	10.76318	11.71870	1.992554
Average	3.679119	4.374463	7.520116	1.677553	1.55700	2.51205	1.875342	1.007950
STD	3.401745	4.240636	5.983056	1.831971	1.997564	2.771254	1.936128	0.099455
Count	43	28	10	77	84	60	72	97

Table 5: Computation Results for One Hundred Runs of Example 3, population size = 5

Mutation Strategy	1	2	3	4	5	6	CODEQ	HCODEQ
Best	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Worst	86.43120	84.00614	1226.673	84.00000	84.00000	840.0000	84.00	3.00
Average	8.218919	13.37741	36.23754	10.29000	5.430000	21.39463	5.5444273	3.00
STD	17.15447	22.96446	148.3339	21.31689	12.19311	85.47584	10.257319	2.37e-10
Count	83	70	66	87	95	55	86	100

Table 6: Computation Results for One Hundred Runs of Example 3, population size = 10

Mutation Strategy	1	2	3	4	5	6	CODEQ	HCODEQ
Best	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Worst	3.00	84.00	84.00	30.00	3.00	3.00	3.00	3.00
Average	3.00	5.160005	5.70	3.81	3.00	3.00	3.00	3.00
STD	1.84e-15	9.918590	12.43529	4.629058	2.22e-15	1.59e-15	2.48e-15	2.56e-15
Count	100	94	94	97	100	100	100	100

Table 7: Computation Results for One Hundred Runs of Example 4, population size = 5

Mutation Strategy	1	2	3	4	5	6	CODEQ	HCODEQ
Best	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
Worst	-0.21546	-0.21546	-0.21546	-0.21546	-0.21546	-0.11326	-1.008884	-1.0316
Average	-1.01513	-1.03163	-0.98741	-1.01531	-1.02347	-1.00195	-1.031298	-1.0316
STD	0.114818	0.081609	0.181142	0.114839	0.081616	0.161086	0.0022964	1.64e-08
Count	93	95	88	98	99	86	86	100

Table 8: Computation Results for One Hundred Runs of Example 4, population size = 10

Mutation Strategy	1	2	3	4	5	6	CODEQ	HCODEQ
Best	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
Worst	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
Average	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
STD	1.12e-15	4.63e-14	1.12e-15	1.12e-15	1.85e-12	1.12e-15	1.12e-15	5.95e-09
Count	100	100	100	100	100	100	100	100

4. Conclusion

The convergence property of HCODEQ, CODEQ, and DE methods are compared via four benchmark functions from the literature. The concepts of chaotic search, opposition-based learning, and quantum mechanics are used in the CODEQ method to overcome the drawback in selecting the crossover factor, scaling factor, and mutation operator used in the original differential evolution (DE) method. The main idea for the HCODEQ method is to use two operations, migrating operation and acceleration operation, to act as a trade-off operator to overcome the drawback associated with the use of a larger population size in CODEQ method. The use of the acceleration operation can speed up the convergence of HCODEQ. And the population diversity can be maintained by the migrating operation. The numbers of the parameter used in the HCODQ, DE, and CODEQ methods are 4, 5, and 2, respectively. Although the number of the parameters used in HCODEQ method is greater than the CODEQ method, the parameter selection problem in HCODEQ method is alleviated than in CODEQ method. From the computational results of the four examples, the convergence property of the HCODEQ method is better than that of CODEQ and DE methods. Finally, the proposed HCODEQ method can be used to solve the optimization problem in the management field, for example, production and inventory control, logistics network, and so on.

Acknowledgement

Financial research support from the Ministry of Science and Technology of the R. O. C. under the grant MOST 104-2221-E-131-017 is greatly appreciated.

References

Amjad, A. M., Salam, Z., & Saif, A. M. A. (2015). Application of differential evolution for cascaded multilevel VSI with harmonics elimination PWM switching. *International Journal of Electrical Power and Energy Systems, 64*, 447-456.

Chiou, J. P., & Chang, C. F. (2010). CODEQ algorithm for feeder reconfiguration in distribution systems. *Proc. of the 31st Symp. on Electric Power Eng.*, 2587-2590.

Chiou, J. P., & Chang, C. F. (2009). Solving the economic dispatch problems by CODEQ algorithm. *Proc. of the 30th Symp. on Electric Power Eng.*

Havangi, R., Nekoui, M. A., Teshnehlab, M. & Taghirad, H. D. (2014). A SLAM based on auxiliary marginalized particle filter and differential evolution. *International Journal of Systems Science, 45*(9), 1913-1926.

Jamil, M., & Yang, X. S. (2013). A literature survey of benchmark functions for global optimization problems. *Int. Journal of Mathematical Modelling and Numerical Optimisation, 4*(2), 150–194. DOI: 10.1504/IJMMNO.2013.055204

Lin, Y. C., Hwang, K. S., & Wang, F. S. (2000). Plant scheduling and planning using mixed-integer hybrid differential evolution with multiplier updating. *Congress on Evolutionary Computation*, San Dego, CA. 593-600.

Michalewicz, Z. (1999). Genetic algorithms + data structures = evolution programs (3rd ed.). New York, NY: Springer.

Molga, M., & Smutnicki, C. (2005). Test functions for optimization needs. http://www.vafaeijahan.com/en/wp-content/uploads/2012/02/Test-functions-for-optimization-needs.pdf

Omran, M. G. H., & Salman, A. (2009). Constrained optimization Using CODEQ. *Chaos, Solitons and Fractals, 42*(2), 662-668.

Omran, M. G. H., & Salman, A. (2010). Improving the performance of CODEQ using quadratic interpolation. *2nd International Conference on Agents and Artificial Intelligence Proceedings, 1*, 265-270.

Omran, M. G. H., & Salman, A. (2009). Optimization of discrete values using recent variants of differential evolution. *Proceedings of IASTED International Conference on Computational Intelligence*, 108-113.

Omran, M. G. H. (2010). CODEQ: An effective metaheuristic for continuous global optimization. *International Journal of Metaheuristics, 1*(2), 108-131.

Price, K. V. (1997). Differential evolution vs. functions of the 2nd ICEC. *IEEE Conference on Evolutionary Computation*, 153-157, Indianapolis.

Reddy, K. S., & Sahoo, S. K. (2014). An approach for FIR filter coefficient optimization using differential evolution algorithm. *AEU – International Journal of Electronics and Communications.*

Storn, R., & Price, K. V. (1996). Minimizing the real functions of the ICEC '96 contest by differential evolution. *IEEE Conference on Evolutionary Computation*, 842-844.

Zamuda, A., & Brest, J. (2014). Vectorized procedural models for animated trees reconstruction using differential evolution. *Information Sciences, 278*, 1-21.

About Authors

Ji-Pyng Chiou, received his Ph. D. degree in Electrical Engineering, from National Chung Cheng University, Chiayi, Taiwan. He is currently an Assistant Professor of the Department of Electrical Engineering at Mingchi University of Tech nology, Taipei, Taiwan. His research interest is in the application of evolutionary algorithms.

Chung-Fu Chang, received his MS degree in Electric Engineering from Chung Tuan Christian University, Chung-Li, Taiwan. His field of interest includes distribution system planning, reliability engineering, optimization techniques, and AI application. He is working toward his Ph.D. in Electric Engineering at National Chung Cheng University, Chiayi, Taiwan.

Yung-Lin Wang is working toward his Master's Degree in Electric Engineering at Ming Chi University of Technology, New Taipei, Taiwan.