



A Hybrid Differential Evolution Algorithm and Particle Swarm Optimization with Alternative Replication Strategy

Lulu Zuo¹, Lei Liu¹, Hong Wang^{1(✉)}, and Lijing Tan^{2(✉)}

¹ College of Management, Shenzhen University, Shenzhen 518060, China
ms.hongwang@gmail.com

² College of Management, Shenzhen Institute of Information Technology,
Shenzhen 518172, China
mstlj@163.com

Abstract. A new hybrid algorithm, combining Particle Swarm Optimization (PSO) and Differential Evolution (DE), is presented in this paper. In the proposed algorithm, an alternative replication strategy is introduced to avoid the individuals falling into the suboptimal. There are two groups at the initial process. One is generated by the position updating method of PSO, and the other is produced by the mutation strategy of DE. Based on the alternative replication strategy, those two groups are updated. The poorer half of the population is selected and replaced by the better half. A new group is composed and conducted throughout the optimization process of DE to improve the population diversity. Additionally, the scaling factor is used to enhance the search ability. Numerous simulations on eight benchmark functions show the superior performance of the proposed algorithm.

Keywords: Particle Swarm Optimization · Differential evolution algorithm
Alternative replication strategy

1 Introduction

Particle Swarm Optimization (PSO), proposed by Kennedy and Eberhart [1], is a well-known global intelligence optimization algorithm. On the basis of individual perception and social perception, the particles of PSO are replaced to enhance the speed of convergence and search accuracy in every iteration. Since then, many modifications have been developed. A recent version of modification was to update the particle positions by using two selected functions and combining individual experiences to avoid getting into the local optimum [2]. A fitness function was defined in [3] to identify the parameters of nonlinear dynamic hysteresis model. To a certain extent, the algorithm improves the convergence of the progress. However, a common problem occurring in the simulation experiments is still the premature clustering in the early part of iteration procedure.

Differential Evolution (DE) [4] is a population-based method of function optimization like Genetic algorithm (GA), including mutation, crossover, and selection. In DE, new individuals are generated from the information of multiple previous

individuals to get out of stagnation. Hence, some improved DE algorithms are proposed for getting better consequents. An improvement of DE with Taguchi method of sliding levers had the powerful ability of global search and obtained a better solution [5]. The fuzzy selection operation was used in an improved DE to reduce the complexity of multiple attribute decisions and enhance population diversity [6]. Though the DE has some advantages in global optimization, there is a dependency on controlling parameters which are not easy to decide when confronting high-dimensional complex problems.

In view of the advantages and disadvantages of PSO and DE, many hybrid algorithms of PSO and DE are described to combine the advantages of both and better solve various practical problems. The hybrid PSO and DE with population size reduction (HPSODEPSR) [7] can achieve optimal or near optimal solution faster than other comparison algorithms. The Bacterial Foraging Optimization (BFO) hybridized [8] with PSO and DE was introduced to effectively handle the dynamic economic dispatch problem. In order to improve the diversity and make each subgroup achieve a different optimal solution in the range of fitness values, Zuo and Xiao [9] used a hybrid operator and multi-population strategy performing the PSO and DE operation in turn. A novel hybrid algorithm PSO-DE [10] jumped out of stagnation, increased the speed of convergence and improved the algorithm's performance. A modified algorithm hybridizing PSO and DE with an aging leader and challengers was advanced to find the optimal parameters of PID controller quickly [11]. Though those hybrid algorithms improve the performance of the original algorithms (i.e. PSO and DE), premature stagnation is still a major problem.

This paper presents a novel hybrid algorithm of PSO and DE (DEPSO) with alternative replication strategy to overcome the above-mentioned problems. In the proposed algorithm, population is separated into two groups which are generated by two different methods, i.e. velocity updating strategy of PSO and mutative strategy of DE. A novel population are produced according to alternative replication strategy. The poor half of the population is eliminated while the other half is reproduced for the new evolution. In order to enhance the diversity of the population, the scaling factor of DEPSO is adjusted according to the linear decreasing rule.

The remaining paper is organized as follows. A brief introduction of PSO algorithm and DE algorithm is provided in Sect. 2. In Sect. 3, the hybrid algorithm of PSO and DE with alternative and replication strategy is described in detail. Section 4 gives experimental results of the Simulations on benchmark functions. Finally, the conclusion is presented in Sect. 5.

2 Description of Algorithms

2.1 Particle Swarm Optimization

In standard PSO algorithm, a group of particles flies in the search space to find the optimal location. The particles are given random positions x and velocities v in the initiate progress. In each iteration, the best position of each particle $pbest_{id}$ and the best

position of global $gbest_{id}$ are learnt, which leads the particle to the new position. Equations (1)–(2) are the updated rule [1]:

$$v_{id}^{G+1} = \omega v_{id}^G + c_1 rand(0, 1)(pbest_{id}^G - x_{id}^G) + c_2 rand(0, 1)(gbest_{id}^G - x_{id}^G) \quad (1)$$

$$x_{id}^{G+1} = x_{id}^G + v_{id}^{G+1}, i = 1, \dots, N, d = 1, \dots, D \quad (2)$$

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) * G/G_{max} \quad (3)$$

where x_{id} and v_{id} are the position and velocity of the i^{th} particle, respectively, N is the number of particles, and D is the dimensions of search space, c_1 and c_2 are acceleration factors. Finally, ω is the inertia weight adjusted by Eq. (3) in [12], ω_{max} and ω_{min} are the maximum and the minimum value of inertia weight, respectively. G and G_{max} are the current number of iteration and the maximum number of iteration, separately.

2.2 Differential Evolution Algorithm

DE, proposed by Storn and Price [4], has the significant effect on solving application problems. The outline of DE can be described as follows:

Step 1: initialize individuals according to the upper and lower bounds of search space, and evaluate the fitness of each individual.

Step 2: compare the fitness of every individual and record the best individual.

Step 3: generate new vectors through mutation process. The mutation rule is as follows Eq. (4):

$$v_i = x_{r1} + F(x_{r2} - x_{r3}), i = 1, \dots, N \quad (4)$$

where N is the number of individuals, x_i is the i^{th} individual and v_i is the updated i^{th} vector through mutating. F is the scaling factor, r_1, r_2 and r_3 not equal to each other are randomly selected from $[1, N]$.

Step 4: cross populations and mutant vectors to get a trial vector. Equation (5) is the crossover formula:

$$u_{ij} = \begin{cases} v_{ij} & \text{if } (rand_j(0, 1) \leq CR) \text{ or } j = j_{rand} \\ x_{ij} & \text{otherwise} \end{cases}, j = 1 \dots D \quad (5)$$

where D is the number of parameters, while u_{ij} represents the i^{th} individual at the j^{th} search space after crossing operation. CR is the crossover probability, and j_{rand} is a randomly selected index in the range of $[0, D]$.

Step 5: The greedy algorithm (i.e. Eq. (6)) is used to select individuals for the next iteration process.

$$x_i = \begin{cases} u_i & \text{if } f(u_i) \leq f(x_i) \\ x_i & \text{otherwise} \end{cases} \quad (6)$$

3 DEPSO Algorithm with Alternative Replication Strategy

Due to rapid information search, the suboptimal solutions might be more frequently obtained by PSO. Different from the PSO, population of DE tends to be more diversity as the number of iterations increases but consumes more computational complexity. To take advantages of those two algorithms, in this paper, a new hybrid DEPSO method is proposed to improve the search capability of particles with smaller computational time.

For the purpose of preventing individuals from sinking into suboptimal solution, we use the following alternative replication strategy to optimize the initial particles of each iteration.

At the beginning of each iteration, a new group P_1 is generated by PSO algorithm (i.e. Eqs. (1)–(2)). Considering the optimal value of every individual x_{pbest} , another group P_2 is renewed by the mutation process of DE algorithm (i.e. Eq. (7)). Based on the fitness value, we compare the updated groups (i.e. P_1 and P_2) with the initial group P_0 and preserve the better individuals to form new P_1 and P_2 .

$$x_i = F(x_{pbest} - x_i) + F(x_{r4} - x_{r5}), i = 1 \dots N \quad (7)$$

where N is individuals' number, x_i is the i individuals, F is the scaling factor, and r_4 and r_5 are indexes selected from $[1, N]$.

Sort the individuals of new groups (i.e. P_1 and P_2) according to the fitness values, and the sorted groups are also compared to retain the superior individuals constituting a group P_3 . The new group P_3 eliminates half of the individuals with poor fitness values, and the rest of the individuals are reproduced to keep the number of individuals.

In order to overcome the shortcoming of population reduction, the mutation factor decreases linearly with the number of iterations increasing in the mutation procedure. The scaling factor is controlled in Eq. (8).

$$F = F_{max} - (F_{max} - F_{min}) * G / G_{max} \quad (8)$$

where F_{max} , F_{min} , G , and G_{max} are the maximum mutation factor, minimum mutation factor, the current number of iteration and the maximum number of iteration, respectively.

Finally, DE algorithm is re-simulated based on the group which is obtained using the alternative replication strategy. The scheme of DEPSO is described as follows in detail:

- Step 1: initialize the position and velocity of every individual, and generate initial group P_0 .
- Step 2: calculate the fitness of each individual, and evaluate the best solution of each individual $pbest$ and the best solution of all individuals $gbest$.
- Step 3: update the position and velocity of individuals using Eqs. (1)–(2), and generate new group P_1 .

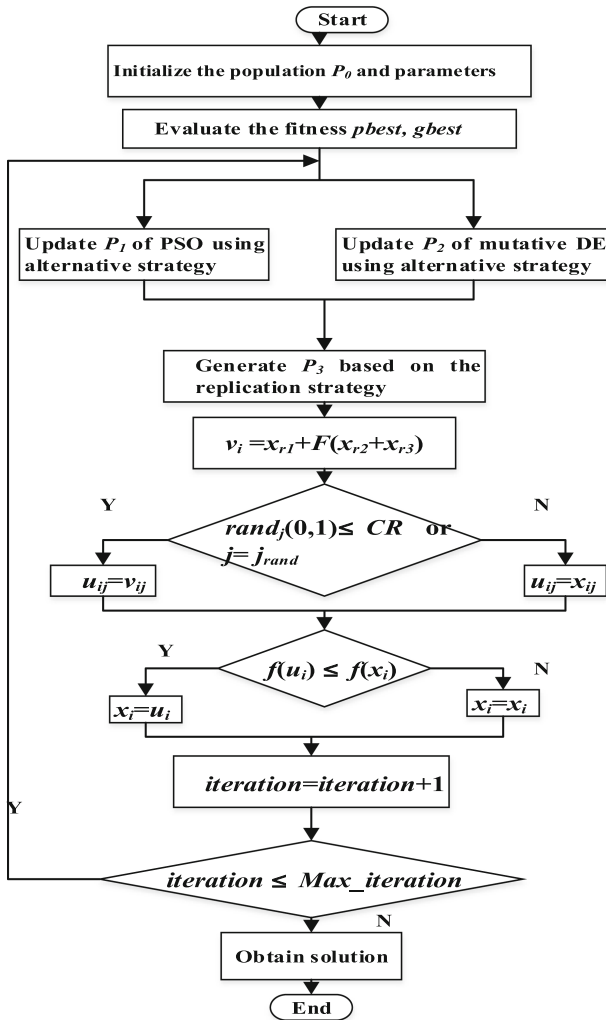


Fig. 1. Flowchart of DEPSO

- Step 4: mutate the initial individuals (i.e. Eq. (7)) and introduce a new group P_2 .
- Step 5: calculate the fitness of two groups (i.e. P_1 and P_2), and compare the fitness with the initial group P_0 to update the groups P_1 and P_2 , respectively.
- Step 6: P_1 and P_2 are sorted based on the fitness. A new group P_3 is constituted of the superb individuals in the comparison of the sorted groups. The group P_3 is updated by alternative replication strategy.
- Step 7: new vectors are formed by the group P_3 in the light of mutation process (i.e. Eq. (4)).
- Step 8: Equation (5) is applied to get trial vectors through the crossover of individuals and mutant vectors in DE algorithm.
- Step 9: select the best individuals using Eq. (6) and new offspring P_0 are introduced to execute the iterative procedure. Figure 1 presents the flowchart of DEPSO.

4 Experiments and Analysis

4.1 Benchmark Functions and Algorithms

To verify the performance of the proposed DEPSO algorithm, eight benchmark functions [13] (i.e. Sphere f_1 , SumPowers f_2 , Rosenbrock f_3 , Quartic f_4 , Rastrigin f_5 , Griewank f_6 , Ackley f_7 , Schwefel2.22 f_8) are applied to test the improved algorithm. One important reason for choosing these eight functions is that they contain unimodal functions (i.e. f_1 , f_2 , f_3 , and f_4) and multimodal functions (i.e. f_5 , f_6 , f_7 , and f_8). Additionally, these functions are minimum problems and the minimum value is known to be zero. The performance of DEPSO is demonstrated through eight benchmark functions and compared with some classic algorithms, i.e. PSO, DE, Genetic algorithm (GA) [14], Artificial Bee Colony algorithm (ABC) [15, 16], and Bacterial Foraging Optimization (BFO) [17]. All functions are tested on these optimization algorithms through MATLAB R2014a software.

4.2 Experimental Parameters

The similar parameters of six optimization algorithms are set as follows: the size of the population is 50, the maximum iterative number is set to 5000. Each function is run 30 times with the search space dimension 30, 50, and 80. The parameters of GA, ABC and BFO are from [18] except the reproduction's number of BFO is 25. More parameters setting of PSO, DE and DEPSO are shown as follows: In PSO, $c_1 = c_2 = 2$, $\omega_{max} = 0.9$, and $\omega_{min} = 0.4$; In DE, $CR = 0.5$, and F ranges from 0.4 to 0.9; In DEPSO, c_1 , c_2 , ω_{max} and ω_{min} are the same as PSO. CR and F are the same as DE. The values are the results of multiple simulations.

Table 1. Experiment results on benchmark functions

<i>Funs.</i>	<i>D</i>	<i>Ret.</i>	PSO	DE	GA	ABC	BFO	DEPSO
f_1	30	mean	4.73E-33	1.86E-40	2.76E-06	1.47E-17	3.98E-01	0.00E+00
		std	1.43E-32	1.86E-40	2.07E-06	2.62E-17	8.93E-02	0.00E+00
	50	mean	4.75E-14	2.03E-16	6.49E-04	2.02E-02	4.36E-01	0.00E+00
		std	6.19E-14	6.72E-17	8.40E-06	2.49E-03	6.02E-01	0.00E+00
	80	mean	5.83E-05	3.26E-05	4.12E-02	6.91E+03	1.01E+00	0.00E+00
		std	6.91E-05	4.26E-07	1.01E-02	5.58E+02	1.42E+00	0.00E+00
f_2	30	mean	3.34E-58	1.90E-81	3.43E-07	3.89E-01	1.54E-04	0.00E+00
		std	4.39E-58	2.66E-81	2.37E-07	4.81E-01	2.17E-04	0.00E+00
	50	mean	1.53E-12	4.18E-26	5.25E-07	5.31E+09	2.06E-04	0.00E+00
		std	1.99E-12	5.86E-26	2.64E-09	5.03E+09	2.91E-04	0.00E+00
	80	mean	1.92E+09	2.42E+00	4.25E-07	1.55E+28	1.08E-03	0.00E+00
		std	2.64E+09	1.02E+00	3.64E-08	1.04E+28	1.52E-03	0.00E+00
f_3	30	mean	6.15E+01	2.11E+01	7.79E+01	5.56E+01	4.92E+01	2.63E+01
		std	4.87E+01	2.53E-01	7.07E+01	4.29E+01	2.78E+01	1.80E+00
	50	mean	1.24E+02	4.60E+01	9.47E+01	5.28E+07	1.02E+02	4.53E+01
		std	1.11E+02	3.62E-01	3.22E+01	1.08E+07	7.46E+01	1.79E-01
	80	mean	3.36E+02	1.57E+02	2.37E+02	3.46E+09	1.84E+02	7.69E+01
		std	4.59E+01	5.97E+00	1.07E+02	6.57E+08	1.48E+02	9.56E-01
f_4	30	mean	4.79E-48	3.79E-61	7.06E-17	2.69E-17	1.25E-01	0.00E+00
		std	6.20E-48	4.01E-61	1.43E-17	2.91E-17	1.77E-01	0.00E+00
	50	mean	5.84E-20	1.07E-24	7.39E-12	3.46E-01	9.02E-01	0.00E+00
		std	8.03E-20	9.60E-25	4.91E-13	2.38E-01	1.28E+00	0.00E+00
	80	mean	1.63E-08	1.99E-09	7.48E-09	1.83E+01	3.34E+00	0.00E+00
		std	1.69E-08	9.76E-10	2.52E-09	2.18E+00	4.73E+00	0.00E+00
f_5	30	mean	4.90E+01	1.31E+03	1.76E+01	1.94E+03	3.66E+02	0.00E+00
		std	2.59E+01	1.29E+02	6.53E-01	3.11E+01	5.17E+02	0.00E+00
	50	mean	1.50E+02	3.76E+03	2.31E+02	5.41E+03	5.78E+02	0.00E+00
		std	4.27E+01	4.86E+02	1.98E+02	3.31E+02	8.17E+02	0.00E+00
	80	mean	4.02E+02	8.44E+03	1.72E+03	1.07E+04	1.39E+03	0.00E+00
		std	3.02E+01	3.43E+02	4.60E+02	3.05E+02	1.96E+03	0.00E+00
f_6	30	mean	0.00E+00	0.00E+00	7.75E-03	1.74E-11	1.04E-02	0.00E+00
		std	0.00E+00	0.00E+00	1.07E-02	2.46E-11	1.41E-02	0.00E+00
	50	mean	9.86E-03	0.00E+00	1.20E-02	1.73E-01	2.19E-02	0.00E+00
		std	5.66E-14	0.00E+00	1.95E-03	1.03E-01	3.05E-02	0.00E+00
	80	mean	9.77E-02	2.86E-05	1.68E-01	6.18E+01	2.55E-02	0.00E+00
		std	1.24E-01	1.15E-05	2.54E-02	1.84E+00	3.57E-02	0.00E+00

(continued)

Table 1. (continued)

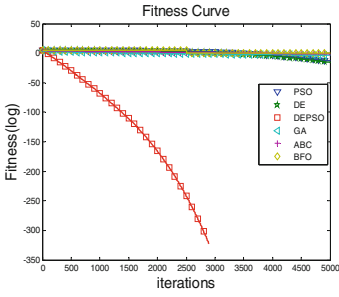
<i>Funs.</i>	<i>D</i>	<i>Ret.</i>	PSO	DE	GA	ABC	BFO	DEPSO
f_7	30	mean	1.15E-14	4.44E-15	3.46E+00	3.74E-09	6.76E-01	8.88E-16
		std	5.02E-15	0.00E+00	6.07E-01	2.38E-09	8.28E-01	0.00E+00
	50	mean	5.77E-07	3.39E-09	5.58E+00	2.74E-01	7.49E-01	8.88E-16
		std	7.10E-07	2.87E-10	7.20E-01	1.72E-01	9.64E-01	0.00E+00
	80	mean	1.98E-02	1.13E-03	4.15E+00	1.14E+01	8.33E-01	4.44E-15
		std	1.37E-02	1.75E-04	1.44E+00	8.78E-02	1.11E+00	0.00E+00
f_8	30	mean	1.26E-21	3.59E-24	2.22E+00	3.42E-15	1.85E+00	0.00E+00
		std	1.76E-21	1.92E-24	2.06E-01	1.25E-15	2.01E+00	0.00E+00
	50	mean	8.29E-11	3.03E-10	2.10E+00	1.77E-04	3.59E+00	0.00E+00
		std	3.55E-11	2.39E-11	2.47E-01	3.58E-06	4.27E+00	0.00E+00
	80	mean	1.08E-04	1.09E-03	2.88E+00	9.71E+00	5.93E+00	0.00E+00
		std	1.03E-04	4.51E-05	9.88E-01	9.38E-01	7.34E+00	0.00E+00

4.3 Experiment Results and Discussion

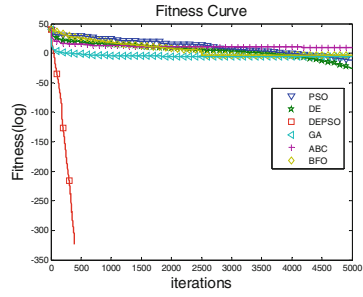
To get experiment results, all benchmark functions are executed by coding these search methods. The mean fitness value and the standard deviation obtained by six optimization algorithms are displayed in Table 1. The bold type is used to underline the best of all the numerical results attained by comparing six optimization algorithms. Figure 2 lists the convergence curves of different test functions gotten by all the algorithms with the dimension 50. In Fig. 2, in order to make the curves clear, the results are the logarithm with base 10.

As shown in Table 1, DEPSO algorithm almost gets the optimal mean and standard deviation among all the algorithms for the eight benchmark functions. It means that the hybrid DEPSO algorithm is better than other algorithms in terms of search accuracy. Additionally, the proposed algorithm obtains the minimum values (i.e. zero) on functions of Sphere, SumPowers, Quartic, Rastrigin, Griewank, and Schwefel2.22. It means that the convergence precision of the DEPSO algorithm is high in the selected functions.

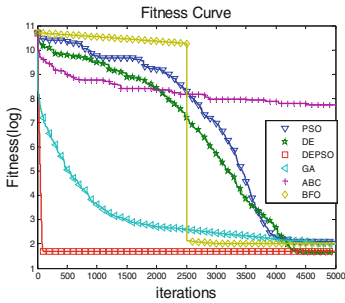
In terms of dimensionality, the DEPSO can perform better than other algorithms when the dimension is increasing. From Fig. 2, we can conclude the convergent speed of DEPSO algorithm is significantly faster than other algorithms and the convergent results are closer to optimal values. Altogether, the performance of DEPSO algorithm performs better whether in unimodal functions or in multimodal functions.



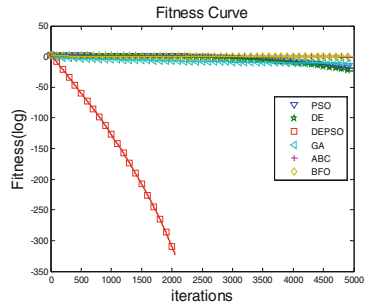
(1) Sphere function f_1



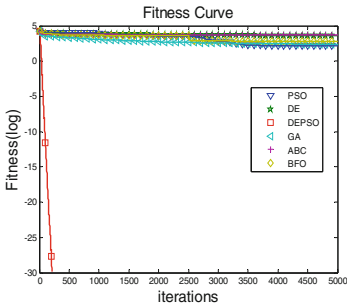
(2) SumPower function f_2



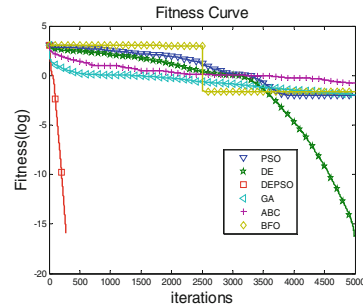
(3) Rosenbrock function f_3



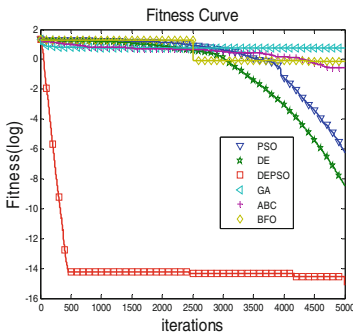
(4) Quartic function f_4



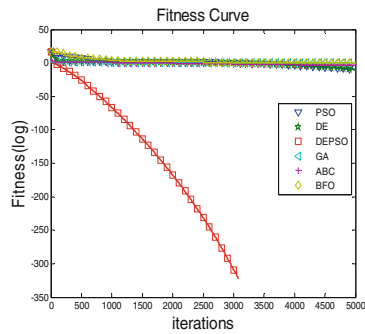
(5) Rastrigin function f_5



(6) Griewank function f_6



(7) Ackely function f_7



(8) Schwefel2.22 function f_8

Fig. 2. The iteration process of the algorithms when the dimensionality is 50

5 Conclusion

A hybrid algorithm integrating the advantages of PSO and DE is proposed in this paper. The results show that the DEPSO algorithm outperforms other algorithms in terms of mean and standard deviation. Though DE with nonlinearly decreasing mechanism of scaling factor can obtain the similar solutions on some benchmark functions (e.g., Rosenbrock and Rastrigin), the convergence speed is not good than the proposed DEPSO method. Thus, the proposed alternative replication strategy can enhance the performance of the hybrid algorithm. Our future study will focus on the application of the proposed algorithm to solve the real-world problems and more hybrid methods will be developed to obtain better solutions.

Acknowledgment. This work is partially supported by The National Natural Science Foundation of China (Grants No. 61472257), Natural Science Foundation of Guangdong Province (2016A030310074), Guangdong Provincial Science and Technology Plan Project (No. 2013B040403005), and Research Foundation of Shenzhen University (85303/00000155).

References

1. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, Piscataway, pp. 1942–1948 (1995)
2. Guedria, N.B.: Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl. Soft Comput.* **40**(40), 455–467 (2016)
3. Zhang, J., Xia, P.: An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models. *J. Sound Vib.* **389** (2016)
4. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1997)
5. Tsai, J.T.: Improved differential evolution algorithm for nonlinear programming and engineering design problems. *Neurocomputing* **148**, 628–640 (2015)
6. Pandit, M., Srivastava, L., Sharma, M.: Environmental economic dispatch in multi-area power system employing improved differential evolution with fuzzy selection. *Appl. Soft Comput.* **28**, 498–510 (2015)
7. Ali, A.F., Tawhid, M.A.: A hybrid PSO and DE algorithm for solving engineering optimization problems. *Appl. Math. Inf. Sci.* **10**(2), 431–449 (2016)
8. Vaisakh, K., Praveena, P., Rao, S.R.M.: A bacterial foraging PSO — DE algorithm for solving reserve constrained dynamic economic dispatch problem. In: 2011 IEEE International Conference on Fuzzy Systems, pp. 153–159 (2011)
9. Zuo, X., Xiao, L.: A DE and PSO based hybrid algorithm for dynamic optimization problems. *Soft. Comput.* **18**(7), 1405–1424 (2014)
10. Liu, H., Cai, Z., Wang, Y.: Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl. Soft Comput.* **10** (2010), 629–640 (2010)
11. Moharam, A., El-Hosseini, M.A., Ali, H.A.: Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers. *Appl. Soft Comput.* **38**, 727–737 (2016)
12. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference Evolutionary Computation, Anchorage, pp. 69–73 (1998)

13. Niu, B., Liu, J., Zhang, F., Yi, W.: A cooperative structure-redesigned-based bacterial foraging optimization with guided and stochastic movements. In: Huang, D.-S., Jo, K.-H. (eds.) ICIC 2016. LNCS, vol. 9772, pp. 918–927. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42294-7_82
14. Holland, J.H.: Adaption in natural and artificial systems. *Control Artif. Intell.* **6**(2), 126–137 (1975)
15. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Engineering Faculty, Computer Engineering Department, Erciyes University, Technical report - TR06 (2005)
16. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. *J. Global Optim.* **39**(3), 459–471 (2007)
17. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst.* **22**(3), 52–67 (2002)
18. Wang, H., Zuo, L., Liu, J., Yang, C., Li, Ya., Baek, J.: A comparison of heuristic algorithms for bus dispatch. In: Tan, Y., Takagi, H., Shi, Y., Niu, B. (eds.) ICSI 2017. LNCS, vol. 10386, pp. 511–518. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61833-3_54