



Dynamic economic dispatch based on improved differential evolution algorithm

Zheng Hongfeng¹

Received: 26 December 2017 / Revised: 3 January 2018 / Accepted: 5 January 2018 / Published online: 12 January 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Dynamic economic dispatch optimum scheduling of power plant generation is of great importance to electric utility systems, it is difficult to solve because of its complex structure, variable parameter, nonlinear characteristics et al. Based on analysis of DE searching mechanism, an improved differential evolution (IDE) algorithm based DE/target-to-best is presented, which adopts an improved mutation strategy that a random vector and the previous best vector is used instead of the current vector in case the DE algorithm may be in early maturity or decline in convergence speed. The algorithm is applied to solve the generators dynamic load economic dispatch problems taking into account the incremental fuel cost function and the valve-point effects. Computer simulation test shows that IDE algorithm provides better solution of less cost. In the case of less generations and outperforms GA, PSO and DE.

Keywords IDE algorithm · Generators · Dynamic load · Economic dispatch

1 Introduction

Dynamic load economic dispatch (here in after referred to as DLED) refers to the planning of the load dispatch of the online generators supply under known load dispatch in a certain period of time. The output of the generating unit is optimized under the condition of satisfying the system operating constraints, achieving the lowest total power generation costs of the power system [1–3].

Because of the complex structure, variable parameter, nonlinear characteristics of the unit and valve point effect (VPE) of the consumption curve, the power system mathematically presents a complex programming problem with non-linearity, high dimension, multi-constraint and difficult dimension, and it is difficult to solve the optimal DLED solution in theory. At present, the conventional methods of solving include dynamic programming (DP), Lagrange multiplier (LM), quadratic programming (QP), liner programming (LP), etc. but these methods are difficult to achieve the desired effect [3–5].

In recent years, the development of artificial intelligence and computer technology has provided new paths and meth-

ods for solving DED problems. Genetic algorithm (GA), artificial neural network (ANN), ant colony optimization (ACO), particle swarm algorithm (PSO) have achieved certain achievements in this field [4–7].

Differential evolution (DE) is a relatively new evolutionary algorithm based on Swarm Intelligent (SI) random search, which has been applied in engineering applications such as image restoration, aerodynamic optimization design and motor parameter identification. It has the advantages of fast convergence, good robustness, few parameters, simple program realization, etc. The disadvantages are that the optimization process sometimes falls into early maturity, or the convergence rate decreases [5–8].

In this paper, an improved differential evolution (IDE) is proposed, which is applied to the power system DLED based on full validation of the performance of the algorithm.

2 Basic DE algorithm

DE algorithm was first proposed by Rainer Storn and Kenneth Price in 1995 to solve the Chebyshev polynomial problem. The basic strategy is: set an initial population vector, and the difference vector of two randomly selected vectors is used as the third vector, and then the mutation is performed on the current population to produce a new individual.

✉ Zheng Hongfeng
hongfenglunwen@126.com

¹ Zhejiang Industry Polytechnic College, Shaoxing, China

Through cross operation, we choose whether the new individual produced by the mutation is retained in the population, recombine to produce a new generation of population, and gradually make the population evolve to the target state of the optimal solution. Set $X_{i,G}$ as the problem vector of the G th generation [6–9].

The variation vector $V_{i,G}$ is defined as:

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

where $i = 1, 2, \dots, ps$, ps is the population size, $r1, r2$ and $r3$ belong to different individuals in the population.

As in the other algorithms, in order to increase the diversity of the interference parameter vector, in the DE algorithm, the cross operation is introduced, and the target vector is generated by selecting the variation vector and the source vector in the population. The “greedy” selection pattern is usually used, that is, if and only if the target evaluation function value of the new individual is better, the new individual is retained in the next generation of group, otherwise the parent individual remains in the group and once again serve as the parent vector of the next generation.

The target vector is:

$$U_{ji,G} = (U_{1i,G}, U_{2i,G}, \dots, U_{Di,G}) \quad (2)$$

where $j = 1, 2, \dots, D$, D is the dimension of the problem.

The formula for each component is:

$$U_{ji,G} = \begin{cases} V_{ji,G}, & \text{if } \text{rand}_j(0, 1) \leq CR \vee j = k \\ X_{ji,G}, & \text{otherwise} \end{cases} \quad (3)$$

where $CR \in [0, 1]$ is the algorithm parameter, which needs to be determined in advance. It controls the diversity of the population and helps the algorithm detach from the local optimal solution. K is an integer chosen randomly from the dimension of the problem to ensure that one digit in the operation must be crossed. $\text{rand}_j(0, 1)$ is the control parameter randomly selected for the j th dimensional component.

3 IDE algorithm

Although the basic DE algorithm can solve the problem of target optimization, it is easy to fall into the local optimization and is difficult to jump out of local optimization to global optimization. To solve the premature, we need to adopt some strategies to strengthen the global search ability. To avoid falling into the local optimization, that is, to maintain the diversity of groups, we need to adopt self-adaptive mutation method to deal with repeated individuals.

DE/target-to-best algorithm strategy variation vector $V_{i,G}$ is defined as:

$$V_{i,G} = x_{r1,G} + F(pb\text{est}x_{i,G} - x_{i,G}) + F(x_{r2,G} - x_{r3,G}) \quad (4)$$

where $pb\text{est}x_{i,G}$ is the previous optimal vector, and substituting the cross vector in formula (1) with the cross vector in formula (4) is the basic method of the IDE algorithm.

For example, a solution of the minimum value by using DE and IDE algorithms are described as follows

```

DE algorithm
Begin
While  $NE < MAX_{NE}$  do
For each  $X_i$  do
Generate vector  $V_i$  according to formula (1);
Generate target vector  $U_i$  according to formula (3);
Calculate fitness (minimum) according to  $U_i$ ;
 $NE++$ ;
Generate the most suitable vector  $X_i$  between  $X_i$  and  $U_i$ ;
End For
End While
End
IDE algorithm
Begin
While  $NE < MAX_{NE}$  do
For each  $X_i$  do
Generate vector  $V_i$  according to formula (4);
Generate target vector  $U_i$  according to formula (3);
Calculate fitness (minimum) according to  $U_i$ ;
 $NE++$ ;
Generate the most suitable vector  $X_i$  between  $X_i$  and  $U_i$ ;
End For
End While
End

```

X_{i-} the i th vector in the population vector, V_i -the variation vector of X_i , U_i target vector, NE evolution iteration variable, and the preset maximum iteration number of MAX_{NE} .

To verify the excellent convergence performance of the IDE algorithm, ten target minimization functions are selected, including four unimodal functions ($f_1 - f_4$) and six multimodal functions ($f_5 - f_{10}$). The program is programmed with different algorithms. The functions are shown in Table 1 [4].

In the software parameter settings, for the IDE algorithm, CR takes 0.55, F takes 0.3; for the DE and IDE algorithms, the population size takes 150, MAX_{NE} is set to 1,000,000 in the program test. Each function runs 30 times using the two algorithms, the best deviation and standard deviation of the ten functions are shown in Table 2. It can be seen from Table 2 that the results of function f_3 are consistent under the two algorithms, the best deviation of function f_5 using DE algorithm is less than that of the IDE, and the other deviation index using IDE algorithm is superior to that of the DE [10–13].

Table 1 Ten target to minimize test functions

Functions	x_i	D	Global optimum
$f_1 = \sum_{i=1}^D x_i^2$	$x_i \in [-100, 100]$	30	0
$f_2 = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i$	$x_i \in [-10, 10]$	30	0
$f_3 = \sum_{i=1}^D (x_i + 0.5)^2$	$x_i \in [-100, 100]$	30	0
$f_4 = \sum_{i=1}^D i x_i^4 + rand[0, 1)$	$x_i \in [-1.28, 1.28]$	30	0
$f_5 = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	$x_i \in [-500, 500]$	30	-12,569.5
$f_6 = \sum_{i=1}^D [x_{i-10}^2 \cos(2\pi x_i) + 10]$	$x_i \in [-5.12, 5.12]$	30	0
$f_7 = -20 * \exp\left(-0.2 * \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$x_i \in [-32, 32]$	30	0
$f_8 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-600, 600]$	30	0
$f_9 = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^D (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$	$x_i \in [-50, 50]$	30	0
$f_{10} = \frac{\pi}{D} \{10 \sin^2(3\pi y_1) + \sum_{i=1}^D (y_i - 1)^2 [1 + \sin^2(3\pi y_{i+1})] + (y_D - 1)^2 [1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	$x_i \in [-50, 50]$	30	0

Table 2 The test results of DE and IDE

Functions	DE		IDE	
	Mean	Std dev	Mean	Std dev
f_1	9.8×10^{-16}	4.68×10^{-15}	7.70×10^{-37}	6.24×10^{-37}
f_2	1.43×10^{-08}	3.77×10^{-08}	9.06×10^{-23}	8.50×10^{-23}
f_3	0	0	0	0
f_4	5.49×10^{-03}	2.59×10^{-03}	3.80×10^{-03}	2.12×10^{-04}
f_5	$-5.81 \times 10^{+03}$	$8.58 \times 10^{+02}$	$-1.26 \times 10^{+04}$	0
f_6	$1.70 \times 10^{+02}$	$1.56 \times 10^{+01}$	$3.34 \times 10^{+01}$	$1.12 \times 10^{+01}$
f_7	6.39×10^{-09}	2.37×10^{-09}	5.52×10^{-15}	1.91×10^{-15}
f_8	9.29×10^{-16}	1.10×10^{-15}	0	0
f_9	5.86×10^{-17}	2.29×10^{-17}	2.99×10^{-17}	0
f_{10}	3.15×10^{-16}	2.16×10^{-16}	2.79×10^{-17}	0

Select functions f_1, f_2, f_4, f_5 and show their evolution process under DE and IDE algorithms, seen from Figs. 1, 2, 3 and 4, under the same number of iterations, the best fitness

value (log) obtained by IDE algorithm is less than that of the DE algorithm, indicating that the IDE algorithm makes it easier for the function to approach the target minimum value.

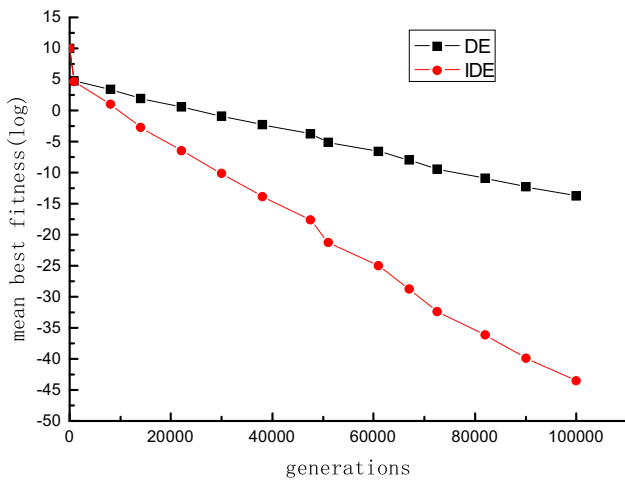


Fig. 1 Mean best fitness (log) comparison of f_1

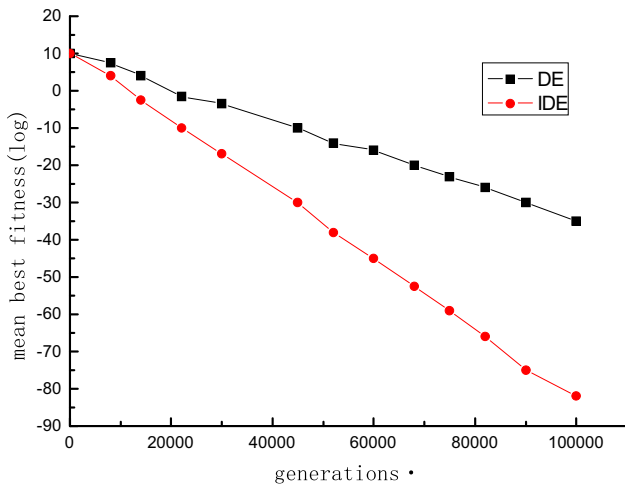


Fig. 2 Mean best fitness (log) comparison of f_2

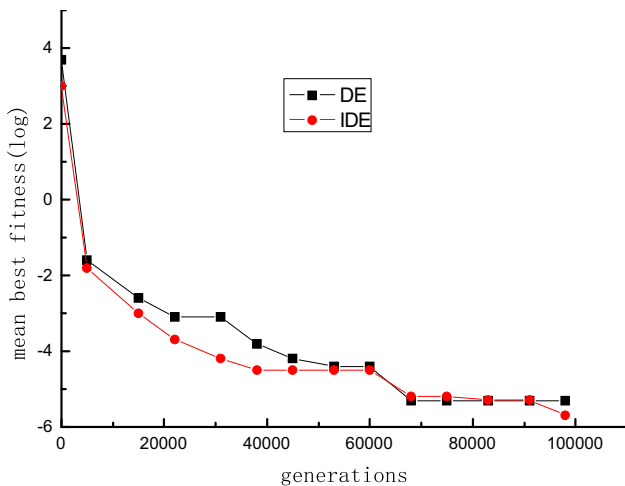


Fig. 3 Mean best fitness (log) comparison

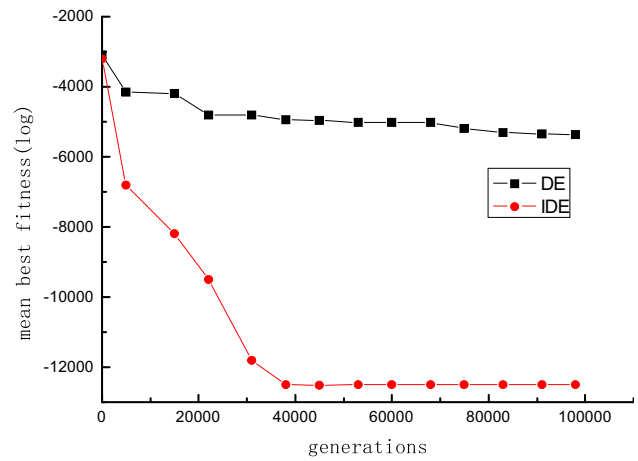


Fig. 4 Mean best fitness comparison of f_5

NSDE, SaDE, IDE algorithms are used for test and comparison, MAX_{NE} is set to 150,000 in the program test, the remaining software parameters are set the same as before, the best deviation values obtained by the three algorithms are shown in Table 3.

The results show that functions f_3 and f_5 obtained the same results by the three algorithms, functions f_6, f_9 and f_{10} obtained better results by SADE algorithm compared with IDE algorithm, function f_6 obtained better results by NSDE algorithm compared with IDE algorithm, the other functions obtained the best results by IDE algorithm.

4 DLED based on IDE algorithm

The target function of the DED problem can be expressed as

$$\text{Minimize } F = \sum_{t=1}^T \sum_{i=1}^{N_g} FC(i, t) \tag{5}$$

where F is the total power generation cost of the system; N_g is the total number of generators in the system; $FC(i, t)$ is the active power of the i th generator at moment t ; T is the system operating time [13–17].

The consumption characteristics of the i th generator can be expressed as

$$FC(i, t) = a_i + b_i P_{it} + c_i P_{it}^2 + |e_i \times \sin \{ f_i \times (P_{it}^{\min} - P_{it}) \}| \tag{6}$$

where e_i, f_i are the specific changes of generator consumption of the i th generator caused by VPE (the superimposed pulse effect on the unit's consumption curve resulting from

Table 3 Results achieved by NSDE, SaDE, IDE

Functions	NSDE Mean	SaDE Mean	IDE Mean
f_1	7.66×10^{-16}	7.40×10^{-20}	2.80×10^{-57}
f_2	4.41×10^{-10}	6.12×10^{-11}	1.17×10^{-34}
f_3	0	0	0
f_4	7.38×10^{-03}	1.15×10^{-02}	3.28×10^{-03}
f_5	$-1.26 \times 10^{+04}$	$-1.26 \times 10^{+04}$	$-1.26 \times 10^{+04}$
f_6	4.69×10^{-02}	3.98×10^{-08}	$2.98 \times 10^{+01}$
f_7	6.52×10^{-09}	8.96×10^{-11}	4.87×10^{-15}
f_8	4.58×10^{-15}	8.69×10^{-18}	0
f_9	5.53×10^{-17}	1.15×10^{-19}	2.95×10^{-17}
f_{10}	5.42×10^{-16}	1.65×10^{-19}	2.81×10^{-17}

wire drawing phenomenon in sudden opening of turbine intake valve) effect coefficient.

The constraints of the DED problem are:

Power balance constraint

$$\sum_{i=1}^{N_g} P_{it}^- P_{Dt} - P_{Lt} = 0 \tag{7}$$

where, $t=1, 2, \dots, T$, P_{Dt} —the total load demand at moment t , P_{Lt} —the line transmission loss at moment t , all can be calculated by formula (8).

$$P_{Lt} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_{it} B_{ij} P_{jt} \tag{8}$$

Generator operating capacity constraint

$$P_i^{\min} \leq P_{it} \leq P_i^{\max} \tag{9}$$

where, P_i^{\min} represents the minimum unit output, P_i^{\max} represents the maximum unit output.

Constraints of unit output climbing

$$P_{it} - P_{i(t-1)} \leq UR_i \tag{10}$$

$$P_{i(t-1)} - P_i \leq DR_i \tag{11}$$

where UR_i is the lower limit value of the climbing constraints for unit i , DR_i is the upper limit value of the climbing constraints for unit i .

Based on the above unit operating constraints, the algorithm steps to solve the DED problem by using the IDE algorithm are as follows:

① **Initialization**

The power generation of the i_{th} generator t at moment t is

$$P_{it} = P_i^{\min} + \rho \left(P_i^{\max} - P_i^{\min} \right) \tag{12}$$

where $\rho[0, 1]$ - represents a randomly assigned factor.

The DED target function is:

$$\text{Minimize } F = \sum_{t=1}^T \sum_{i=1}^{N_g} FC(P_{it}) + \sum_{Z=1}^{N_C} \lambda_Z |VIOL_Z| \tag{13}$$

To suppress the variation of grid power ratio, the modified DLED minimum function is used, where λ is the penalty factor, N_C is the number of suppression numbers, $VIOL$ is the suppression amplitude.

The IDE algorithm variation vector is

$$\hat{V}_{b,G+1} = V_{P,G} + F \times (P_{best} V_{i,G} - V_{i,G}) + F (V_{j,G} - V_{k,G}) \tag{14}$$

where, j and k are random numbers. G represents evolutionary algebra.

② **Selection**

$$V_{b,G+1} = \begin{cases} \hat{V}_{b,G+1}, & \text{if } F(\hat{V}_{b,G+1}) < F(\hat{V}_{b,G}) \\ \hat{V}_{b,G+1} - \alpha \nabla \Psi, & \text{otherwise} \end{cases} \tag{15}$$

where, $V_{b,G+1}$ represents the selection result, $\nabla \Psi$ represents the finite amplitude that can be calculated. The new $V_{b,N}$ can be solved by constantly correcting the replacement factor $\alpha [0, 1]$ and finally satisfy:

$$F(V_{b,N}) < F(V_{b,G+1}) \tag{16}$$

and then replace the worst individuals in the population with $V_{b,N}$ to improve the speed of rapid convergence of the system.

③ **Propagation**

With the continuous evolution of the population, each individual is gradually close to the optimal individual, the

Table 4 Generator parameters of generators 5 buses

Unit	P_i^{\min} MW	P_i^{\max} MW	a_i MW	b_i \$/MW	c_i \$/MW ²	e_i \$/h	f_i 1/MW	UR_i MW/h	DR_i MW/h
P1	10	75	20	2.0	0.0080	100	0.0042	30	30
p2	20	125	60	1.8	0.0030	140	0.0040	30	30
p3	30	175	100	2.1	0.0012	160	0.0038	40	40
p4	40	250	120	2.0	0.0010	180	0.0037	50	50
p5	50	300	40	1.8	0.0015	200	0.0035	50	50

Table 5 Load demand for 24 hours

Time (H)	Load MW	Time (H)	Load MW	Time (H)	Load MW	Time (H)	Load MW
1	408	7	625	13	698	19	649
2	437	8	655	14	696	20	709
3	473	9	685	15	649	21	682
4	532	10	709	16	585	22	603
5	555	11	716	17	557	23	524
6	613	12	744	18	609	24	466

difference vector generated by the mutation operation in the algorithm will gradually become smaller, which will weaken the diversity of the algorithm search space to a certain extent. Therefore, some mutation operations are necessary, so that the mutated vector can open up new search space, so as to improve the global optimization ability of the algorithm. The small cross factor causes the population to produce fewer new individuals after the cross operation, thus weakening the ability of the algorithm to develop new space. While large cross factors cannot keep the population stable and will reduce the stability of the algorithm.

The new individual is generated based on the previous optimum of $V_{hb,G+1}$, the h th gene of the i th individual can be described as:

$$V_{hi,G+1} = \begin{cases} V_{hb,G+1} + \delta_{hi} (V_{h \min} - Z_{hb}^{G+1}), & \text{if } \tilde{\delta}_{hi} < \frac{V_{hb,G+1} - V_{h \min}}{V_{h \max} - V_{h \min}} \\ V_{hb,G+1} + \delta_{hi} (V_{h \max} - V_{hb,G+1}), & \text{otherwise} \end{cases} \quad (17)$$

where δ and $\tilde{\delta}$ are two different random numbers. In order to make the new individual join the search population and help the system to avoid early aging, the propagation operation in the algorithm can be controlled by making the group density p less than the expected tolerance density $\epsilon 1$ in formula (18).

$$p = \frac{\left\{ \sum_{i=1}^{N_p} \left(\sum_{h=1}^n \eta_z \right) \right\}}{n (N_p - 1)} < \epsilon 1 \quad (18)$$

where

$$\eta_z = \begin{cases} 1, & \text{if } \left| \frac{V_{hi} - V_{hb}}{V_{hb}} \right| > \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

Reproduce new individuals by this method until no better individual is produced.

5 Computer simulation

Select five power generation units as the test system, the generator data parameters are as shown in Table 4, the load demand side considers the 24-hour change as shown in Table 5. In order to illustrate the robustness of the system, the test system takes into account the VPE effect, but eliminates the generator limit band, the effect data can be found in the reference [18–23].

Figure 5 is the dispatch plan of the 5 node units and the total dynamic load obtained by IDE algorithm.

The relationship between total unit cost curve and the number of iterations in DED problem obtained by using the IDE algorithm, GAGA, PSOPS are as shown in Fig. 6.

From the simulation results in Fig. 6, it can be seen that the cost of the dispatch plan of unit dynamic load obtained by the IDE algorithm is less than that of DE, GA and PSO, verifying the good performance of IDE algorithm in solving the unit DLED.

6 Conclusion

DLED is one of the key technologies in power system optimization and this paper proposes and validates an improved

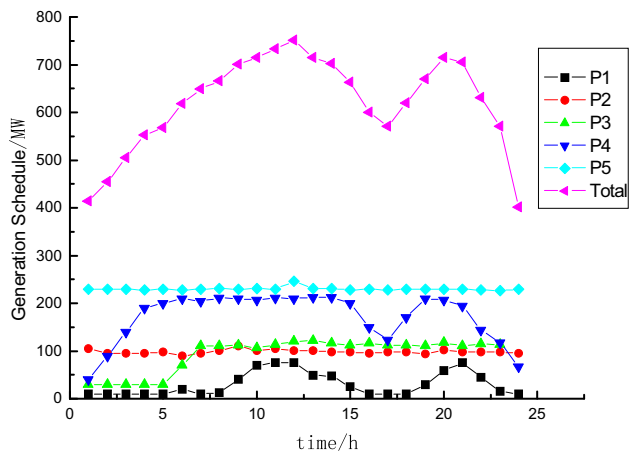


Fig. 5 Optimal generation schedules for 5 generators based on IDE

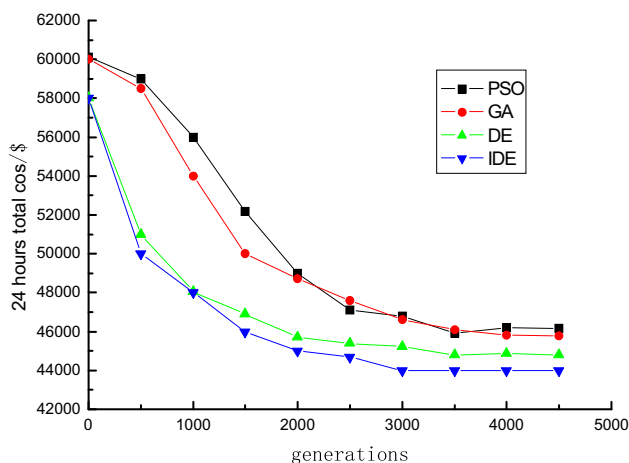


Fig. 6 Cost curves of different algorithms

DE. Through a comparison test of ten typical target optimization functions, compared with NSDE, SaDE and basic DE algorithms under the same number of iterations, IDE algorithm target function shows the best fitness. By applying IDE algorithm to power system DLED and comparing the cost curves under different algorithms, the result of IDE algorithm is the least cost and the convergence rate is the fastest compared with PSO, GA and DE algorithms.

Acknowledgements Zheng Hongfeng gratefully acknowledge the support through Zhejiang public welfare projects Grant (2016c31055) and Shaoxing science and technology innovation team Grant (2016).

References

- Hu, Z., Wang, H., He, J., et al.: Improved control method for solar auto-tracking based on difference evolution algorithm. *Acta Energetica Solaris Sinica* **35**(6), 1016–1021 (2014)
- Mohamed, A.W.: Restart differential evolution algorithm with local search mutation for global numerical optimization. *Egypt. Inform. J.* **15**(3), 175–188 (2014)
- Qin, H., Wei, H.: A quantum-inspired approximate dynamic programming algorithm for large-scale unit commitment problems. *Proc. CSEE* **35**(19), 4918–4927 (2015)
- Xiong, W., Xu, M., Xu, B.: Differential bee colony algorithm for non-convex economic load dispatch. *Control Decis.* **26**(12), 1813–1823 (2011)
- Zhao, J., Wang, M.: Dynamic economic dispatch of microgrid based on dynamic programming. *J. Northeast Dianli Univ.* **36**(2), 19–25 (2016)
- Bin, Z., Yan, S., Jinming, L., et al.: Multiobjective optimal generation dispatch using equilibria-based multi-group synergistic searching algorithm. *Trans. China Electro Tech. Soc.* **30**(22), 181–189 (2015)
- Daqing, W., Liu, L., Jianguo, Z., et al.: Environmental economic power dispatch based on multi-objective evolution algorithm with adaptive space partition. *Control Decis.* **30**(11), 1974–1980 (2015)
- Lianghong, W., Yaolan, W., Xiaofang, Y., et al.: Fast self-adaptive differential evolution algorithm for power economic load dispatch. *Control Decis.* **28**(4), 557–562 (2013)
- Wu, X., Wang, X., Wang, J., et al.: Economic generation scheduling of a microgrid using mixed integer programming. *Proc. CSEE* **33**(28), 1–9 (2013)
- Zheng, H., Zheng, H., Yang, Y., et al.: A novel quasi-primitive composite control strategy applied in photovoltaic grid-connected inverter. *Acta Energetica Solaris Sinica* **37**(5), 1190–1196 (2016)
- Zheng, H., Yang, Y., Zheng, H., et al.: Grid frequency tracking technology of double closed-loop orthogonal vector lock strategy. *Acta Energetica Solaris Sinica* **37**(10), 2497–2504 (2016)
- Heng-Dong, X., Hao, Z., Wei, G., et al.: Active droplet sorting in microfluidics: a review. *Lab Chip* **17**, 751–771 (2017)
- Meng, A., Hu, H., Lu, H.: Crisscross optimization algorithm for large scale dynamic economic dispatch. *Power Syst.* **23**, 18–23 (2015)
- Chen, Z., Hu, Z.: A modified hybrid PSO-BBO algorithm for dynamic economic dispatch. *Power Syst. Prot. Control* **18**, 44–49 (2014)
- Zheng, X.: Multi-area economic dispatch of power system based on artificial bee colony optimization. *Comput. Eng. Sci.* **37**(8), 1533–1539 (2015)
- Basu, M.: Teaching-learning-based optimization algorithm for multi-area economic dispatch. *Energy* **68**(8), 21–28 (2014)
- Secui, D.C.: The chaotic global best artificial bee colony algorithm for the multi-area economic/emission dispatch. *Energy* **93**, 2518–2545 (2015)
- Basu, M.: Quasi-oppositional group search optimization for multi-area dynamic economic dispatch. *Electr. Power Energy Syst.* **78**, 356–367 (2016)
- Li, Z., Wu, W., Zhang, B., et al.: Decentralized multi-area dynamic economic dispatch using modified generalized benders decomposition. *Power Syst.* **31**(1), 526–538 (2016)
- Meng, A., Hu, H., Yin, H., et al.: Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects. *Energy* **93**, 2175–2190 (2015)
- Anbo, Meng, Peng, Mei, Haiming, Lu: Crisscross Optimization Algorithm for Combined Heat and Power Economic Dispatch. *J. Power System Protection and Control* **44**(6), 90–97 (2016)
- Jang, H., Dong, Yao, Wang, Jiang Zhou, et al.: Intelligent optimization models based on hard-ridge penalty and RBF for forecasting global solar radiation. *Energy Convers. Manage.* **95**, 42–58 (2015)
- Anbo, A., Longfei, Y., Xing, L., et al.: Research on reactive power optimization using quantum evolutionary algorithm based on NW small world model. *Electric Power* **48**(1), 107–114 (2015)



Zheng Hongfeng is a professor in Zhejiang Industry Polytechnic College, Shaoxing, China. He received his master degree from Huazhong university of science and technology. Since 2007, he has been a professor at Zhejiang Industry Polytechnic College. His research interests include Power Electronics and Applications, intelligent control, algorithm research.