Solving Optimal Power Flow Using Cuckoo Search Algorithm with Feedback Control and Local Search Mechanism

Gonggui Chen, *Member, IAENG,* Xingting Yi, Zhizhong Zhang*, and Shiyuan Qiu

***Abstract*—Cuckoo search (CS) algorithm is a novel heuristic algorithm, which can effectively solve the optimization problem by simulating the brood parasitism of some cuckoo species and combining with Lévy flight mechanism. However, it has also been shown to have certain weaknesses, especially falling into local optimums. Therefore, a novel CS algorithm with feedback control and local search mechanism (FLCS) is proposed in this paper. In the FLCS, feedback control is introduced to enhance the efficiency of search process, and local search mechanism is guided by the global optimal solution for improving the poor local search ability of CS method. To verify the performance of our approach, 21 test functions of different types are first employed. Then, the FLCS has been performed on the IEEE 30-bus power flow test case for optimal power flow (OPF) problem with valve point effect. The results indicate that the proposed FLCS method clearly has better performance than CS in the solution accuracy and convergence speed. In addition, the comparison results show that FLCS performs better than other evolutionary methods from literature for different functions.**

***Index Terms*—Cuckoo search algorithm, Feedback control, Local search mechanism, Test functions, Optimal power flow**

I. Introduction

MANY practical problems can be transformed into the optimization problems of searching the global optimal solution, and optimization problems are becoming more and more important in many scientific research fields [1]. However, the optimization is also becoming increasingly complicated, such as the optimal power flow (OPF) problem, which increases the difficulty of problem solving. In the past decades, a number of evolutionary algorithms, such as particle swarm optimization (PSO) [2-4], differential evolution (DE) [5, 6], ant colony optimization (ACO) [7, 8], biogeography based optimization (BBO) [9], krill herd algorithm (KHA) [10], and artificial bee colony (ABC) [11], have been proposed for handling different optimization problems.

Manuscript received September 29, 2018; revised December 21, 2018. This work was supported by Innovation Team Program of Chongqing Education Committee (CXTDX201601019), Chongqing University Innovation Team under Grant (KJTD201312) and the National Natural Science Foundation of China (Nos. 51207064 and 61463014).

Gonggui Chen is with the Key Laboratory of Network control & Intelligent Instrument, Chongqing University of Posts and Telecommunications, Ministry of Education, Chongqing 400065, China; Key Laboratory of Complex Systems and Bionic Control, Chongqing University of Posts and Telecommunications, Chongqing 400065, China(e-mail:chenggpower@126.com).

Xingting Yi is with the Key Laboratory of Complex Systems and Bionic Control, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: s1219629110@163.com).

Zhizhong Zhang is with the Key Laboratory of Communication Network and Testing Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (corresponding author, e-mail: zhangzztx@163.com).

Shiyuan Qiu is with China Mobile (Suzhou) Software Technology Co., Ltd, Suzhou 215000, China (e-mail: 2550649632 @qq.com).

In the past decades, a number of evolutionary algorithms,

Cuckoo search (CS) algorithm is a meta-heuristic optimization algorithm recently proposed by Yang [12], which is inspired by the breeding parasitic characteristics of cuckoo and combined with the Lévy flights behavior. The cuckoo will constantly evolve to reduce the possibility of the egg being abandoned to increase their survival rate. Generally, this algorithm is easy to be implemented but highly efficient. In order to improve the performance of the algorithm, several modified CS algorithm have been proposed to achieve better effect in various industries. Naik and Panda [13] proposed an adaptive cuckoo search (ACS) for face recognition. ACS method is a parameter free algorithm and can adaptively decide the step size, which is validated using 23 standard benchmark test functions and several famous face databases. Huang et al. [14] presented the novel hybrid algorithm named as teaching learning based cuckoo search to achieve high product quality, which combines the powerful search ability of Lévy flight with the fast convergence rate of teaching-learning process. Nguyen et al. [15] presented an adaptive cuckoo search algorithm (ACSA) to optimize network topology, and obtained the results on three different network systems showing the effectiveness of ACSA. For minimizing two objective functions of the short-term hydrothermal scheduling problem, Nguyen et al. [16] proposed the modified cuckoo search algorithm which applied a population classification mechanism and two modified updating methods. Li et al. [17] presented the self-adaptive cuckoo search (SACS); in which self-adaptive parameter is applied to strengthen the diversity of population and two novel search methods are used for improving the searching ability of CS method. Rakhshani et al. [18] proposed snap-drift cuckoo search (SDCS) based on the learning mechanism and information sharing strategy. Simulation results prove that SDCS has superior convergence rate and robustness through statistical analysis and comparison.

Although a number of efforts have been made to strengthen the search capability and enhance convergence rate of CS, many further work needs to be done for improving the performance of CS method. It should be

(Advance online publication: 27 May 2019)

noted that these improved algorithms mentioned above can improve CS method in a certain extent, but they cannot dynamically feedback the evolutionary search process, which is passive and mechanical. In order to remedy these weaknesses mentioned above, this paper proposes a novel cuckoo search method with feedback control and local search mechanism (FLCS) for optimization problems. On the one hand, the feedback control is able to enhance the efficiency of search process which can be regarded as a closed loop control process of population characteristics, and the population improvement rate as feedback is introduced into the CS algorithm, which can dynamically adjust the search process instead of using unalterable learning probability and thus make the reliability of the algorithm further improved. On the other hand, a local search mechanism based on the guidance of best solution is developed to increase local exploration capability. The new mutation operator can effectively balance the global search of Lévy flight mechanism. In this way, the FLCS method can obtain better search performance in the whole evolution process. To evaluate the effectiveness of FLCS, 21 well-known test functions of three different types are adopted for simulation experiments under various different simulation conditions. And FLCS method has been used for solving the OPF problem of power system with two different objective functions. The statistical comparison of results indicates that the proposed method can significantly enhance solution accuracy and the convergence rate of CS. Moreover, FLCS algorithm is quite competitive and better than most other evolution algorithms by analyzing the experimental results.

The rest of this paper is organized as follows: Section II introduces the basic CS. Our proposed FLCS is specifically illustrated in Section III. After that, in Section IV, abundant experimental tests have been carried out to verify the performance of the FLCS method. Section V demonstrates the effectiveness of FLCS though simulation studies on IEEE 30-bus power flow test case of power system. Finally, Section VI provide the conclusions.

II. Cuckoo Search Algorithm

Cuckoo search (CS) algorithm is a meta-heuristic algorithm inspired by the breeding parasitic characteristics of cuckoo and combined with the Lévy flights behavior. It is worth mentioning that the host may find that the egg is not its own with a probability $p_a \in [0, 1]$. In this case, the host will abandon the invasive egg from the nest or form a new nest. [19]. To establish the mathematic model of the CS algorithm, we define three idealized assumptions: i) a cuckoo can only lay one egg in a random nest for one time; ii) the excellent nests with high-quality eggs will be retained to the next generation; iii) the available host nests are invariant during the whole searching process.

In CS algorithm, a nest is regarded as a candidate solution. Let $X_i(t)$ denote the ith solution (for i = 1, 2,…, NP) at t generation, represented as $X_i = (x_i^1, x_i^2,\ldots, x_i^D)$ in the D-dimension problem. In the initial process of the CS algorithm, the ith component is randomly generated in a certain range, as shown in (1).

$$X_i(t) = X_{\min} + rand[D]\cdot(X_{\max} - X_{\min}) \quad (1)$$

where $rand[D]$ is a random value obeying uniform distribution in [0, 1] of D-dimension; X_{min} and X_{max} are the minimum and maximum boundaries, respectively.

When producing the next solution $X_i(t+1)$ of the ith individual, a Lévy flight can be performed as follows:

$$X_i(t+1) = X_i(t) + \alpha \oplus Lévy(\lambda) \quad (2)$$

where $\alpha > 0$ denotes the step size which should be associated with the scale of the optimized problem and usually taken as 1; the special symbol $\oplus$ denotes the entry wise multiplication. The Lévy flight follows the random walk, which can be defined according to the Lévy distribution as follows:

$$Lévy(\lambda) \sim u = t^{-\lambda}, \qquad (1 < \lambda \le 3) \quad (3)$$

This is a stochastic equation of heavy tailed probability distribution with an infinite variance [20]. The Lévy distribution is a random walk process with a heavy tail. In this form of walking, it may be short distance step and occasionally a long step. In the process of exploring a wide range of space, Lévy flight is more efficient than Brownian motion.

In the iteration process, step size is an important factor reflecting the performance of CS method, which is given as:

$$stepsize_j = 0.01\left(\frac{u_j}{v_j}\right)^{1/\lambda}\cdot(v - X_{best}) \quad (4)$$

where $u = t^{-\lambda} \times randn\,[D]$ and $v = randn\,[D]$ are taken from a normal distribution. Then, a new solution vector is computed as follows:

$$v_i = v_i + stepsize_j * randn[D] \quad (5)$$

After producing the new solution v_i, CS method will adopt the greedy strategy to select the better solution recorded as X_i according to their objective function values.

The last operation in CS algorithm can be seen as the mutation strategy by discovering a new solution, which is formulated as:

$$v_i = \begin{cases} X_i + rand\cdot(X_{r1} - X_{r2}) & , rand < p_a \\ X_i & , otherwise \end{cases} \quad (6)$$

where X_{r1} and X_{r2} are two randomly selected solutions. If the objective function of v_i is smaller than X_i, v_i is regarded as the next generation solution, otherwise X_i would remain unchanged.

III. The Improved FLCS Algorithm

A. Feedback control

In the standard CS algorithm, new solutions can evolve continuously, but this evolution is passive and mechanical. In order to adapt the algorithm proactively, it is necessary to introduce some methods of regulating evolution process. The search process of dynamic algorithm can be regarded as a closed loop control process, in which the population characteristics are the feedback quantity, the expected population characteristics are the reference quantity, and the evolutionary algorithm is the control strategy.

For improving the convergence speed and optimization efficiency, Rechenberg proposed the famous 1/5 success theorem [21]. Generally, the objective value is improved 1 times in the 5 variation. Therefore, the control parameters of the algorithm should be dynamically adjusted with the success rate of new solutions, and the improvement rate is remained at 1/5. In the generation mechanism of the CS solution, there are two parameters controlling the characteristics of the offspring population: the step-size factor (α) and the discovery factor (p_a). In summary, the improved rate can be chosen as feedback and 0.2 is the expected value; additionally, the step-size factor and the discovery factor are the control variables.

For the step size factor, according to the principle, there are three adjustment strategies: i) The improvement rate is greater than 0.2, which shows that the searching space is relatively smooth, and the algorithm can find a better solution with a larger probability. In order to enhance search efficiency and reduce the number of calculation of objective function, the step size should be properly increased. ii) The improvement rate is less than 0.2. It indicates that the searching space is more complex in such a situation and the possibility of discovering a better solution is relatively low. We should reduce the step size appropriately to strengthen the exploration in searching space. iii) The improvement rate exactly equals 0.2, which shows that the current step size is just to make the population improvement rate at the best value, and do not need to be adjusted. However, the probability that the improvement rate exactly equals 0.2 is very slim, which makes the parameters to be changed frequently in a larger range. For the stability of the dynamic parameters, we will change the improvement rate remained at 0.2 to the interval [0.2, 0.4]. The step size based on this principle can be described as:

$$\alpha(t+1)=\begin{cases}\alpha(t)\cdot f_\alpha, & R>0.4\\ \alpha(t), & \text{else}\\ \alpha(t)/f_\alpha, & R<0.2\end{cases} \tag{7}$$

where R is the improvement rate of new solutions, f_α is learning factor of step size. Similarly, the discovery probability is modified as follows:

$$p_a(t+1)=\begin{cases}p_a(t)\cdot f_p, & R>0.4\\ p_a(t), & \text{else}\\ p_a(t)/f_p, & R<0.2\end{cases} \tag{8}$$

where f_p is the learning factor of discovery probability. In addition, it should be noted that the upper and lower limits should be determined before the start of the operation to prevent the overshoot of the parameter.

B. Local search mechanism

The basic principle of CS algorithm is to generate step length by Lévy flight, which helps to avoid falling into local optimum and achieve excellent global search capability. However, Lévy flight cannot make full use of the information of the local area so that the local search ability is poor, which is mainly due to the highly random jumps. For enhancing local search ability, and accelerating the convergence of CS method, a novel mutation mechanism is presented based on the guidance of best solution. It is worth noting that this local search mechanism is inspired by the cognitive learning mechanism of PSO method. In short, the new solution vector of every mutant according to the position of the best vectors achieved so far by the whole population of a particular generation by following the same direction of the best. The modified crossover operator is as follows:

$$v_i=\begin{cases}X_i+R_1\left(X_{r1}-X_{r2}\right)+R_2\left(X_{gb}-X_i\right), & rand<p_a\\ X_i, & otherwise\end{cases} \tag{9}$$

where R_1 and R_2 are two uniform random variables between 0 and 1; X_{r1} and X_{r2} are two randomly chosen solutions and X_{gb} is the best optimal solution in the entire population. Obviously, the guidance of the best solution is added to the original random mutation mechanism, which can make the new mutant vector learn to the optimal solution. As a result, the local exploration ability of CS algorithm is enhanced. The new mutation operator can effectively balance the global search of Lévy flight mechanism. Therefore, the search performance of FLCS will be better than the basic CS method. In fact, the combination of feedback control and local search mechanism can utilize the advantages of both local optimization and global optimization in the search process.

FLCS procedure

Begin
  Initialize the function evolution numbers FEs=1
  Generate a random population X_i, i=1, 2, …, NP, and set the initial parameters
  Evaluate fitness (F) for every individual and find the best individual X_{gb}
  While $FEs < MaxFEs$ **do**
    for i=1 to NP **do**
      $v_i = v_i + stepsize_j * randn[D]$
    end
    for i=1 to NP **do**
      for j=1 to D **do**
        if $rand < p_a$ **then**
        $v_{i,j} = X_{i,j} + R_1\left(X_{r1,j} - X_{r2,j}\right) + R_2\left(X_{gb,j} - X_{i,j}\right)$
        end if
      end for
    end for
    for i=1 to NP **do**
      Evaluate the fitness of v_i
      if $F(v_i) < F(X_i)$ **then**
        Replace X_i with v_i
        The number of improved individuals increase 1
      end if
    end for
    Calculate population improvement rate R
    Perform the feedback control mechanism to update the α and p_a
    Update the best individual X_{best}
    FEs=FEs+1
  End while
End

C. Bound handling mechanism

In the standard CS algorithm, the control variables of solution vector change within a certain constraint. If the value of some control variables exceeds the constraint limit during the iteration process, CS algorithm will make these variables equal to their corresponding boundary values. However, this control mechanism may make the value of many control variables equal to the critical value. The drawback is that the population diversity will reduce and the evolution may fall into the local optimum, if there are local optimal solutions at the boundary. In order to overcome the defect, a new bound handling mechanism is presented as follows:

$$X_i = \begin{cases} L_i + F_1 \cdot (L_i - X_i) & \text{if } X_i < L_i \\ U_i - F_2 \cdot (X_i - U_i) & \text{if } X_i > U_i \\ X_i & \text{otherwise} \end{cases} \tag{10}$$

where L_i and U_i indicate the lower and upper critical values of the *i*th dimension; F_1 and F_2 are two random variables between 0 and 1. According to this mechanism, each of any solution is examined for the feasibility. If there is violation constraints, the solution will reset the feasible values for variables. It is worth mentioning that, the solution vector will not change overall, only change the control variables that violate their constraints [22].

By combining the feedback control and the local search mechanism, the pseudo code of FLCS is developed and shown in "FLCS procedure".

TABLE I
DESCRIPTION OF THE UNIMODAL BENCHMARK TEST FUNCTIONS WITH $D = 30$

Name	Functions	Range of search	Optimum
Sphere	$f_1(X) = \sum_{i=0}^{D} x_i^2$	$[-100, 100]^D$	$f_1(\vec{0}) = 0$
Schwelfel_2.22	$f_2(X) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$	$f_2(\vec{0}) = 0$
Schwelfel_1.2	$f_3(X) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$	$f_3(\vec{0}) = 0$
Schwelfel_2.21	$f_4(X) = \max_i \{\lvert x_i \rvert, 1 \le i \le D\}$	$[-100, 100]^D$	$f_4(\vec{1}) = 0$
Rosenbrock	$f_5(X) = \sum_{i=1}^{D-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-30, 30]^D$	$f_5(\vec{0}) = 0$
Step	$f_6(X) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]^D$	$f_6(\vec{0}) = 0$
Quartic	$f_7(X) = \sum_{i=1}^{D} i x_i^4 + random[0,1)$	$[-1.28, 1.28]^D$	$f_7(\vec{0}) = 0$
SumSquare	$f_8(X) = \sum_{i=1}^{D} i x_i^2$	$[-10, 10]^D$	$f_8(\vec{0}) = 0$
Elliptic	$f_9(X) = \sum_{i=1}^{D} (10^6)^{[(i-1)/(D-1)]} x_i^2$	$[-100, 100]^D$	$f_9(\vec{0}) = 0$

TABLE II
DESCRIPTION OF THE MULTIMODAL BENCHMARK TEST FUNCTIONS WITH $D = 30$

Name	Functions	Range of search	Optimum
Rastrigin	$f_{10}(X) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$[-5.12, 5.12]^D$	$f_{10}(\vec{0}) = 0$
Ackley	$f_{11}(X) = -20\exp\left(-0.2\sqrt{D^{-1}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(D^{-1}\sum_{i=1}^{D}\cos 2\pi x_i\right) + 20 + e$	$[-32, 32]^D$	$f_{11}(\vec{0}) = 0$
Griewank	$f_{12}(X) = \sum_{i=1}^{D} x_i^2 / 4000 - \prod_{i=1}^{D} \cos\left(x_i / \sqrt{i}\right) + 1$	$[-600, 600]^D$	$f_{12}(\vec{0}) = 0$
Penalized_1	$f_{13}(X) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ where, $y_i = 1 + ((x_i + 1)/4)$, and $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$	$f_{13}(\vec{1}) = 0$
Penalized_2	$f_{14}(X) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_D - 1)^2\left[1 + \sin^2(3\pi x_D)\right]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$ where, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$	$f_{14}(\vec{1}) = 0$
Alpine	$f_{15}(X) = \sum_{i=1}^{D} \lvert x_i \sin x_i + 0.1 x_i \rvert$	$[-10, 10]^D$	$f_{15}(\vec{0}) = 0$

IV. RESULTS OF TEST FUNCTIONS

A. *Simulation setup*

In order to examine the performance of FLCS, 21 widely used test functions displayed in Tables 1-3 are applied in this test. TABLE I consists of 9 unimodal test functions, which have only one peak in the search range. In TABLE II, f_{10}-f_{15} are multimodal functions, which have a lot of local optimal values. However, the optimal value of the test function is only one. And f_{16}-f_{21} in TABLE III are multimodal test functions with fixed dimension. The specific parameter values of those functions can be obtained from [13].

In this simulation experiment, the population size *NP* is set to 30. The maximum function evolution numbers (*MaxFEs*) are set as: 150,000 for f_1, f_6, f_8-f_9, f_{11}, f_{13}-f_{15}; 200,000 for f_2, f_{12}; 300,000 for f_7, f_{10}; 500,000 for f_3-f_5 and 10,000 for f_{16}-f_{21}. In order to ensure the reliability of the experimental results, all cases have been run 30 times independently.

In order to analyze and compare experimental results, we have counted four indicators, which are the Best, Worst, Mean and standard deviation (Std.) of test function. The 'Best' and 'Worst' indicate the minimum and maximum value by 30 independent run, respectively. The 'Mean' and ' Std.' indicate the average value and the standard deviation of 30 experimental results.

B. Comparison between CS and FLCS

To evaluate the performance of our proposed FLCS in optimization problem, we compare the basic CS and FLCS methods based on the test functions. TABLE IV presents the experimental results of the test functions given in TABLE I-II, and TABLE V reports the experimental results of the multimodal functions with fixed dimension in TABLE III .

TABLE III
DESCRIPTION OF MULTIMODAL BENCHMARK TEST FUNCTIONS WITH FIXED DIMENSION

Name	Functions	Range of search	Optimum
Six-Hump Camel-Back	$f_{16}(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	$[-5, 5]^2$	$f_{16}(0.08983,-0.7126)$ *or* $f_{16}(-0.08983,0.7126)=-1.03163$
Branin	$f_{17}(X)=\left(x_2-\left(5.1/4\pi^2\right)x_1^2+\left(5/\pi\right)x_1-6\right)^2+10\left(1-1/8\pi\right)\cos x_1+10$	$-5\le x_1\le 10$, $0\le x_2\le 15$	$f_{17}(-3.142,12.275)$ or $f_{17}(3.142,2.275)$ or $f_{17}(9.425,2.425)=0.398$
Goldstein-Price	$f_{18}(X)=\left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times\left[(2x_1-3x_2)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)+30\right]$	$[-2, 2]^2$	$f_{18}(0,-1)=3$
Hartman 3	$f_{19}(X)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_i-p_{ij}\right)^2\right)$	$[0, 1]^3$	$f_{19}(0.114,0.556,0.852)=-3.86$
Hartman 6	$f_{20}(X)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}\left(x_i-p_{ij}\right)^2\right)$	$[0, 1]^6$	$f_{20}(0.201,0.15,0.447,0.275,0.311,0.657)=-3.32$
Shekel's Family	$f_{21}(X)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	$[0, 10]^4$	$f_{21}(\vec{X})=-10.5363$

TABLE IV
PERFORMANCE EVALUATION OF THE BENCHMARK TEST FUNCTIONS DESCRIBED IN TABLE I-II

F	CS				FLCS			
	Best	Worst	Mean	Std.	Best	Worst	Mean	Std.
f_1	3.6522E-18	1.9525E-17	8.2311E-18	3.6290E-18	**1.8614E-142**	**5.9974E-140**	**2.1225E-140**	**3.2832E-140**
f_2	1.9644E-14	7.4346E-13	4.1155E-13	1.3154E-13	**2.9216E-114**	**6.0845E-113**	**3.2631E-113**	**2.4877E-113**
f_3	8.5753E-08	9.8316E-07	4.0050E-08	3.3838E-08	**1.1024E-77**	**5.3656E-74**	**8.9041E-75**	**2.8186E-74**
f_4	8.0841E-05	6.7016E-04	3.2216E-04	3.3732E-04	**4.4439E-28**	**9.6643E-26**	**5.7715E-26**	**4.2416E-26**
f_5	2.3653E-05	7.5339E-04	3.4434E-04	5.0920E-04	**3.3967E-09**	**4.0037E-08**	**1.0717E-08**	**2.7608E-08**
f_6	8.9030E-13	5.1479E-12	3.0285E-12	1.1458E-12	**9.2479E-33**	**6.4116E-32**	**2.5157E-32**	**5.4152E-33**
f_7	1.5714E-02	7.3130E-02	3.5814E-02	3.5590E-02	**1.8321E-03**	**7.6432E-03**	**2.7256E-03**	**2.4214E-03**
f_8	6.2703E-14	6.6017E-13	1.9874E-13	1.3020E-13	**1.7045E-144**	**7.3842E-142**	**2.8979E-142**	**1.3265E-142**
f_9	7.4464E-10	8.1306E-09	2.8488E-10	4.1567E-10	**1.3296E-135**	**3.9153E-131**	**1.4928E-132**	**2.0269E-132**
f_{10}	**2.0900E+01**	**3.5024E+01**	**2.6298E+01**	**7.5098E+00**	3.6025E+01	5.3871E+01	3.2326E+01	6.9312E+00
f_{11}	4.1185E-02	5.9821E-01	2.1726E-01	8.3251E-02	**7.9386E-15**	**8.9962E-14**	**8.3926E-15**	**2.3973E-15**
f_{12}	6.9441E-06	7.1556E-02	2.6952E-02	3.4441E-02	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
f_{13}	2.3108E-10	8.7666E-09	1.3017E-09	1.8781E-09	**1.5705E-32**	**1.5705E-32**	**1.5705E-32**	**1.5458E-34**
f_{14}	1.7168E-07	4.2237E-07	2.5127E-07	2.5657E-07	**1.5705E-32**	**8.5859E-31**	**3.7125E-32**	**2.3846E-32**
f_{15}	5.8723E-03	9.1539E-02	1.5179E-03	3.3989E-03	**5.3453E-16**	**8.7025E-14**	**2.7816E-15**	**5.4855E-15**

TABLE V
PERFORMANCE EVALUATION OF THE BENCHMARK TEST FUNCTIONS DESCRIBED IN TABLE III

F	CS				FLCS			
	Best	Worst	Mean	Std.	Best	Worst	Mean	Std.
f_{16}	-1.0316	-1.0316	-1.0316	7.3536E-06	**-1.0316**	**-1.0316**	**-1.0316**	**6.0809E-16**
f_{17}	0.3979	0.3980	0.3979	2.6376E-05	**0.3979**	**0.3979**	**0.3979**	**5.2537E-13**
f_{18}	3.0000	3.0003	3.0001	8.1270E-05	**3.0000**	**3.0000**	**3.0000**	**1.9943E-15**
f_{19}	-3.8628	-3.8628	-3.8628	3.4805E-06	**-3.8628**	**-3.8628**	**-3.8628**	**1.2691E-15**
f_{20}	-3.3219	-3.3040	-3.3148	1.2387E-02	**-3.3220**	**-3.3220**	**-3.3220**	**5.9386E-11**
f_{21}	-10.5327	-10.4791	-10.5049	9.0847E-03	**-10.5364**	**-10.5364**	**-10.5364**	**2.3387E-12**

We can see that FLCS outperforms CS in terms of the solution quality in 20 out of 21 functions. Only the test results of f_{10} have no obvious difference between FLCS and CS, but FLCS is able to achieve a more reliable solution. Additionally, it can be seen that FLCS and CS have no significant difference for the fix low-dimensional functions in TABLE V.

In order to show the evolutionary process of FLCS more directly, some convergent curves of representative test functions are plotted and shown in Figs. 1-2. It can be seen that FLCS method converges faster than CS on these unimodal and multimodal test functions. However, the advantage is not obvious on these functions with fixed dimension. From the comparative analysis, we could conclude that FLCS algorithm significantly outperforms CS method not only in solution quality but also in convergence rate.

C. Comparison with other algorithms

The FLCS algorithm is compared with seven other well-known evolutionary algorithms in Table 6 on functions f_1–f_7 and f_{10}–f_{14}. These algorithms include GL-25 [23], SaDE [5], CDE [24], MOBBO [25], MLBBO [26], OLCS [27] and HLXDE [28]. The results of GL-25 and SaDE are obtained from [29]. In order to make a rigorous comparison,

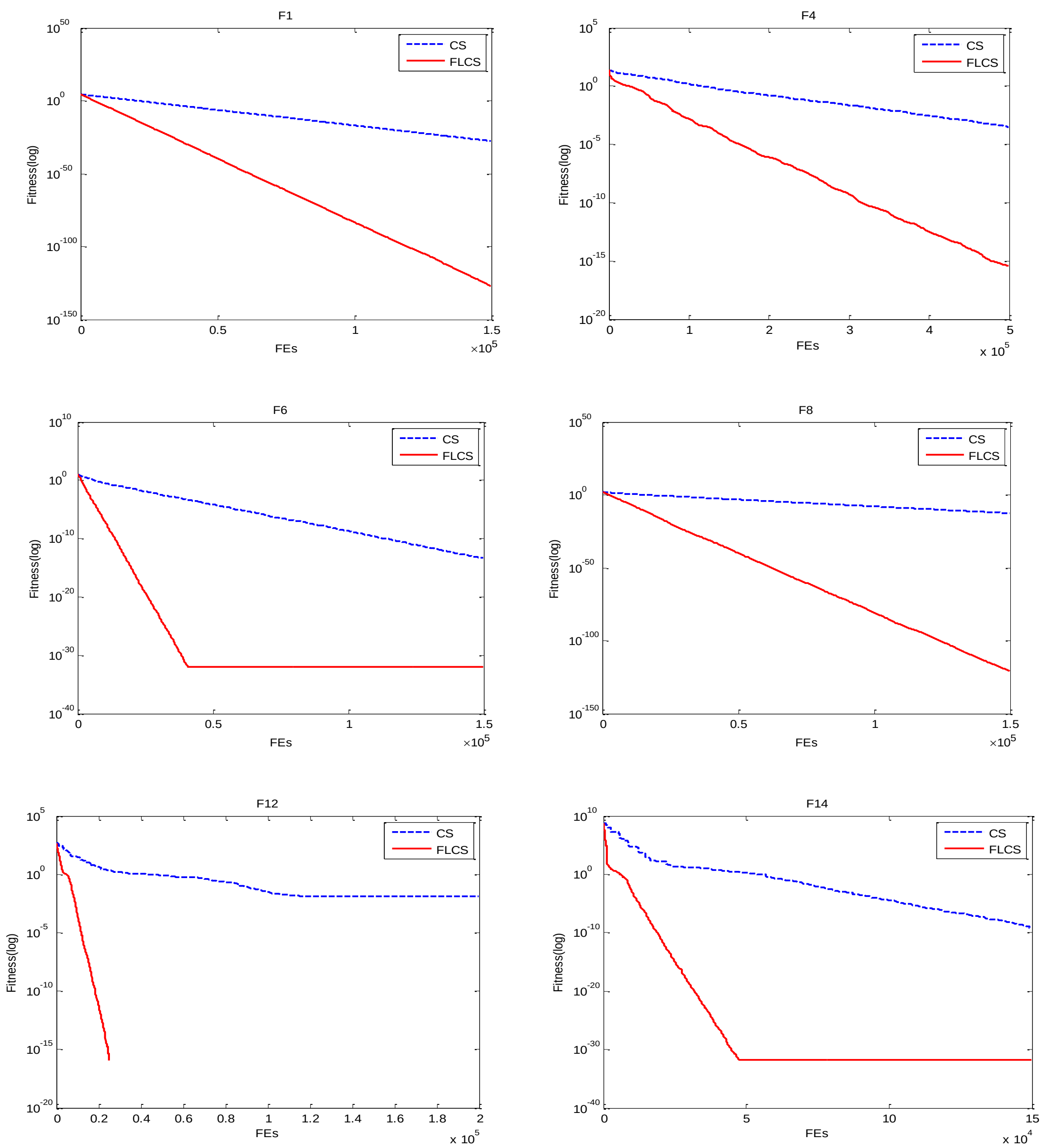

Fig. 1. Convergence graphs of CS and FLCS for six representative test functions

for GL-25, SaDE, CDE, MOBBO, MLBBO, OLCS and HLXDE methods, all the parameters are set as the same used in their original literature. All algorithms are carried out at D = 30 and NP =100, and the comparison results are presented in TABLE VI.

From the results of 12 functions, it can be seen that FLCS method obtains the optimal solutions on 7 test functions. Especially, FLCS algorithm achieves better results than SaDE on all compared functions. GL-25 gets the optimal result on f_1 among the 8 methods; CDE algorithm can obtain the best results on two functions f_5 and f_{11}. Compared with MOBBO, FLCS method wins on 12 functions except on f_6, for which both algorithms obtain the same results. For MOBBO and OLCS, each one only performs better than FLCS on two functions. Although LXDE outperforms FLCS on two functions and gets the equivalent results on two other functions, FLCS achieves better performance on eight functions. Summarizing the above statements, we can draw a conclusion that FLCS is highly competitive to the above-mentioned well-known evolutionary algorithms, which has superior search ability to find the optimal solution.

TABLE VI
COMPARISON OF FLCS WITH SOME STATE-OF-THE-ART ALGORITHMS (*NP* =100, *D*=30)

F		GL-25	SaDE	CDE	MOBBO	MLBBO	OLCS	HLXDE	FLCS
f_1	Mean	**2.87E-120**	1.48E-18	1.07E-28	1.70E-09	2.11E−31	1.19E-07	4.66E-43	7.71E-54
	Std.	**7.31E-120**	9.28E-19	7.65E-29	9.30E-09	1.25E−31	1.89E-07	8.83E-43	5.23E-54
f_2	Mean	2.56E-38	3.16E-15	4.21E-21	2.74E-05	1.58E−21	3.27E-07	1.56E-29	**8.95E-39**
	Std.	9.90E-38	1.34E-15	1.85E-21	1.48E-04	7.25E−22	2.35E-07	1.44E-29	**6.43E-39**
f_3	Mean	2.47E-01	4.02E-20	1.64E-34	3.25E-02	1.42E−20	5.62E-08	4.38E-32	**2.39E-50**
	Std.	4.82E-01	4.89E-20	9.18E-34	4.50E-02	1.70E−20	5.92E-08	1.16E-31	**3.47E-50**
f_4	Mean	4.04E-02	8.01E-10	6.48E-22	2.43E−02	5.28E−08	7.13E-07	2.99E-16	**4.28E-32**
	Std.	2.46E-02	3.49E-10	1.18E-21	6.74E−03	1.02E−07	3.50E-07	4.23E-16	**2.85E-32**
f_5	Mean	2.12E+01	7.97E-02	**0.00E+00**	6.32E +01	5.34E−21	1.62E +01	6.64E-01	3.09E-11
	Std.	7.93E-01	5.64E-01	**0.00E+00**	8.33E +01	1.10E−20	4.51E-01	1.51E+00	1.52E-11
f_6	Mean	1.86E-31	3.01E+04	1.87E+04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std.	3.53E-31	1.07E+03	1.05E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	Mean	1.80E-03	6.21E-03	1.27E-03	9.79E-04	2.33E−03	**2.23E-04**	3.74E-03	1.20E-03
	Std.	5.64E-04	1.42E-03	7.37E-04	8.00E-04	1.04E−03	**1.52E-04**	1.04E-03	3.71E-04
f_{10}	Mean	2.47E+01	2.49E+05	2.17E+05	5.45E-05	**0.00E+00**	8.88E-16	**0.00E+00**	5.56E+01
	Std.	6.49E+00	5.48E+03	4.92E+03	2.99E-04	**0.00E+00**	1.72E-15	**0.00E+00**	2.27E+01
f_{11}	Mean	1.03E-12	3.08E-10	**5.28E-15**	9.74E-08	6.10E−15	1.69E-04	6.15E-15	7.99E-15
	Std.	3.05E-12	8.70E-11	**1.67E-15**	5.97E-09	6.49E−16	1.14E-04	1.79E-15	3.15E-16
f_{12}	Mean	1.24E-08	1.39E+05	9.42E+04	5.08E-03	1.73E−03	7.54E-10	1.80E-03	**0.00E+00**
	Std.	3.91E-08	3.66E+03	2.46E+03	1.05E-02	4.00E−03	5.33E-10	4.74E-03	**0.00E+00**
f_{13}	Mean	1.03E-02	4.48E-20	1.79E-30	3.46E-03	2.35E−32	1.02E-02	3.45E-03	**1.57E-32**
	Std.	3.27E-02	3.10E-20	1.50E-30	1.89E-02	3.98E−32	3.90E-03	1.89E-02	**3.63E-47**
f_{14}	Mean	4.20E-03	4.83E-18	9.42E-29	6.33E-13	5.64E−32	5.17E-02	**1.35E-32**	**1.35E-32**
	Std.	7.40E-03	3.96E-18	8.40E-29	3.44E-12	3.38E−32	1.10E-02	**0.00E+00**	**4.52E-48**

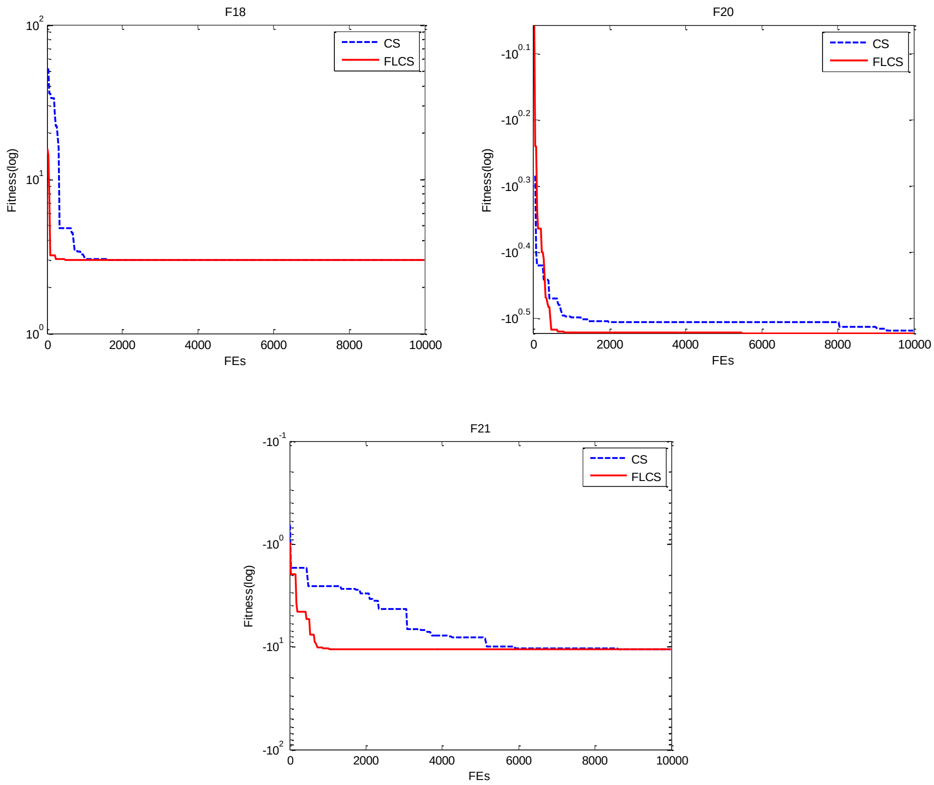

Fig. 2. Convergence graphs of CS and FLCS for three representative test functions

V. OPTIMAL POWER FLOW

In power systems, optimal power flow (OPF) is an important tool for planning and operation of the whole system. The primary purpose of OPF is to make the chosen goal to achieve the optimal state by adjusting the available control variables, meanwhile satisfy all the operating constraints [30, 31]. The objective function can be fuel cost, power losses, voltage deviations and so on, among which fuel cost is generally considered to be the most basic and important objective function. OPF is a multi-variable, multi-constraint, and nonlinear optimization problem, which is widely used for planning and controlling of the electric system and has important research significance.

A. Problem formulation

The mathematical model of OPF problem is mainly composed of the objective functions and various constraints. In this paper, this objective is to minimize the fuel cost of all generating units which can be described as:

$$\min\ f_1(x) = \sum_{i=1}^{NG} a_i + b_i P_{Gi} + c_i P_{Gi}^2 \tag{11}$$

$$x^T = [P_{G2} \cdots P_{GNG}, V_{G1} \cdots V_{GNG}, T_1 \cdots T_{NT}, Q_{C1} \cdots Q_{CNC}] \tag{12}$$

where f_1 is the fuel cost for all generators; x is a column vector of control variables; P_{Gi} is the active power of the ith generator; a_i, b_i and c_i are fuel cost coefficients of the ith generator which can be obtained from [32]; V_G, T and Q_C respectively represent generator voltage, transformer ratio and reactive power compensation; NG, NT and NC are the number of generators, transformer branches and shunt compensators, respectively.

However, the fuel cost curve has non-derivable points with considering valve point effects of generator in practical application for better reflecting the real situation. Thus the OPF optimization problem becomes a non-convex complex problem and the value of fuel cost increased, which cannot be solved by traditional optimization methods. When considering valve-point effect, the quadratic cost functions at buses 1 and 2 should be added a rectifying sinusoidal section as follows:

$$f_i = a_i + b_i P_{Gi} + c_i P_{Gi}^2 + \left| d_i \sin\left(e_i \left(P_{Gi}^{\min} - P_{Gi}\right)\right)\right| \tag{13}$$

where d_i and e_i indicate the parameters of the ith generator related to the valve point effect. Therefore, the total function considered valve point effect can be described as:

$$\begin{aligned} f_2 = &\left(\sum_{i=1}^{2} a_i + b_i P_{Gi} + c_i P_{Gi}^2 + \left| d_i \sin\left(e_i \left(P_{Gi}^{\min} - P_{Gi}\right)\right)\right|\right) \\ &+ \left(\sum_{i=3}^{NG} a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) \end{aligned} \tag{14}$$

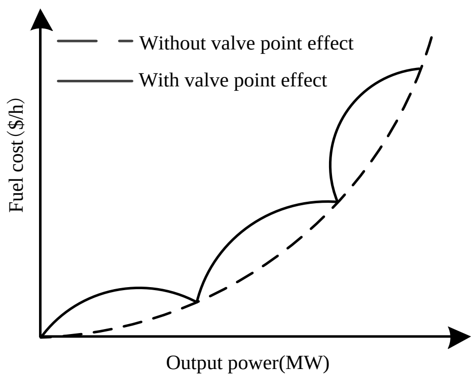

Fig. 3. Fuel cost function with and without valve point effect

The fuel cost function with and without considering the valve-point effect is depicted in Fig. 3. It can be seen that the optimization problem becomes more difficult to solve when the simulation is more accurate, and the fuel cost will be larger than that without valve point effect. Compared with the traditional algorithm, the intelligent evolutionary algorithms can better solve these optimization problems because the fuel cost curve is not continuous differentiable.

In addition, OPF is a large-scale nonlinear problem requiring satisfying various constraint conditions, which include equal constraints and unequal constraints. The equal constraints of OPF problem are a set of power flow equations of active power and reactive power. The unequal constraints represent various operating constraints on system variables which can be taken from [32].

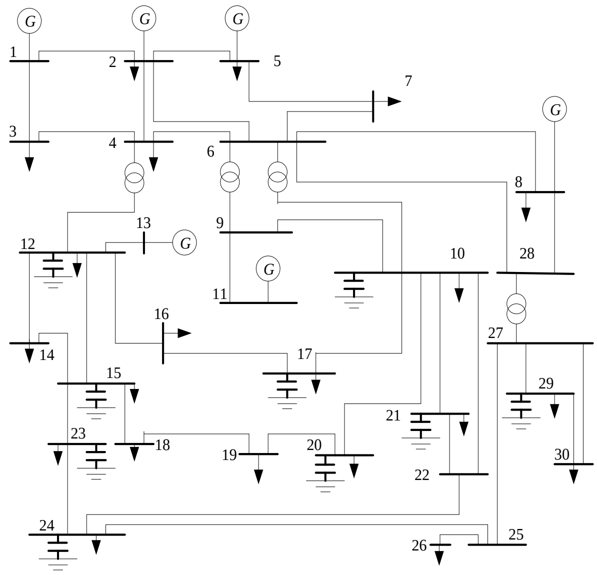

Fig. 4. System structure diagram of IEEE 30-bus system

TABLE VII
THE COST COEFFICIENTS FOR BASIC FUEL COST FUNCTION

Bus	Cost coefficients		
	a	b	c
1	0.00	2.00	0.00375
2	0.00	1.75	0.01750
5	0.00	1.00	0.06250
8	0.00	3.25	0.00834
11	0.00	3.00	0.02500
13	0.00	3.00	0.02500

TABLE VIII
THE COST COEFFICIENTS WITH VALVE POINT EFFECT

Bus	Cost coefficients				
	a	b	c	d	e
1	150.00	2.00	0.0016	50.00	0.0630
2	25.00	2.50	0.0100	40.00	0.0980

B. Experimental results of OPF problem

In this paper, the program is compiled in MATLAB R2014a environment, and simulation experiment on IEEE 30-bus system is carried out to test the effectiveness of FLCS method for OPF problem. The maximum iteration numbers are all set as 200 for the two different objective functions, and the population size NP is set as 30 in this simulation experiment. For testing the robustness of the proposed method, both of CS and FLCS algorithm perform 30 independent runs for solving the OPF problem. The system structure diagram of IEEE 30-bus system can be seen in Fig 4.

This system includes 4 transformers, 9 reactive power compensation device and 41 branches. The 9 shunt capacitors are connected at buses 10, 12, 15, 17, 20, 21, 23, 24 and 29; the 4 transformers are at lines 6–9, 6–10, 4–12 and 28–27 [14]. In addition, the system has 6 generators at buses 1, 2, 5, 8, 11 and 13, where bus 1 indicates the slack bus. The generator fuel cost coefficients of 30-bus system are given in TABLE VII-VIII. It should be explained that the cost coefficients are different at buses 1 and 2 when valve loading effects are considered, however the cost coefficients at other buses are unchanged. The maximum and minimum values for the control variables can be seen in TABLE IX.

TABLE IX
THE LIMITS OF THE CONTROL VARIABLES

Variables	Min	Max	Variables	Min	Max
P_1(MW)	50	200	T_{12}(p.u.)	0.90	1.1
P_2(MW)	20	80	T_{15}(p.u.)	0.90	1.1
P_5(MW)	15	50	T_{36}(p.u.)	0.90	1.1
P_8(MW)	10	30	Q_{C10}(p.u.)	0.00	0.05
P_{11}(MW)	12	40	Q_{C12}(p.u.)	0.00	0.05
P_{13}(MW)	0.95	1.1	Q_{C15}(p.u.)	0.00	0.05
V_1(p.u.)	0.95	1.1	Q_{C17}(p.u.)	0.00	0.05
V_2(p.u.)	0.95	1.1	Q_{C20}(p.u.)	0.00	0.05
V_5(p.u.)	0.95	1.1	Q_{C21}(p.u.)	0.00	0.05
V_8(p.u.)	0.95	1.1	Q_{C23}(p.u.)	0.00	0.05
V_{11}(p.u.)	0.95	1.1	Q_{C24}(p.u.)	0.00	0.05
V_{13}(p.u.)	0.95	1.1	Q_{C29}(p.u.)	0.00	0.05
T_{11}(p.u.)	0.90	1.1			

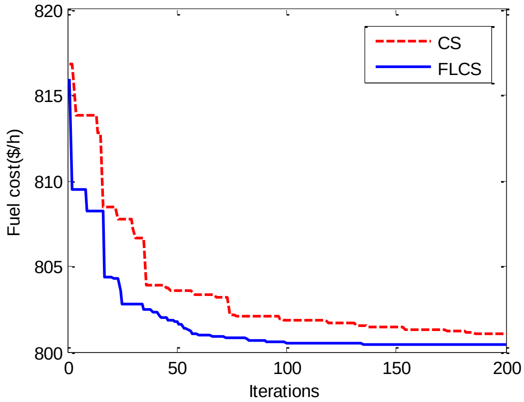

Fig. 5. Convergence curve without valve point effect of CS and FLCS

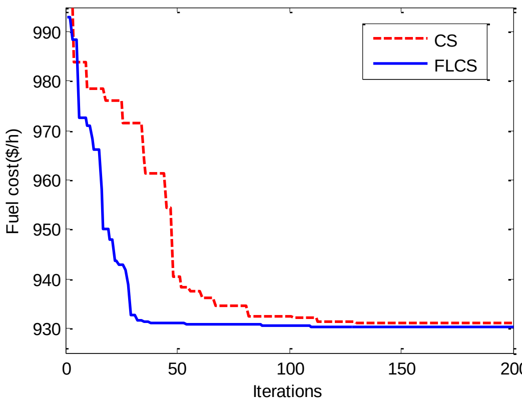

Fig. 6. Convergence curve with valve point effect of CS and FLCS

The optimal results of CS and FLCS algorithm are given in TABLE X. As seen in TABLE X, the minimum fuel cost without considering valve point effect is 800.4233$/h obtained from the FLCS method, which is better than the basic CS algorithm and other two algorithms. When the chosen objective function is fuel cost minimization with valve-point effect, the optimal result by the FLCS algorithm is 930.1713$/h, which is also the best solution among the four algorithms in Table 10. In addition, the convergence curves of the fuel cost function for two test cases are shown in Figs 5-6.

TABLE X
SIMULATION RESULTS FOR IEEE 30-BUS SYSTEM

Control variables	Basic fuel cost function		Fuel cost function with valve point effect	
	FLCS	CS	FLCS	CS
P_1(MW)	177.4957	175.2733	199.5891	196.9777
P_2(MW)	48.59783	48.99473	50.79088	51.7458
P_5(MW)	21.35871	20.54202	15.00002	15.0000
P_8(MW)	20.94908	22.50384	10.00027	10.0000
P_{11}(MW)	11.99756	12.71854	10.00000	10.0000
P_{13}(MW)	12.02936	12.39165	12.00000	12.0000
V_1(p.u.)	1.0822	1.0851	1.0289	1.0384
V_2(p.u.)	1.0637	1.0640	1.0065	1.0097
V_5(p.u.)	1.0323	1.0325	1.0266	0.9507
V_8(p.u.)	1.0372	1.0397	0.9834	0.9675
V_{11}(p.u.)	1.0939	1.0446	0.9503	1.0958
V_{13}(p.u.)	1.0470	1.0132	1.0994	1.0412
T_{11}(p.u.)	1.0600	1.0500	1.0000	1.0400
T_{12}(p.u.)	0.9100	0.9200	1.1000	0.9200
T_{15}(p.u.)	0.9700	0.9800	0.9400	0.9800
T_{36}(p.u.)	0.9700	1.0100	0.9600	0.9800
Q_{C10}(p.u.)	0.0070	.0.0270	0.0370	0.0055
Q_{C12}(p.u.)	0.0160	0.0330	0.0500	0.0000
Q_{C15}(p.u.)	0.0300	0.0160	0.0050	0.0000
Q_{C17}(p.u.)	0.0500	0.0460	0.0180	0.0320
Q_{C20}(p.u.)	0.0350	0.0230	0.0450	0.0070
Q_{C21}(p.u.)	0.0500	0.0170	0.0040	0.0390
Q_{C23}(p.u.)	0.0320	0.0450	0.0200	0.0320
Q_{C24}(p.u.)	0.0490	0.0420	0.0010	0.0110
Q_{C29}(p.u.)	0.0220	0.0370	0.0500	0.0500
Fuel cost	**800.4233**	800.9886	**930.1713**	931.1146

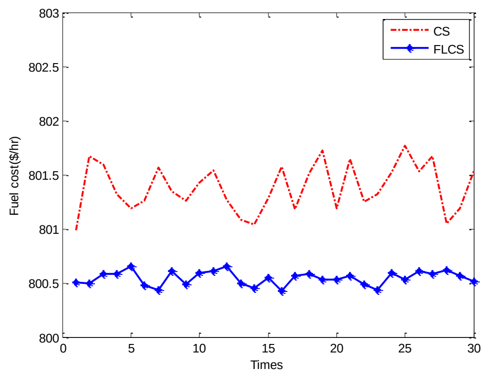

Fig. 7. Comparative distribution of the values without valve point effect

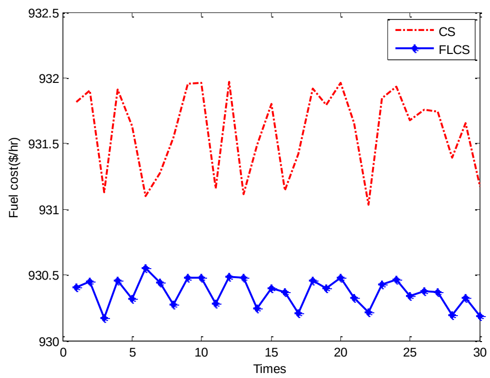

Fig. 8. Comparative distribution of the values with valve point effect

Comparative distribution of the values of basic fuel cost function and fuel cost function with valve point effect are shown in Fig.7 and Fig.8, respectively, which manifests that the result uniformity of the proposed FLCS is better compared with CS.

Furthermore, the statistical results of average, best, worst objective values and standard deviation from the 30 independent trials obtained by the different methods are illustrated in TABLE XI. The Standard deviation, which has been compared with the MSA [30] and EADPSO [33] methods reported in the literature make it clear that the distribution of the results of FLCS was more concentrated in a smaller range than that of CS, MSA and EADPSO. Moreover, statistical analysis is made to check the robustness of the proposed FLCS and other methods.

All analysis and comparison show that the modified FLCS is suitable for OPF control. Its performance is better than that for original CS algorithm.

TABLE XI
COMPARISON FOR OPTIMIZED OBJECTIVES

Algorithms	Basic fuel cost function					Fuel cost function with valve point effect				
	Trials	Best	Worst	Mean	Std.	Trials	Best	Worst	Mean	Std.
FLCS	**30**	**800.4233**	**800.6515**	**800.5431**	**0.0626**	**30**	**930.1713**	**930.5488**	**930.3688**	**0.1047**
CS	30	800.9886	801.7661	801.3836	0.2180	30	931.1146	931.9706	931.5957	0.3156
MSA [30]	30	800.5099	-	-	-	30	930.7441	-	-	-
EADPSO	50	800.2276	800.3274	800.2625	0.0303	50	930.7454	931.1094	930.8800	0.0926

VI. CONCLUSION

In this paper, the novel cuckoo search algorithm based on the feedback control and local search mechanism (FLCS) is proposed for solving the global optimization problem. In the FLCS algorithm, the presented feedback control mechanism can enhance the efficiency of searching process, which can be modeled as a closed loop control process of population characteristics. In addition, in order to enhance the local tendency and accelerate the convergence of CS, a novel mutation mechanism is presented based on the guidance of best solution, which can effectively balance the global search of Lévy flight mechanism. Moreover, the bound handling mechanism can prevent the solution out of bounds or too many solutions on the boundary. In order to examine the performance of FLCS method, 21 widely used test functions are applied in the simulation experiment. A variety of test cases are carried out based on 21 test functions. Moreover, the FLCS method has been applied to OPF problem of power system for proving its effectiveness and applicability. The experimental results indicate that FLCS algorithm clearly outperforms the basic CS not only in solution quality but also in convergence speed. Moreover, the comparison results also demonstrate that FLCS algorithm is highly competitive to those well-known evolutionary algorithms and obtains the optimal solutions for different cases.

ACKNOWLEDGEMENTS

The authors would like to thank the editors and the reviewers for their constructive comments and suggestions. This work is supported by Innovation Team Program of Chongqing Education Committee (CXTDX201601019), Chongqing University Innovation Team under Grant (KJTD201312) and the National Natural Science Foundation of China (Nos. 51207064 and 61463014).

REFERENCES

[1] G. Chen, Z. Lu, Z. Zhang, and Z. Sun, "Research on hybrid modified cuckoo search algorithm for optimal reactive power dispatch problem," *IAENG International Journal of Computer Science,* vol. 45, no. 2, pp. 328-339, 2018.

[2] G. Chen, X. Yi, Z. Zhang, et al. "Applications of multi-objective dimension-based firefly algorithm to optimize the power losses, emission, and cost in power systems," *Appl Soft Comput.* vol.68, pp.322-342, 2018.

[3] S. Pisei, J. Choi, W. Lee, and D. Won, "Optimal power scheduling in multi-microgrid system using particle swarm optimization," *J. Electr. Eng. Technol.,* vol. 12, no. 4, pp. 1329-1339, 2017.

[4] G. Chen, L. Liu, P. Song, and Y. Du, "Chaotic improved PSO-based multi-objective optimization for minimization of power losses and L index in power systems," *Energ. Convers. Manage.,* vol. 86, pp. 548-560, 2014.

[5] G. Chen, X. Yi, Z. Zhang, et al. "Solving the Multi-Objective Optimal Power Flow Problem Using the Multi-Objective Firefly Algorithm with a Constraints-Prior Pareto-Domination Approach," *Energies*. vol.11, no.12, pp.3438-3456, 2018.

[6] Q. Pan, P. N. Suganthan, L. Wang, L. Gao, and R. Mallipeddi, "A differential evolution algorithm with self-adapting strategy and control parameters," *Comput. Oper. Res.,* vol. 38, no. 1, pp. 394-408, 2011.

[7] G. Nápoles, R. Falcon, Z. Dikopoulou, E. Papageorgiou, R. Bello, and K. Vanhoof, "Weighted aggregation of partial rankings using Ant Colony Optimization," *Neurocomputing,* vol. 250, pp. 109-120, 2017.

[8] C. Wang, W. Song and L. Liu, "An adaptive bat algorithm with memory for global optimization," *IAENG International Journal of Computer Science,* vol. 45, no. 2, pp. 320-327, 2018.

[9] A. Ramesh Kumar and L. Premalatha, "Optimal power flow for a deregulated power system using adaptive real coded biogeography-based optimization," *International Journal of Electrical Power & Energy Systems,* vol. 73, pp. 393-399, 2015.

[10] L. M. Abualigah, A. T. Khader, E. S. Hanandeh, and A. H. Gandomi, "A novel hybridization strategy for krill herd algorithm applied to clustering techniques," *Appl. Soft Comput.,* vol. 60, pp. 423-435, 2017.

[11] A. Banharnsakun, T. Achalakul and B. Sirinaovakul, "The best-so-far selection in Artificial Bee Colony algorithm," *Appl. Soft Comput.,* vol. 11, no. 2, pp. 2888-2901, 2011.

[12] X. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. 2009 World Congress on Nature & Biologically Inspired Computing*, 2009, pp. 210-214.

[13] M. K. Naik and R. Panda, "A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition," *Appl. Soft Comput.,* vol. 38, pp. 661-675, 2016.

[14] J. Huang, L. Gao and X. Li, "An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes," *Appl. Soft Comput.,* vol. 131, no. 15, pp. 349-356, 2015.

[15] T. T. Nguyen, A. V. Truong and T. A. Phung, "A novel method based on adaptive cuckoo search for optimal network reconfiguration and distributed generation allocation in distribution network," *International Journal of Electrical Power & Energy Systems,* vol. 78, pp. 801-815, 2016.

[16] T. T. Nguyen and D. N. Vo, "Modified cuckoo search algorithm for multiobjective short-term hydrothermal scheduling," *Swarm and Evolutionary Computation,* vol. 37, pp. 73-89, 2017.

[17] X. Li and M. Yin, "Modified cuckoo search algorithm with self adaptive parameter method," *Inform. Sciences,* vol. 298, pp. 80-97, 2015.

[18] H. Rakhshani and A. Rahati, "Snap-drift cuckoo search: A novel cuckoo search optimization algorithm," *Appl. Soft Comput.*, vol. 52, pp. 771-794, 2017.
[19] B. Zeng, J. Zhang, Y. Zhang, X. Yang, J. Dong, and W. Liu, "Active distribution system planning for low-carbon objective using cuckoo search algorithm," *J. Electr. Eng. Technol.*, vol. 9, no. 2, pp. 433-440, 2014.
[20] X. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1616-1624, 2013.
[21] T. Bäck, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. New York Oxford: Oxford University Press, 1996.
[22] N. Padhye, K. Deb and P. Mittal, "Boundary Handling Approaches in Particle Swarm Optimization," *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, vol. 201, pp. 287-298, 2013.
[23] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, "Global and local real-coded genetic algorithms based on parent-centric crossover operators," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1088-1113, 2008.
[24] Z. Cai, W. Gong, C. X. Ling, and H. Zhang, "A clustering-based differential evolution for global optimization," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 1363-1379, 2011.
[25] X. Li and M. Yin, "Multi-operator based biogeography based optimization with mutation for global numerical optimization," *Comput. Math. Appl.*, vol. 64, no. 9, pp. 2833-2844, 2012.
[26] Q. Feng, S. Liu, Q. Wu, G. Tang, H. Zhang, and H. Chen, "Modified Biogeography-Based Optimization with Local Search Mechanism," *Journal of Applied Mathematics*, vol. 2013, pp. 1-24, 2013.
[27] X. Li, J. Wang and M. Yin, "Enhancing the performance of cuckoo search algorithm using orthogonal learning method," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1233-1247, 2014.
[28] Y. Cai and J. Wang, "Differential evolution with hybrid linkage crossover," *Inform. Sciences*, vol. 320, pp. 244-287, 2015.
[29] G. Xiong, D. Shi and X. Duan, "Enhancing the performance of biogeography-based optimization using polyphyletic migration operator and orthogonal learning," *Comput. Oper. Res.*, vol. 41, pp. 125-139, 2014.
[30] A. A. Mohamed, Y. S. Mohamed, A. A. M. El-Gaafary, and A. M. Hemeida, "Optimal power flow using moth swarm algorithm," *Electr. Pow. Syst. Res.*, vol. 142, pp. 190-206, 2017.
[31] X. X. Ma and J. S. Wang, "An improved flower pollination algorithm to solve function optimization problem," *IAENG International Journal of Computer Science*, vol. 45, no. 3, pp. 364-370, 2018.
[32] S. Duman, U. Güvenç, Y. Sönmez, and N. Yörükeren, "Optimal power flow using gravitational search algorithm," *Energ. Convers. Manage.*, vol. 59, pp. 86-95, 2012.
[33] K. Vaisakh, L. R. Srinivas and K. Meah, "Genetic evolving ant direction particle swarm optimization algorithm for optimal power flow with non-smooth cost functions and statistical analysis," *Appl. Soft Comput.*, vol. 13, no. 12, pp. 4579-4593, 2013.