An Improved Differential Evolutionary Algorithm Based on Simulated Annealing and Levy Flights Mechanism

Ziqiao Yu, Xiaoxia Zhang, Xin Shen

Abstract—**The Differential Evolutionary algorithm (DE) is a well-known and effective metaheuristic method for continuous optimization problems. However, due to contraction stagnation and early convergence problems, the performance of the DE algorithm can be influenced at the convergence speed and optimization accuracy. Therefore, this paper proposes an improved DE algorithm with the Levy flights mechanism and the Simulated Annealing algorithm (SA), namely DESA-LF algorithm, which effectively uses the advantages of them with generating the better candidate solutions to reduce the convergence speed for making the optimal solution towards the global optimal direction. Finally, we use six typical functions with different complexities and make the comparisons with the DE algorithm, the DE-LF algorithm and other meta-heuristic algorithms to verify the efficient of DESA-LF. From the results and convergence curves, the DESA-LF has the advantages of extending the convergence speed and enhancing the optimization accuracy. Therefore, it demonstrates that convergence speed and optimization accuracy of DESA-LF are highly efficient and superior.**

Index Terms—**Differential Evolutionary algorithm, Simulated Annealing algorithm, Levy Flights mechanism, function optimization problem**

I. Introduction

OPTIMIZATION represents a behavior that is finding the maximum and minimum solution among the feasible solutions in a specific problem. With the fast development of studying algorithms, more and more meta-heuristic algorithms have been applied in different functions. They include CS [1], FFA [2], ABC[3], FA [4], FFO [5], DE [6] and so on. For the advantages of efficiency and simple, the DE shows better performance than other heuristic and meta-heuristic algorithms. However, it has two aspects of shortcomings in the DE algorithm. One aspect is the stagnation of contraction problem. If the optimal solution cannot be found, the algorithm will stop moving towards the global direction and get into the local optimum. The other is the early convergence problem. When the population loses diversity, the algorithm will stop searching rapidly and get into the local optimal state. Therefore, the aim of improving the algorithm is finding a useful way to get over two problems in the DE algorithm. There are many ways of improving the DE algorithm including setting the initial parameters, selecting the efficiency mutation strategy, improve the population structure and hybridize with other algorithms with the DE algorithm. The methods of hybridizing with other algorithms and changing the population structure are widely adopted to increase DE algorithm capability among four improved methods in the literature.

A good algorithm needs to make a balance between exploitation and exploration. Therefore, we propose an improved DE hybridized with SA algorithm and Levy flights mechanism, namely the DESA-LF algorithm, to help DE algorithm get over the stagnation of contraction and the early convergence problems. The Levy flights mechanism is a method of generating new solutions in larger random scope. Therefore, this method can enhance the diversity of the population to solve the early convergence problem caused by the premature population. Moreover, the SA algorithm can provide an alternative between the individuals and new solutions. It can improve searching capacity of the DE to escape from local optimal solution towards global optimal direction for solving stagnation of contraction problem. Therefore, this paper is organized as follows. In Section 2, the principle of DE is described. In Section 3, DESA-LF is shown in detail. The simulation experiments of six optimization functions are introduced in Section 4. At last, the conclusion is illustrated in the last part.

II. Basic Differential Evolutionary Algorithm

A. Principle and population initialization

DE algorithm was put forward by Stron and Price in 1997. Through the constant development, the DE has been widely used in multiple problems especially continuous optimization problems. There are three main evolution processes: mutation, crossover and selection in the DE. In the processes of mutation and crossover, Storn and Price proposed many variables and expressed them in the general formula ***DE/x/y/z***. In this definition, $\boldsymbol{x}$ represents the generating methods of based individual, $\boldsymbol{y}$ is the amount of the disturbance individual and $\boldsymbol{z}$ shows the method of the crossover scheme. The ***DE/rand/1/bin*** as the common differential strategy shows that an individual is randomly choosed ($\boldsymbol{x}$ = rand) from population as the base vectors and one pair of vectors ($\boldsymbol{y}$ = 1) as the disturbance vectors

Manuscript received September 19, 2020, revised March 7, 2021. This work was supported by Project of Liaoning Xincheng Co., Ltd (Grant No. L20170989).

Z. Q. Yu is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan, 114051, China (e_mail: yzq13624128823@163.com).

X. X. Zhang is a Professor of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan,110451,China (corresponding author, phone:86-0412-5929812; e_mail: aszhangxx@163.com).

X. Shen is a Master Student of School of Computer Science and Software Engineering, University of Science and Technology LiaoNing, Anshan, 114051, China (e_mail: shenxin_4395@163.com).

together generates the mutation individuals in the mutation process. The specific definitions of variables are described as follows.

One population is made up of *NP* individuals X_i^{G}. One individual includes *D* variables in the range of searching space [$x_{\min}, x_{\max}$]. The initial individual X_i^{G} is randomly produced within range of searching scope. And the mutation and crossover processes can evolute into the new individuals V_i^{G} and U_i^{G}. V_i^{G} represents the mutation individual after the mutation process. U_i^{G} represents the trial individual after the crossover process. In the last evolution process, the individual with the best fitness value is selected between X_i^{G} and U_i^{G} to consist of new population in next generation.

B. Principle and population initialization

For the strategy of ***DE/rand/1/bin***, it randomly chooses three individuals $X_{r1}^{\mathrm{G}}, X_{r2}^{G}$ and X_{r3}^{G} from current population and generates the mutation individual V_i^{g} according to Eq (1).

$$V_i^{\mathrm{G}} = X_{r1}^{G} + F\left(X_{r2}^{G} - X_{r3}^{G}\right) \qquad r1 \neq r2 \neq r3 \tag{1}$$

Where ***F*** is the mutation parameter or scale factor which is between 0 and 2. The shrinkage level of the disturbance individual $(X_{r1}^{\mathrm{G}} - X_{r2}^{G})$ depends on *F* and influences the searching direction of global optimal solution in DE algorithm.

C. Crossover Process

When the mutation process finished, the crossover process will start. In the crossover process, the operating objects include the initial and mutation individuals of $V_i^G = [v_{i,1}^G, v_{i.2}^G, \ldots, v_{i,D}^G]$ and $X_i^G = [x_{i,1}^G, x_{i.2}^G, \ldots, x_{i,D}^G]$. And the binomial crossover scheme will be adopted to generate the trail individual in Eq (2).

$$u_{i,j}^{G} = \begin{cases} v_{i,j}^{G} & \text{if } \; rand(0,1) \leq CR \cup j = j_{rand} \\ x_{i,j}^{G} & \text{otherwise} \end{cases} \tag{2}$$

Where *CR* is crossover factor which is between 0 and 1. j_{rand} is an integer which randomly generates between 1 and *D*. $j = j_{rand}$ can guarantee that at least one trial individual variable after crossover operation is from the mutation individual variable. It can avoid search stagnation due to the population homogenization. Therefore, if the mutation process is characterized by the generation of mutation individuals forming the next iteration population, then the crossover process is characterized by the selection of individual variables with the certain probability between the variation and the initial individual variables to form the trial individual.

D. Selection Process

DE algorithm uses one to one competition between parents and offspring to greedily select. This strategy is different from other evolutionary algorithms such as tournament selection, sort selection and so on. This greedy selection can improve the quality of population by selecting the best fitness individual. Therefore, Eq (3) describes the selection strategy which chooses the individuals of the population to compose the next generation population from the initial individuals and corresponding trail individuals in the current generation.

$$X_i(g) = \begin{cases} U_i(g) & \text{if } f(U_i(g)) < f(X_i(g)) \\ X_i(g) & \text{otherwise} \end{cases} \tag{3}$$

f () is fitness function to evaluate U_i^{G} and X_i^{G} in the current generation. $X_i^{\mathrm{G+1}}$ shows that DE algorithm chooses the individual with better fitness value between U_i^{G} and X_i^{G} as the new individual combining the next population. The searching will stop when the iterations *G* exceeds the maximum iterations. Otherwise, DE algorithm will continue searching for optimizing the population until meeting with all conditions. The specific algorithm is shown in Fig 1.

```
DE Algorithm
Initialize parameters Population NP; Dimension D;
and Generation G
Let f() be the given Objective function
G←1
for i = 1 to NP do
  for j = 1 to D do
    x^G_{i,j} = x_{min,j} + rand(0,1)*(x_{max,j} - x_{min,j})
  end
end
while stopping iteration criterion is not satisfied do
  for i = 1 to NP do
      ▸ (Mutation and Crossover)
      for j = 1 to D do
          v^G_{i,j} = Mutation ( x^G_{i,j})
          u^G_{i,j} = Mutation ( x^G_{i,j} , v^G_{i,j})
      end
      ▸ (Selection)
      if f(U^G_i) < f(f(X^G_i)) then
          X^{G+1}_i ← U^G_i
      else
          X^{G+1}_i ← X^G_i
      generate X^{G+1}_i within the allowed bounds
      end if
  end
  choose best individual as X_gbest
    G = G + 1
end while
```

Fig. 1 DE algorithm Pseudo code

At the beginning of the DE, it initializes the parameters which are the population size *NP*, the dimension of each individual *D* and the maximum generation *G* (step 1). Let the given objective function be fitness value function and calculate fitness values of each individual from population (step 2). The method of generating the initial population adopts random generation in the certain searching range (step 3). From step 9 to 15, the certain evolutionary strategy of the mutation and the crossover process can update the population individuals and the variables of individuals to generate the mutation individuals and trail individuals. The next generation individuals which choose the best fitness value can be generated in the selection process (step 16-20) and save the best individual X_{gbest}. Therefore, the whole algorithm can find the optimal individual and update the high-quality population by the evolutionary strategy.

III. DESA-LF ALGORITHM

A. Simulated Annealing Algorithm

Simulated Annealing algorithm (SA) is a nature-inspired meta-heuristic algorithm to utilize whatever in continuous optimization problem or combination optimization problem. Especially the Metropolis principle in SA algorithm is the efficient method of updating the population [7]. The specific SA algorithm is shown in Fig.2 as follows:

```
Simulated Annealing Algorithm
1: Initialize X, T0, Tend, q and L
2: While T0 > Tend :
3:    fitness_current = f(X)
4:    Xnew = Neighbor(S)
5:    fitness_new = f(Snew)
6:    dc = fitness_new - fitness_current
7:    if (dc < 0) then
8:       S = Snew and fitness_current = fitness_new
9:    else
10:      if (rand(0,1)) < e ^ (dc / T) then
11:         S = Snew and fitness_current = fitness_new
12:      end if
13:   end if
14:   After every L iterations Set T = q*T
15: end while
```

Fig. 2. SA algorithm pseudo code.

It begins from an initial random population with *NP* solutions *X* and sets parameters T_0 (initial temperature), *T* (termination temperature), *q* (annealing rate) and *L* (annealing iteration times). Calculate fitness values of solutions in current population (step 3). Generate (step 4) and calculate the new neighborhood (step 5). A better is always accepted because of its best fitness value. Meanwhile, an inferior can be accepted with the certain probability. And the probability can be decided by the fitness values and current temperature *T* (step 10). The temperature firstly maintains at a higher level so that the inferior solution shifts towards the global direction and is dropped according to a factor *q* (step 14) by degrees to reduce the possibility of receiving the inferior solutions. SA algorithm can achieve the goal of moving from local optimal direction to global optimal direction.

B. Levy flights mechanism

Levy flights mechanism belongs to one type of random walks. There are two parts of Levy flights including the selection of random orientation and the yielding steps which follow the choice Levy distribution. Lots of methods can realize it, but one of the most resultful ways is Mantegna algorithm for a symmetric Levy stable distribution. Fig. 3 shows trace of Levy flights of 1000

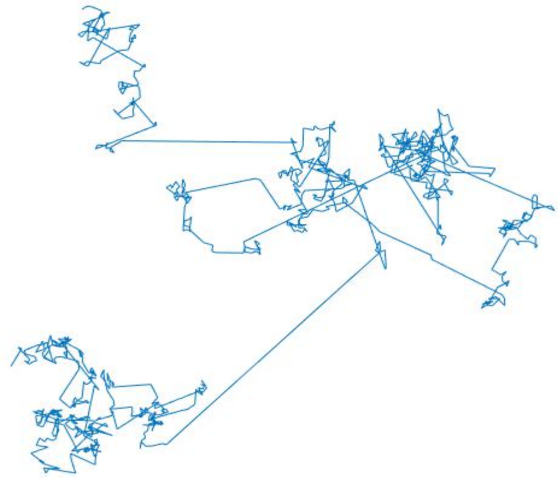


Fig.3. Levy flights trace.

steps beginning from (0,0). From the figure, we can obviously see the individuals of population can be generated in a larger range with the certain step. Therefore, individuals can be obtained more diverse than the randomly generating method of the traditional DE algorithm [8].

In Mantegna's algorithm, step length *s* can be defined by

$$s = \frac{\mu}{|v|^{1/\beta}} \tag{4}$$

Where μ and v come from normal distributions.

That is

$$\mu \sim N\left(0, \sigma_\mu^2\right), \; v \sim N(0.\sigma_v^2) \tag{5}$$

Where

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)]\beta\, 2^{(\beta-1)/2}} \right\}^{1/\beta} \quad \sigma_v = 1 \tag{6}$$

Therefore, next population is updated by Eq (8) as follow:

$$X_{t+1} = X_t + \alpha S \tag{7}$$

Where *S* represents the updating step of population individuals. By adding the Levy flights mechanism in DE algorithm, it can search in the space far from the former optimal solution, which ensures the system not to get into the local optimum and continue finding optimal value towards the global direction to help the algorithm escape from the local optimum. Meanwhile, the search efficiency is maximized in the uncertain environment to increase the probability of searching the better optimum solution. Levy Flights pseudo is shown in Fig.4.

```
Levy Flights
Initialize α, β ,μ, ν, s
according to Eq(7) to update the μ
according to Eq(8) to update the step s
for j = 1 : D do
    Xnew(j) = X(j) + α*S
enter into boundary adjustment
end
```

Fig. 4. Levy Flights mechanism pseudo code.

C. DESA-LF Algorithm

A good algorithm needs getting a balance between exploitation and exploration. When exploration and exploitation are unbalanced, the DE algorithm will lead to the stagnation of contraction problem and early convergence problem. For solving this unbalance, the meta-heuristic algorithm must show better exploration abilities at the earlier phases and improve exploitation abilities to later phases. Therefore, we propose a new DE algorithm called DESA-LF by adding the SA algorithm and Levy flights mechanism to get over the problems. In DESA-LF algorithm, Levy flights mechanism as a method of generating new solutions can enrich the diversity of the population for solving population premature problem. Meanwhile, the SA algorithm will adjust new solution and update the population of the DE algorithm with the Metropolis principle. It plays a role in finding the global optimal direction to escape from local optimal individual for stagnation of contraction problem in DE algorithm.

In Fig 5, the new parameters in SA are led into DE algorithm (step 3). According to the Metropolis principle, When the temperature of annealing T_0 is high, the whole algorithm can accept the inferior solution according to one possibility to help algorithm jump out of local optimum. When temperature is gradually down, the probability of accepting the inferior solution is also getting small and until the temperature of annealing T_0 is lower than the terminal temperature T, the annealing process is over. Meanwhile, the Levy flights mechanism is different from the traditional random method of generating new solutions. It can generate the solutions which are far away from the current optimal individual in the range of searching space to form a new solution X_{new} (step 26). And the new solution X_{new} will join in the next evolutionary population based on the Metropolis principle. Moreover, there are two iterative cycles in the DESA-LF algorithm. L and G represent the inner annealing and outer iterative cycle. And the annealing operation will repeat L times searching at each annealing temperature in the annealing process (step 27 to 33). When the iteration time G reaches the maximum iteration time G_m, the whole algorithm will be ended.

IV. EXPERIMENTS AND RESULTS ANALYSIS

Six benchmark functions are applied for examining performance of DESA-LF. These functions from Table I are divided into two types according to modality. In the first type, f_4 and f_5 are unimodal functions which can prove the convergence speed of one algorithm. In the second type, f_1 - f_3 and f_6 are multimodal functions which can test the capacities of global search. Among them, f_1 - f_3 and f_6 have many local minimums. Function f_4 has d local minimums except for the global one. According to the minimum, these functions can also be divided into two types. In the first type, the minimum values of $f_1 - f_5$ are all 0. The second type is f_6 and the minimum is -418.982. In Table II, Best and Worst represent best and worst values in the DE, the DE-LF and the DESA-LF algorithms among ten runs of the six functions. The *Average* and *Std* represent that the average results and standard deviations of these three algorithms. It can be learned that the accuracy and stability among three algorithms. Fig 6 shows convergence curves of six test functions. There are three convergence curves in each function, which represent three different algorithms.

```
DESA-LF Algorithm
1:Initialize NP, D, G, α , T0 and L
2: Let f () be the given Objective function
3:Initialize  α ,T0 and L
4: T ← T0
5:G ← 1
6: for i = 1 to NP do
7:     for j = 1 to D do
8:         x^G_{i,j} = x_{min,j} + rand(0,1)*(x_{max,j} − x_{min,j})
9:     end
10: end
11:while stopping iteration criterion is not satisfied do
12:     for i = 1 to NP do
13:         ► (Mutation and Crossover)
14:         for j = 1 to D do
15:             v^G_{i,j} = Mutation (x^G_{i,j})
16:             u^G_{i,j} = Crossover(x^G_{i,j}, v^G_{i,j})
17:         end
18:         ► (Selection)
19:         if f (U^G_i) < f (X^G_i) then
20:             X^{G+1}_i ← U^G_i
21:         else
22:             X^{G+1}_i ← X^G_i
23:         generate X^G_i within the allowed bounds
24:         end if
25:   choose best inidividual as X_gbest
26:   generate the  new individual X_new by Levy Flights
27:    dc = fitness(X_new) - fitness(X_gbest)
28:    if (dc < 0  OR  rand(0,1)) < e ^ (dc / T)) then
29:   Update new  population by replacing X_gbest by  X_new
30:   end if
31:  end for
31:    After every L iterations Set T =  α T
32:    G = G + 1
33:end while
```

Fig. 5. DESA-LF pseudo code.

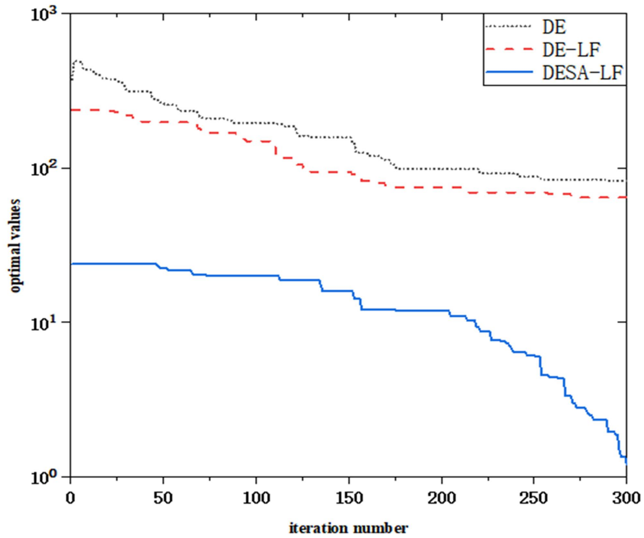
From the results, the DESA-LF is better than the DE and DE-LF. The results of three algorithms are in 30 dimensions. It can be learned from ten results of DE-LF that Levy flights mechanism has shown itself advantage of enlarging the searching scope and enrich the diversity of population to increase the probability of obtaining a better function value. But the Levy flights mechanism is unstable

TABLE I
SIMULATION TESTING FUNCTIONS

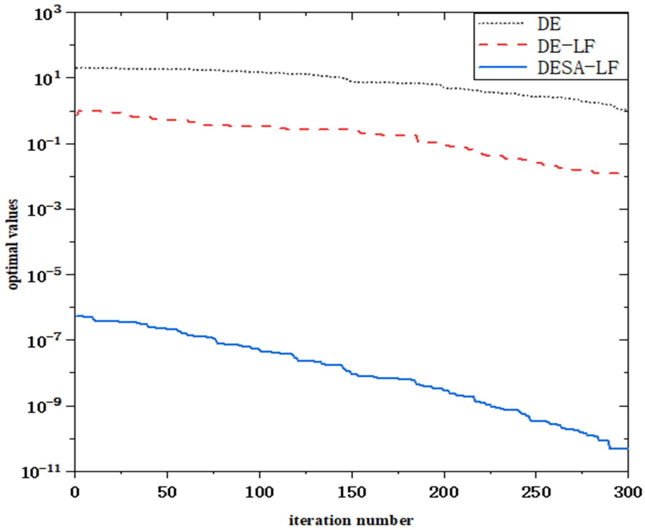
Function name	Expression	Range
Rastrigin	$f_1(x) = n*10 + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right)$	[-5.12,5.12]
Ackley	$f_2(x) = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2p\,x_i))$	[-32,32]
Girewank	$f_3(x) = \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]
Sphere	$f_4(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]
Step	$f_5(x) = \sum_{i=1}^{n}([x_i+0.5])^2$	[-100,100]
Schwefel Problem	$f_6(x) = \sum_{i=1}^{n} - x_i\sin(\sqrt{\lvert x_i \rvert})$	[-500,500]

TABLE II
THE SIMULATION RESULTS OF THREE ALGORITHMS

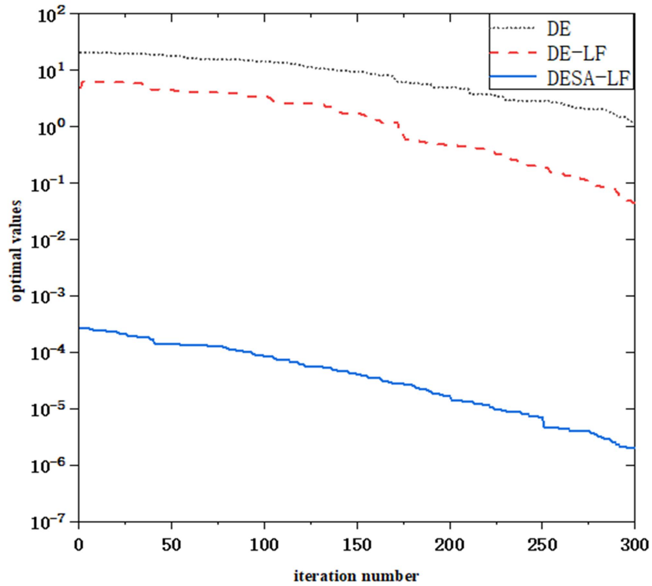
Function name	algorithm	*Best*	*Worst*	*Average*	*Std*	*f*min
f_1	*DE*	8.25E+01	1.31E+02	8.88E+01	1.68E+01	0
	DE-LF	7.05E+01	1.22E+01	8.78E+01	1.66E+01	0
	DESA-LF	9.12E-01	8.66E+00	6.63E+00	2.93E+00	0
f_2	*DE*	8.60E-01	1.62E+00	1.39E+00	2.77E-01	0
	DE-LF	2.44E-02	5.30E-02	3.94E-02	9.99E-03	0
	DESA-LF	1.49E-06	2.92E-06	1.78E-06	2.41E-07	0
f_3	*DE*	9.53E-01	9.98E-01	9.82E-01	1.36E-02	0
	DE-LF	6.16E-04	1.23E-02	1.39E-02	3.53E-03	0
	DESA-LF	1.89E-12	1.87E-11	6.69E-12	6.90E-12	0
f_4	*DE*	3.04E+00	1.15E+01	6.15E+00	2.70E+00	0
	DE-LF	2.50E-03	1.43E-03	3.20E-03	1.00E-03	0
	DESA-LF	1.39E-12	8.92E-12	5.44E-12	2.60E-12	0
f_5	*DE*	2.14E+00	1.06E+01	6.80E+00	2.64E+00	0
	DE-LF	1.70E-03	5.57E-03	3.32E-03	1.33E-03	0
	DESA-LF	3.49E-12	1.84E-11	9.83E-12	4.17E-12	0
f_6	*DE*	-0.96 E+04	-1.02E+04	-0.98E+04	3.51E+02	-418.982
	DE-LF	-3.23E+03	-3.93E+03	-3.47E+03	2.83E+02	-418.982
	DESA-LF	-4.73E+02	-2.56E+02	-4.51E+02	8.18E+01	-418.982



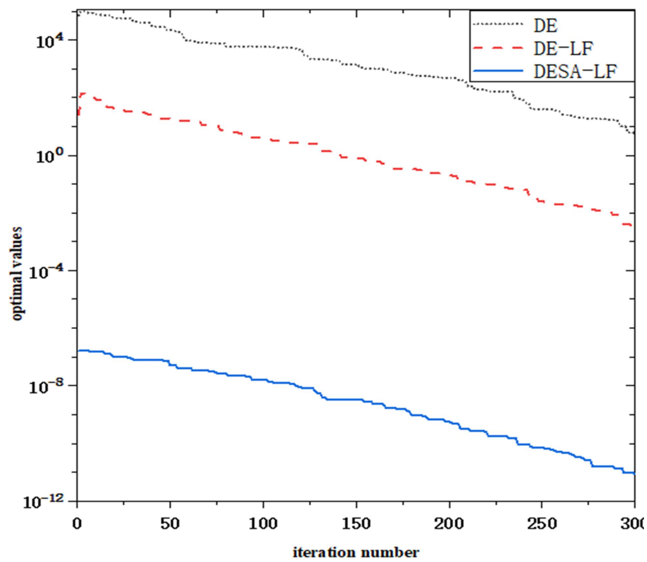
f_1



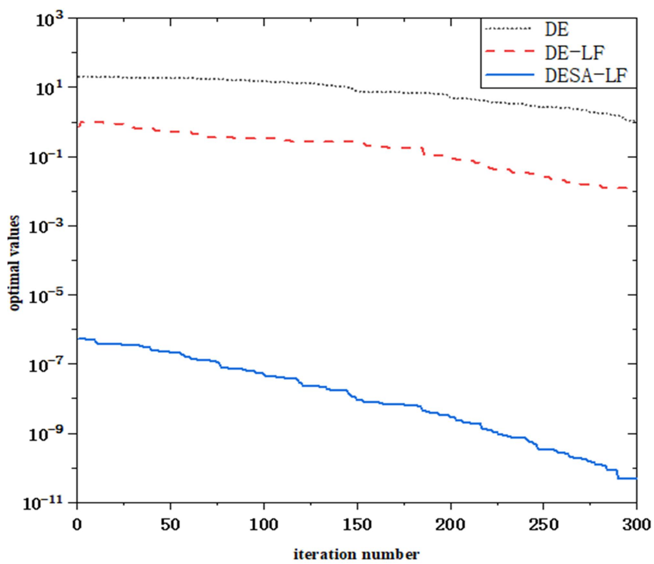
f_4



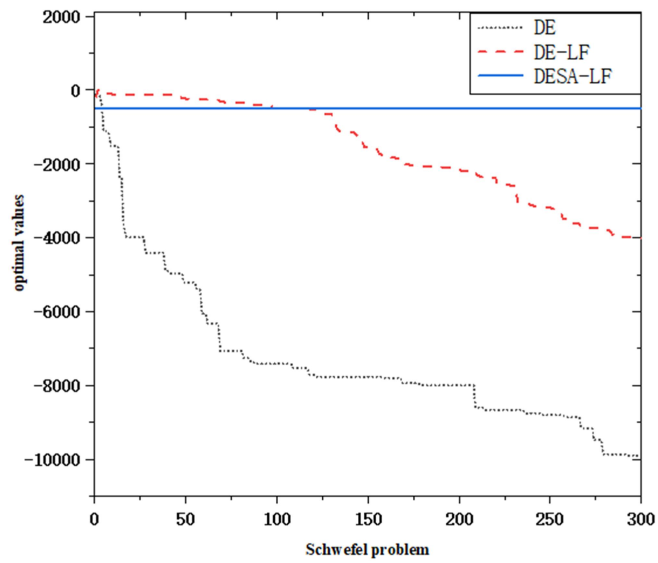
f_2



f_5



f_3



f_6

TABLE III
MANY RESULTS OF ALGORITHMS COMPARISON

Function	*f*min	DESA-LF	BA	BBO	CSA	PSO
Rastrigin	0	9.12E-01	6.63E+01	7.58E+00	4.35E+01	5.22E+01
Ackley	0	1.49E-06	1.45E+01	3.31E-01	4.43E+00	1.86E+00
Girewank	0	1.89E-12	1.13E+02	6.14E+01	1.57E+01	1.50E-02
Sphere	0	1.39E-12	1.07E+04	2.28E+01	7.25E+01	1.67E-03
Step	0	3.49E-12	1.16E+04	1.29E+00	6.58E+01	1.20E-03
Schwefel Problem	-418.982	-4.73E+02	-2.84E-71	-1.02E+04	-6.26E+03	-6.84E-03

Fig. 6. Simulated results of three algorithms.

in obtaining the best and worst solutions, which can influence the optimal accuracy. Therefore, the SA algorithm can achieve the adjustment to accept the inferior solution which Levy flights mechanism generates on a certain probability to stable the optimal accuracy in the DESA-LF. From the convergence curves of three algorithms, the convergence speed of the DESA-LF has been slower than DE and DE-LF in Fig 6. Therefore, the DESA-LF has been proved the efficiency in convergence speed and accuracy. Then in Table III, the DESA-LF is compared with the four nature-inspired meta-heuristic algorithms so as to make full algorithm analysis. The four meta-heuristic algorithms are BA [9], CSA [10], BBO [11], and PSO [12]. The experiment environment of the DESA-LF algorithm is the same with four nature-inspired meta-heuristic algorithms. The results of the DESA-LF algorithm are obviously superior to other algorithms.

V. CONCLUSION

An improved algorithm called DESA-LF algorithm is proposed for solving the stagnation of contraction and early convergence problems in DE algorithm. In the DESA-LF algorithm, the SA algorithm uses the annealing process to help the DE algorithm escape from the local optimal towards global direction. Especially the Metropolis principle can determine whether to accept the new individual, which avoids the stagnation of contraction problem. Meanwhile, Levy flights mechanism can enlarge the range of searching space and increase the probability of obtaining the optimal value to avoid the early convergence problem. The experiment shows that the improved DESA-LF algorithm has good performance in convergence speed and accuracy.

REFERENCES

[1] G. G. Chen, S. Y. Qiu, Z. Z. Zhang and Z. Sun, “Quasi-oppositional Cuckoo Search Algorithm for Multi-objective Optimal Power Flow,” *IAENG International Journal of Computer Science*, vol. 45, no. 2, pp. 255-266, 2018.

[2] J. Wu, Y. G. Wang, K. Burrage, Y. C. Tian, B. Lawson and Z. Ding, “An improved firefly algorithm for global continuous optimization problems,” *Expert System with Applications*, vol. 149, pp. 113340-113351, July 2020.

[3] W. Gao, S. Liu S and L. Huang, “A global best Artificial Bee Colony algorithm for global optimization,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp.2741-2753, 2012.

[4] Q. Qu, M. L. Qi, R. B. Gong and X.B. Chen, “An Improved Enhanced Fireworks Algorithm Based on Adaptive Explosion Amplitude and Levy Flight,” *Engineering Letters*, vol. 28, no.4, pp.1348-1357,2020.

[5] L. Wang, Y. N. Xiong, S.W. Li and Y. R. Zeng, “New fruit fly optimization algorithm with joint search strategies for function optimization problems,” *Knowledge-Based Systems*, vol. 176, pp. 77-96, 2019.

[6] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal. Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[7] D.C. Tiwari, “Spider Monkey Optimization Algorithm Based on Metropolis Principle,” *Springer*, vol. 546, pp. 113, 2017.

[8] J. Ellis, N Petrovskaya, S Petrovskii., “Effect of density-dependent individual movement on emerging spatial population distribution: Brownian motion vs Levy flights.” *Journal of theoretical biology*, vol. 464, pp. 159-178, 2019.

[9] X. S. Yang, “A new metaheuristic bat-inspired algorithm,” in *Nature Inspired Cooperative Strategies for Optimization*, Springer, 2010, pp. 65-74.

[10] A. Askarzadeh, “A novel metaheuristics method for solving constrained engineering optimization problems: Crow search algorithm,” *Computers Structures,* vol. 169, pp. 1-12, 2016.

[11] D. Simon, “Biogeography-based optimization optimization,” *IEEE Transactions on Evolutionary Computation,* vol. 12, no. 6, pp.702-713, 2008.

[12] R. Eberhart and J.Kennedy, “Particle swarm optimization” in *Proc. IEEE Int. Conf. Neural Netw*., vol. 4, pp. 1942-1948, 1995.