An Efficient Improvement of CMA-ES Algorithm for the Network Security Situation Prediction

Guan-Yu Hu and Pei-Li Qiao*

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, HeiLongJiang, 150080, PR China

Abstract: An improved covariance matrix adaptation evolution strategy algorithm (CMA-ES) is proposed and it is used to train the forecasting model of the network security situation in this paper. A new recombination strategy which adds a heuristic component is developed in the improved CMA-ES algorithm, and the search speed is accelerated. The experimental results show that, compare with original algorithm and its variants, the improved CMA-ES algorithm can greatly increased the search speed in high dimensional problems. The improved CMA-ES algorithm is an efficient evolutionary algorithm which can be applied to the network security situation prediction.

Keywords: Network security situation prediction, covariance matrix adaptation evolution strategy, evolutionary algorithm.

1. INTRODUCTION

The network security situation is a series of data showing that the comprehensive information of the network security status [1], and the prediction of the network security situation is very important to active defense [2] of the network platform. In order to predict the network security situation, many methods have been proposed, such as HMM (hidden Markov model) [3-6], and the neural network [7]. All of these forecasting model need the parameter optimization in order to achieve proper effect, and its parameter optimization has the following characteristics:

1) There are too many parameters of the forecasting model. Therefore the optimization is a high dimensional problem and it is difficult to find the optimal solution.

2) The forecasting model of the network security situation is a complex and nonlinear system, so its optimization is easily to fall into the local optimal solution.

There are many optimization algorithms are used to optimize the parameters of the forecasting model, such as GA[8], PSO[9] and DE[10], but these algorithms can not solve the problems as mentioned above. For this reason, an improved CMA-ES algorithm is proposed and it is used to train the forecasting model of the network security situation in this paper.

The covariance matrix adaptation evolution strategy (CMA-ES) [11] is a high-performance evolutionary algorithm of real-parameter optimization for non-linear non-convex problems, and it is considered as a state-of-the-art algorithm in the evolutionary computation. CMA-ES performs well on a number of test functions [12,13,14] and was successfully applied to many practical problems [15].

There are three operations in the CMA-ES algorithm: the sampling of the new solutions with multivariate normal distribution, the selection and recombination, and the adapting of the covariance matrix. The covariance matrix describes the rotation and the scale of the mutation distribution, which provides more guidance to the population.

At present, more and more varieties of the CMA-ES algorithm are designed. Anne Auger and Nikolaus Hansen presented a restart CMA evolution strategy with increasing population size [16] in 2005. Christian Igel, etc. presented the (1+1)-CMA-ES [17] which only generates one candidate solution per iteration step in 2006. The candidate solution would become the new mean if it is better than the current mean. Jastrebski, G. A presented the active-CMA-ES which calculate with active covariance matrix adaptation [18] in 2006. Christian Igel, etc. extended the CMA-ES to multi-objective optimization called MO-CMA-ES [19] in 2007. Nikolaus Hansen proposed a multistart CMA-ES with equal budgets for two interlaced restart strategies [20] in 2009.

A simple but efficient improvement of the CMA-ES algorithm is proposed in this paper, which is called CMA-mES. A new recombination strategy which add a heuristic component is developed in CMA-mES.

We choose some typical benchmark functions and some variants of CMA-ES to test the performance of the new improvement. The experimental results show that CMA-mES has a great advantage for the optimization of high dimensional and nonlinear problems, which is suitable to the forecasting model of the network security situation.

The paper is organized as follows. In section 2, the principles of CMA-ES will be introduced. In section 3, CMA-mES will be described. In section 4, the performance of the proposed algorithm will be analyzed. In section 5, a case study for predicting the network security situation is presented, and the paper is concluded in section 6.

1874-4443/15 2015 Bentham Open

2. THE PRINCIPLES OF CMA-ES ALGORITHM

The basic CMA-ES has three parts: Sampling, Selection and Recombination, and Adapting the covariance matrix. Here are the principles.

2.1. Sampling

In the CMA-ES, a solution of the population is generated by a multivariate normal distribution:

$$x_k^{g+1} \sim m^g + \sigma^g \mathbf{N}\left(0, C^g\right) \quad k = 1, \cdots, \lambda \tag{1}$$

where $\mathbf{N}$ denotes normal distribution, C denotes the covariance matrix, m denotes the mean of the search distribution, g denotes the generation, k denotes the k-th offspring, σ denotes the step size.

Each element of the covariance matrix is a covariance between the dimensions. The covariance matrix generalizes the notion of variance to multiple dimensions.

2.2. Selection and Recombination

The selection operation will select μ optimal solutions from the population. The basis of the selection is the fitness value of the objective function.

The recombination operation is the update strategy of the mean. The new mean can updated by the following rule:

$$m^{g+1} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{g+1} \tag{2}$$

$$\sum_{i=1}^{\mu} w_i = 1 \tag{3}$$

where μ denotes the offspring population size, λ denotes the parent population size, w denotes the weight coefficients, $x_{i:\lambda}$ denotes the i optimal solutions in the λ solutions. Eq. 3 shows that the new mean of the population depends on the offspring population, which is the weighted average of μ individuals. A variance effective selection mass is also introduced and will be repeatedly used in the following.

$$\mu_{eff} = \left(\frac{\|w\|_1}{\|w\|_2}\right)^2 = \frac{\|w\|_1^2}{\|w\|_2^2} = \frac{1}{\|w\|_2^2} = \left(\sum_{i=1}^{\mu} w_i^2\right)^{-1} \tag{4}$$

2.3. Adapting the Covariance Matrix

The initial covariance matrix is a symmetric, and positive definite matrix:

$$C = BD^2B^T \tag{5}$$

$$B^TB = BB^T = \mathbf{I} \tag{6}$$

where B denotes an orthogonal matrix, D denotes a diagonal matrix with square roots of eigenvalues of C as diagonal elements. The new covariance matrix in next generation is:

$$C^{g+1} = \left(1 - c_1 - c_\mu\right)C^g + c_1 p_c^{g+1}\left(p_c^{g+1}\right)^T + c_\mu \sum_{i=1}^{\mu} w_i y_{i:\lambda}^{g+1}\left(y_{i:\lambda}^{g+1}\right)^T \tag{7}$$

$$c_1 \approx \frac{2}{n^2}, c_\mu \approx \min\left(\frac{\mu_{eff}}{n^2}, 1 - c_1\right) \tag{8}$$

where c_1 and c_μ denotes the learning rate for updating the covariance matrix, n denotes dimension. where y is:

$$y_{i:\lambda}^{g+1} = \frac{\left(x_{i:\lambda}^{g+1} - m^g\right)}{\sigma^g} \tag{9}$$

p_c denotes the evolution path, the initial $p_c = 0$, the new p_c is:

$$p_c^{g+1} = \left(1 - c_c\right)p_c^g + \sqrt{c_c\left(2 - c_c\right)\mu_{eff}}\,\frac{m^{g+1} - m^g}{\sigma^g} \tag{10}$$

where c_c denotes the backward time horizon of the evolution path, $c_c \leq 1$, and σ is updated by Eq. 11:

$$\sigma^{g+1} = \sigma^g \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\left\|p_\sigma^{g+1}\right\|}{\mathbf{E}\left\|\mathbf{N}(0,\mathbf{I})\right\|} - 1\right)\right) \tag{11}$$

where d_σ denotes a damping parameter. $\mathbf{E}$ denotes a expectation of the Euclidean norm:

$$\mathbf{E}\left\|\mathbf{N}(0,\mathbf{I})\right\| = \frac{\sqrt{2}\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)} \approx \sqrt{n} + O\left(\frac{1}{n}\right) \tag{12}$$

where c_σ denotes the backward time horizon in Eq. 11, p_σ denotes a conjugate evolution path, the initial $p_\sigma = 0$, the new p_σ reads:

$$p_\sigma^{g+1} = \left(1 - c_c\right)p_\sigma^g + \sqrt{c_\sigma\left(2 - c_\sigma\right)\mu_{eff}}\,C^{(g)-\frac{1}{2}}\frac{m^{g+1} - m^g}{\sigma^g} \tag{13}$$

3. THE CMA-MES ALGORITHM

The mean value of CMA-ES denotes a weighted average of offspring solutions after the selection, which leads the whole population in the solution space. The action of the mean has little heuristic and guidance which are very important to a evolutionary algorithm. The CMA-mES is proposed to solve this problem, a new recombination whose improvement is an efficient updating strategy of the mean is proposed in the CMA-mES algorithm.

3.1. The New Recombination

According to the Eq. 2, a heuristic component is added to the update strategy of the mean. The new mean reads:

$$m^{g+1} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{g+1} - \alpha x_{best}^{g+1} \tag{14}$$

where x_{best}^{g+1} denotes the best individual of the population, α denotes a reduction factor. An optimization case is shown in Fig. (**1**):

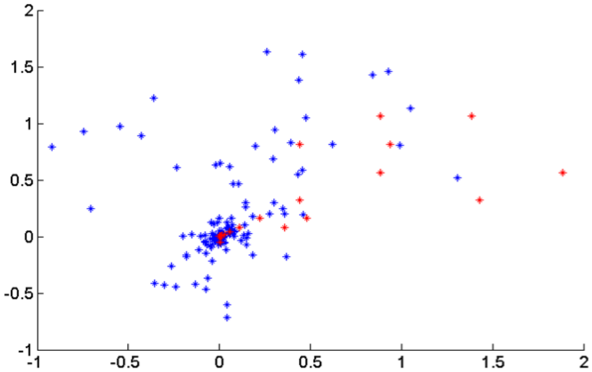

Fig. (1). An optimization case between CMA-ES and CMA-mES.

Fig. (**1**) describes the trace of the individuals. The blue trace belongs to the CMA-ES algorithm. The red trace belongs to the CMA-mES algorithm. It can be seen that the convergence speed of the improved algorithm is significantly faster than the original algorithm.

In order to determine the α, we run a series of tests to find the reasonable value. The large value of α will lead the population to miss the optimal solution and the small value of α will make the component loss effect. The test result in different value is shown as follows.

Figs. (**2**)-(**17**) describe the comparison results with a different α in dimensions 10, 20, 50, 100. The black line represents the original algorithm. The red line represents the value of α is 0.1. The green line represents the value of α is 0.3. The blue line represents the value of α is 0.5. The purple line represents the value of α is 0.7. The results show that the reasonable α is 0.3 for all the dimensions.

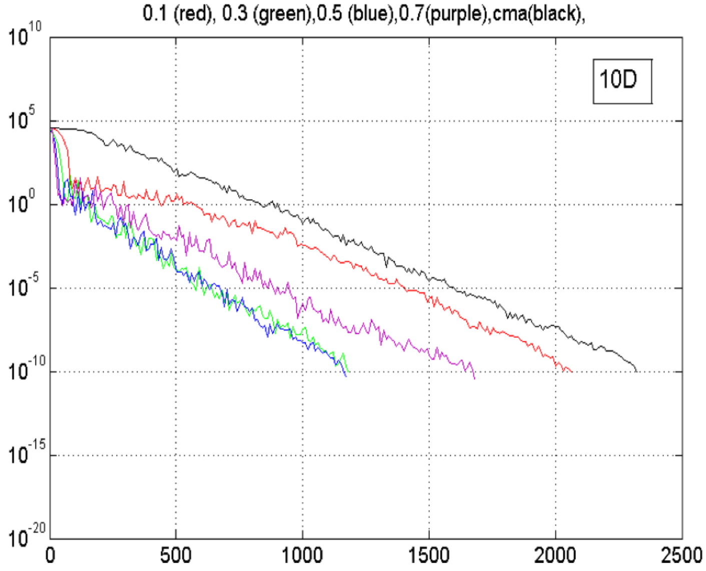

Fig. (2). Different value of α in f1 (10 D).

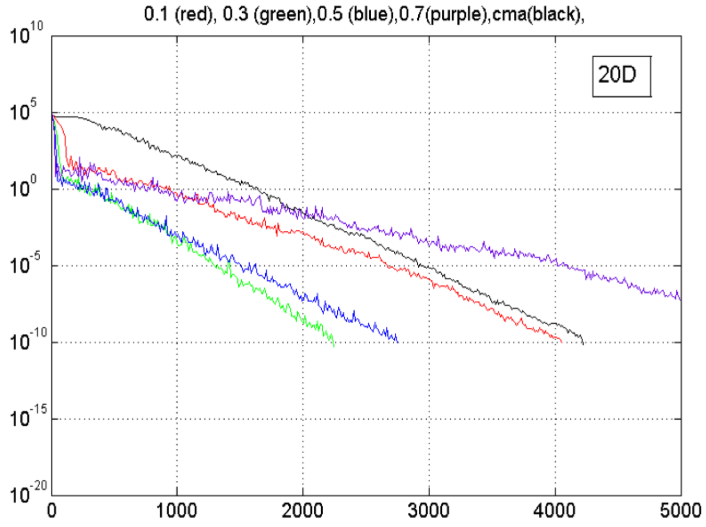

Fig. (3). Different value of α in f1 (20 D).

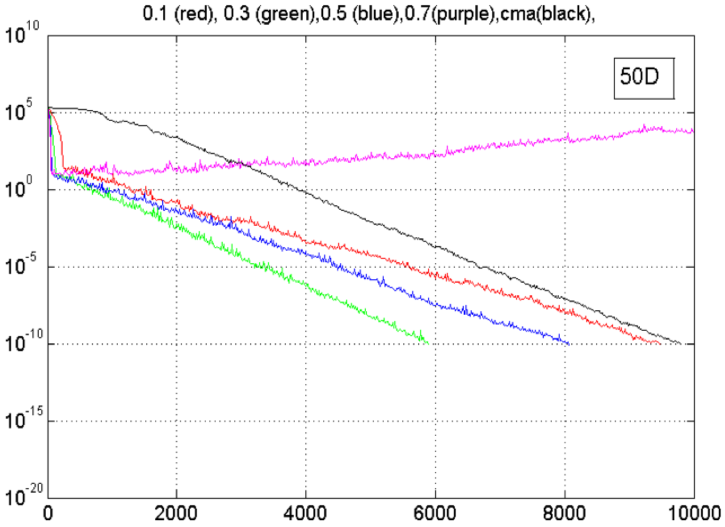

Fig. (4). Different value of α in f1 (50 D).

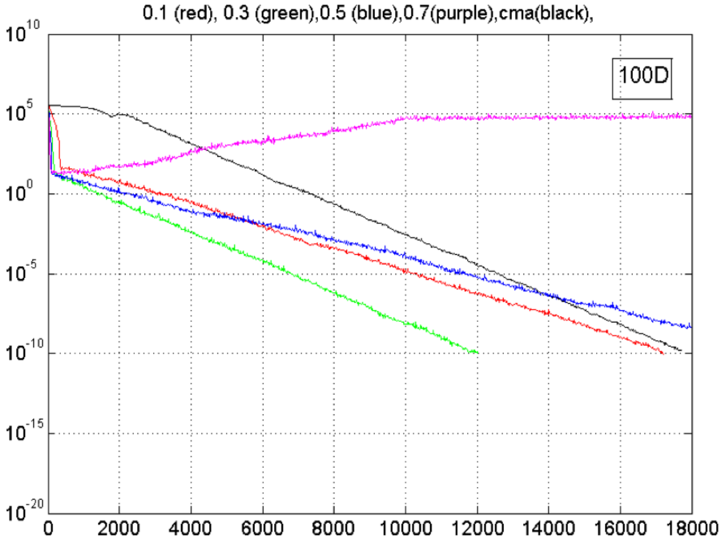

Fig. (5). Different value of α in f1 (100 D).

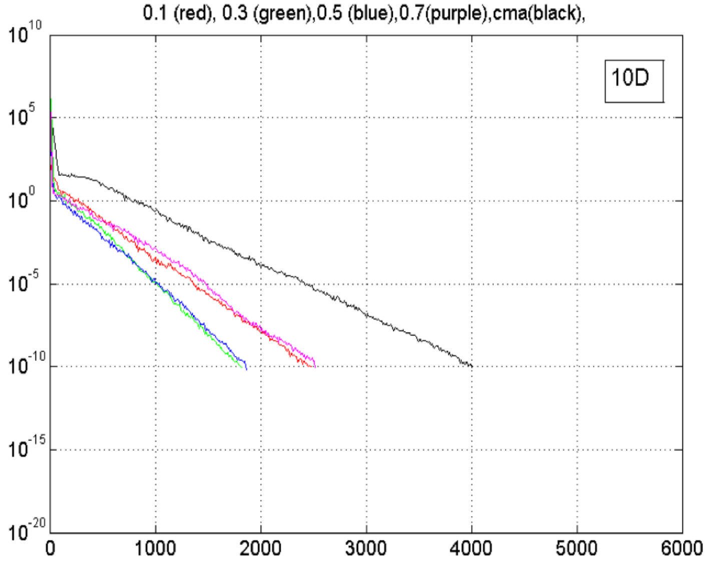

Fig. (6). Different value of α in f2 (10 D).

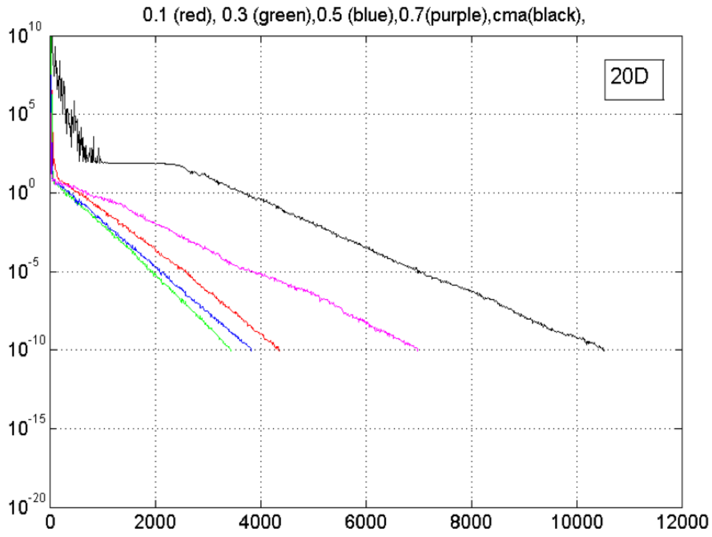

Fig. (7). Different value of α in f2 (20 D).

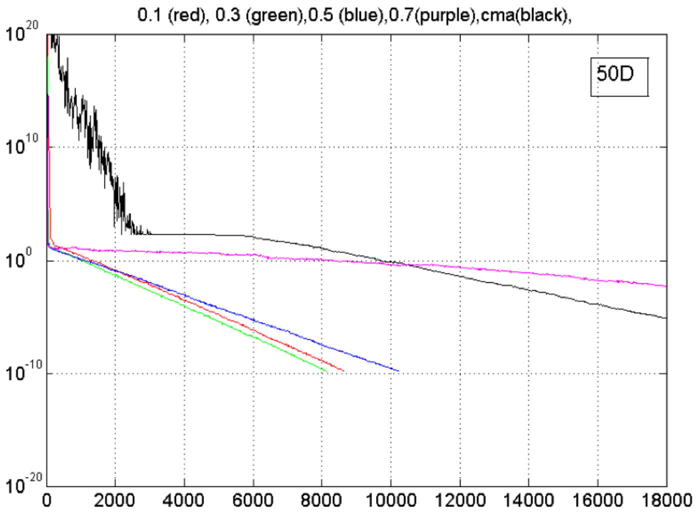

Fig. (8). Different value of α in f2 (50 D).

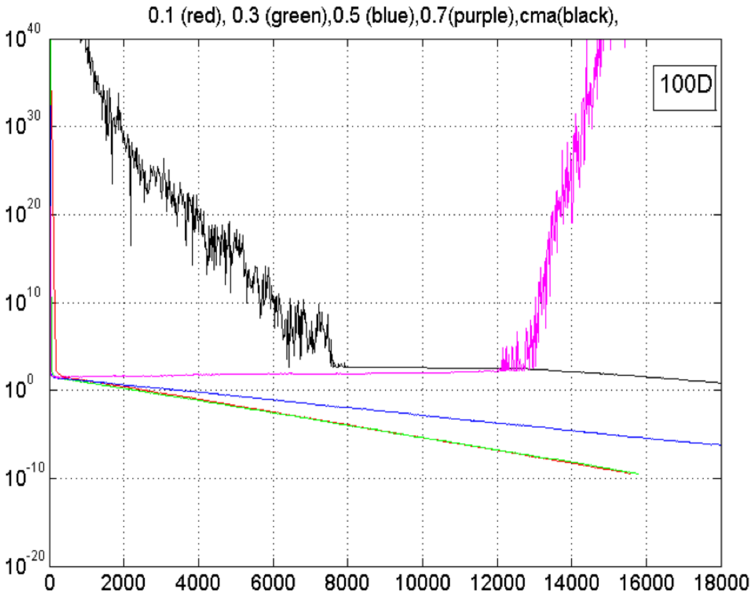

Fig. (9). Different value of α in f2 (100 D).

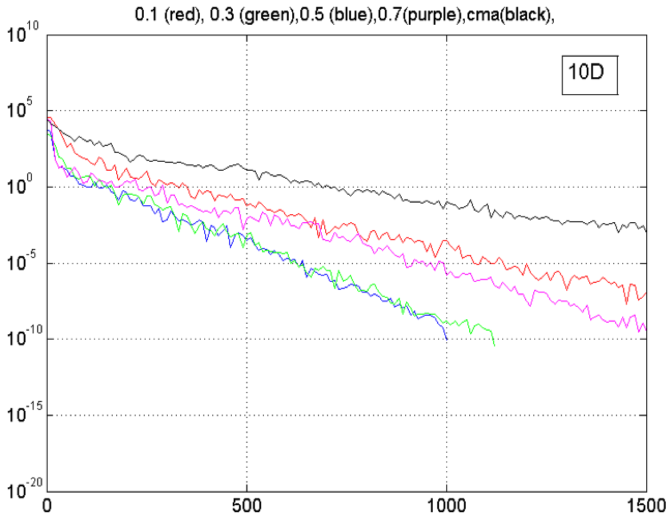

Fig. (10). Different value of α in f3 (10 D).

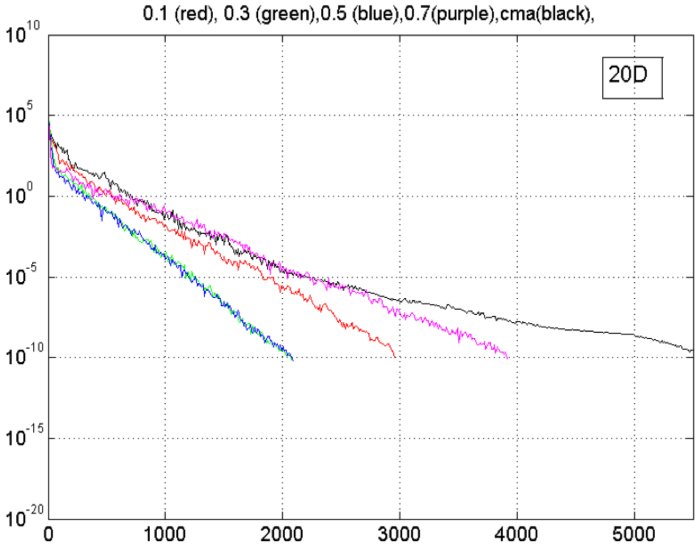

Fig. (11). Different value of α in f3 (20 D).

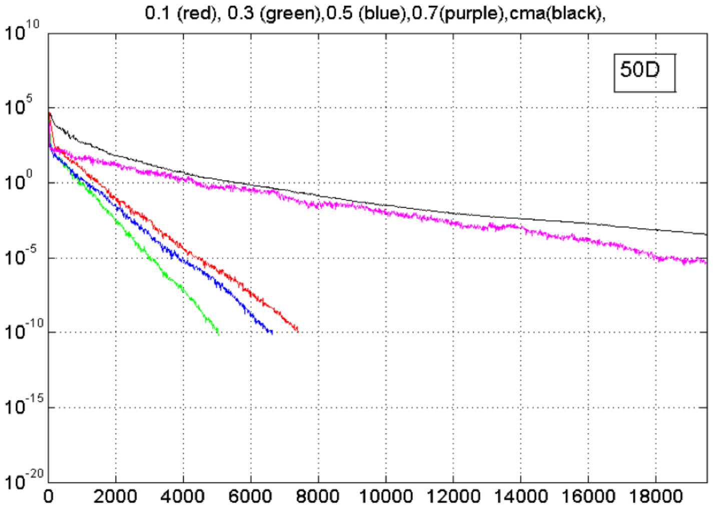

Fig. (12). Different value of α in f3 (50 D)

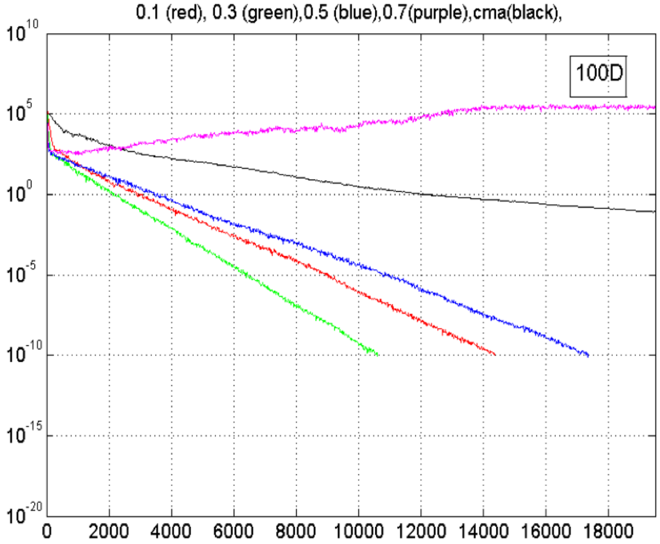

Fig. (13). Different value of α in f3 (100 D).

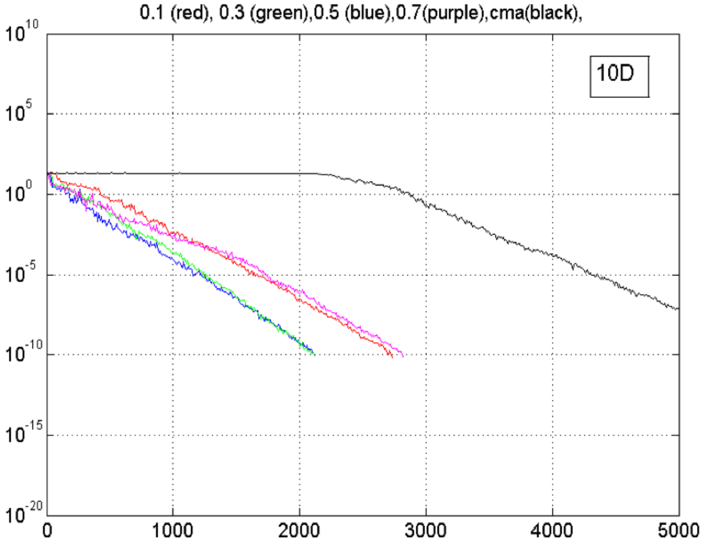

Fig. (14). Different value of α in f5 (10 D).

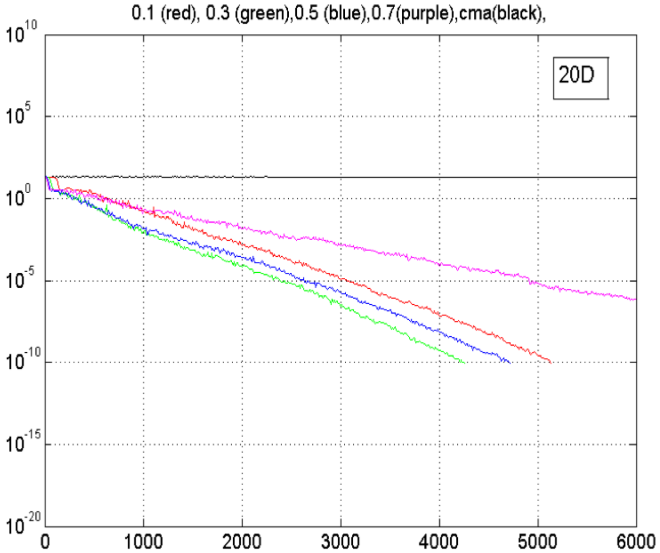

Fig. (15). Different value of α in f5 (20 D).

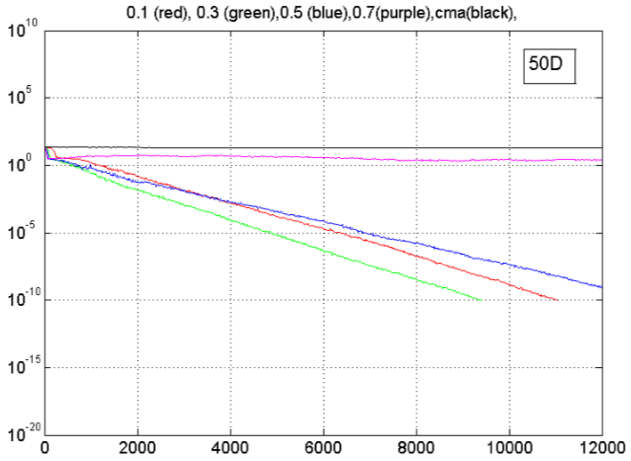

Fig. (16). Different value of α in f5 (50 D).

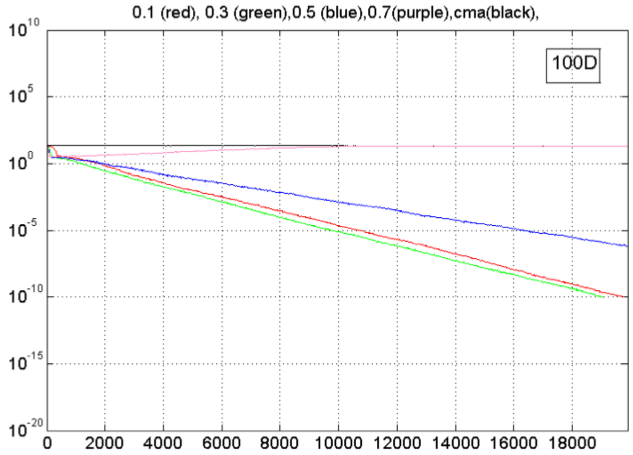

Fig. (17). Different value of α in f5 (100 D).

3.2. The Flow of CMA-mES

The overall algorithm of CMA-mES is:

1. Initialization: The number of generations $G \in N$, the number of individuals $\lambda \in N$, the range of solutions [xmin ,xmax], the dimension of solutions $D \in N$, and the parameters of the algorithm:

1) The mean $\in$[xmin ,xmax];

2) The step size $\sigma \in N$;

3) The offspring number $\mu \le \lambda$;

4) The initial weights w;

5) The measure μ_{eff};

6) The learning rate c_1 and c_μ;

7) The initial covariance matrix C.

2. Circulation:

```
While $I \le G$ do
Sample individuals according to the mean;
Compute the fitness value of individuals;
Select $\mu$ optimal solutions;
Select the best individual;
Update mean with the best individual;
Update the conjugate evolution path $p_\sigma$;
Update the evolution path $p_c$;
Update the covariance matrix;
Update the step size $\sigma$;
Update the global optimal solution;
Update the global optimal fitness value;
I=I+1;
end While
```

3. Output:

The global optimal solution.

4. THE PERFORMANCE TEST

4.1. Experimental Design

In order to test the performance of the proposed algorithm, the original algorithm and some variants (Active-CMA-ES and IPOP-CMA-ES) are used to compare with CMA-mES. Some benchmark functions are selected. The program is run 30 times and took the average value.

The dimensions of the test are [10,20,50,100,200]. The objective functions are given in Table **1**, where D denotes the dimension. The range of the objectives is given in the left column. The parameter values of the algorithms are given in Table **2**.

Table 1. The objective functions.

f1: Sphere Function	
[-0.5 0.5]D	$f(\vec{x}) = \sum_{i=1}^{D} x_i^2$
f2: Schwefel's Problem 2.22 Function	
[-10 10]D	$f(\vec{x}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$
f3: Powell Function	
[-4 5]D	$f(\vec{x}) = \sum_{i=1}^{D/4} \begin{bmatrix} (x_{4i-3} + 10x_{4i-2})^2 \\ +5(x_{4i-1} - x_{4i})^2 \\ +(x_{4i-2} - 2x_{4i-1})^4 \\ +10(x_{4i-3} - x_{4i})^4 \end{bmatrix}$
f4: Quartic Function i.e. Noise	
[-2 2]D	$f(\vec{x}) = \sum_{i=1}^{D} i \times x_i^4 + rand[0,1)$
f5: Ackley Function	
[-40 40]D	$f(\vec{x}) = -20\exp(-0.2\sqrt{\sum_{i=1}^{D} \frac{x_i^2}{D}})$ $-\exp(\sum_{i=1}^{D} (\frac{\cos(2\times\pi\times x_i)}{D}))$ $+20+e$
f6: Generalized Griewank Function	
[-500 500]D	$f(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2$ $-\prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
f7: Generalized Rastrigin Function	
[-5 5]D	$f(\vec{x}) = 10D$ $+\sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)]$
f8: Rotated Hyper-Ellipsoid Function	
[-100 100]D	$f(\vec{x}) = \sum_{i=1}^{D}\sum_{j=1}^{i} x_j^2$
f9: Rotated Hyper-Ellipsoid Function	
[-100 100]D	$f(\vec{x}) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$
f10: Schwefel's Problem 1.2 Noise Function	
[-100 100]D	$f(\vec{x}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_i\right)^2$ $\times(1+0.4\lvert N(0,1)\rvert)$

Table 2. The parameters of the testing algorithms.

Parameters for CMA-ES:

population size=4+floor(3*log(D))
sigma = 0.5
offspring population size =
population size /2;
Dimensions= [50,100,500,1000]

Parameters for Active-CMA-ES:

population size=4+floor(3*log(D))
sigma = 0.5
offspring population size =
population size /2;
negative update of covariance matrix
Dimensions= [50,100,500,1000]

Parameters for IPOP-CMA-ES:

population size=4+floor(3*log(D))
sigma = 0.5
restart=3
offspring population size =
population size /2;
Dimensions= [50,100,500,1000]

Parameters for CMA-mES:

population size=4+floor(3*log(D))
sigma = 0.5
α=0.3
offspring population size =
population size /2;
Dimensions= [50,100,500,1000]

4.2. The Comparison Tests

The results in Tables **3-12** show that the convergence performance of CMA-mES is better than the other algorithm. CMA-mES is an efficient evolutionary algorithm, which greatly increases the search speed in high dimensions.

Table 3. The comparison results of f1.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	2311	7.632e-11	+(2e-10,4e-10)
Active-CMA	2241	7.964e-11	+(4e-10,1e-09)
IPOP-CMA	2481	6.118e-11	+(1e-10,2e-10)
CMA-mES	1172	1.957e-11	+(1e-09,2e-08)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	4513	5.845e-11	+(1e-10,2e-10)
Active-CMA	4345	5.773e-11	+(1e-10,2e-10)
IPOP-CMA	4405	4.523e-11	+(7e-11,2e-10)
CMA-mES	2390	1.186e-11	+(6e-10,2e-08)

(Table 3) contd….

50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	9676	8.934e-11	+(5e-11,8e-11)
Active-CMA	9571	6.590e-11	+(6e-11,1e-10) |
IPOP-CMA	10081	6.533e-11	+(4e-11,1e-10)
CMA-mES	5896	1.910e-11	+(4e-10,6e-09)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	18038	9.4748e-11	+(3e-11,5e-11)
Active-CMA	18157	6.823e-11	+(3e-11,5e-11)
IPOP-CMA	18531	9.235e-11	+(3e-11,8e-11)
CMA-mES	12020	1.976e-11	+(1e-10,9e-10)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	34315	9.946e-11	+(2e-11,3e-11)
Active-CMA	34258	7.684e-11	+(3e-11,5e-11)
IPOP-CMA	35531	6.948e-11	+(1e-11,3e-11)
CMA-mES	24549	1.690e-11	+(2e-10,1e-09)

Table 4. The comparison results of f2.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	4001	7.010e-11	+(1e-10,2e-10)
Active-CMA	4141	6.753e-11	+(8e-11,2e-10)
IPOP-CMA	3991	8.409e-11	+(3e-11,9e-11)
CMA-mES	1821	4.547e-11	+(4e-11,9e-11)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	10525	9.179e-11	+(7e-11,1e-10)
Active-CMA	8197	9.878e-11	+(4e-11,9e-11)
IPOP-CMA	7861	6.241e-11	+(6e-11,9e-11)
CMA-mES	3445	4.700e-11	+(4e-11,1e-10)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	35251	1.973e-10	+(6e-12,1e-11)
Active-CMA	19216	1.790e-10	+(2e-11,3e-11)
IPOP-CMA	18841	1.394e-10	+(3e-11,6e-11)
CMA-mES	8146	1.464e-10	+(2e-11,5e-11)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	148179	7.576e-09	+(2e-11,5e-11)
Active-CMA	126107	6.075e-08	+(1e-11,3e-11)
IPOP-CMA	80481	2.298e-10	+(6e-11,1e-10)
CMA-mES	15556	2.169e-10	+(1e-10,2e-10)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	167201	3.331e-02	+(3e-06,1e-05)
Active-CMA	796101	4.467e-07	+(2e-10,3e-10)
IPOP-CMA	442701	6.073e-03	+(1e-05,3e-05)
CMA-mES	33061	3.111e-10	+(6e-11,1e-10)

Table 5. The comparison results of f3.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	5582	6.219e-11	+(1e-10,3e-10)
Active-CMA	3171	7.503e-11	+(1e-10,7e-10)
IPOP-CMA	4811	9.825e-11	+(4e-12,1e-11)
CMA-mES	1001	3.348e-11	+(1e-09,5e-09)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	20149	9.953e-11	+(5e-12,1e-11)
Active-CMA	13513	9.228e-11	+(4e-11,8e-11)
IPOP-CMA	22285	9.190e-11	+(2e-11,5e-11)
CMA-mES	2077	1.707e-11	+(1e-10,3e-10)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	113206	9.985e-11	+(2e-12,3e-12)
Active-CMA	67246	9.923e-11	+(5e-12,2e-11)
IPOP-CMA	102856	1.167e-10	+(3e-14,8e-14)
CMA-mES	5041	6.290e-11	+(2e-10,4e-10)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	181901	4.815e-08	+(1e-10,2e-10)
Active-CMA	136001	4.276e-08	+(3e-10,1e-09)
IPOP-CMA	215901	5.693e-09	+(2e-10,4e-10)
CMA-mES	10609	1.869e-11	+(8e-11,2e-10)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	463601	6.571e-08	+(1e-09,2e-09)
Active-CMA	372401	5.930e-08	+(1e-10,3e-10)
IPOP-CMA	655976	5.369e-10	+(1e-14,3e-14)
CMA-mES	22839	1.544e-11	+(4e-11,8e-11)

Table 6. The comparison results of f4.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	44001	8.503e-02	+(7e-01,1e+00)
Active-CMA	23101	3.538e-02	+(9e-01,2e+00)
IPOP-CMA	48087	7.551e-03	+(5e-01,1e+00)
CMA-mES	7701	1.763e-03	+(4e-01,9e-01)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	179401	8.2076e-02	+(8e-01,2e+00)
Active-CMA	139101	2.339e-01	+(5e-01,2e+00)
IPOP-CMA	197557	1.267e-02	+(6e-01,1e+00)
CMA-mES	7801	5.885e-04	+(5e-01,1e+00)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	137601	3.690e-01	+(4e-01,1e+00)
Active-CMA	136001	4.987e-01	+(1e+00,3e+00)
IPOP-CMA	27557	1.767e-02	+(6e-01,1e+00)
CMA-mES	94401	2.312e-04	+(2e-01,9e-01)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	242401	9.592e-01	+(5e-01,1e+00)
Active-CMA	34201	9.105e-01	+(6e-01,1e+00)
IPOP-CMA	79548	9.201e-02	+(4e-01,1e+00)
CMA-mES	232201	3.045e-04	+(4e-01,1e+00)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	106001	1.852e+00	+(8e-01,2e+00)
Active-CMA	78001	9.962e-01	+(9e-01,1e+00)
IPOP-CMA	77010	3.911e-01	+(6e-01,1e+00)
CMA-mES	50001	6.850e-03	+(4e-01,8e-01)

Table 7. The comparison results of f5.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	2611	1.991e+01	+(4e-15,2e-14)
Active-CMA	2511	1.979e+01	+(4e-15,7e-15)
IPOP-CMA	1001	1.999e+01	+(2e-05,7e-05)
CMA-mES	2031	5.435e-11	+(1e-10,3e-10)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	5413	1.984e+01	+(4e-15,7e-15)
Active-CMA	4681	1.978e+01	+(0e+00,7e-15)
IPOP-CMA	1201	1.999e+01	+(3e-03,7e-03)
CMA-mES	4261	4.341e-11	+(5e-11,2e-10)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	17566	1.971e+01	+(4e-15,7e-15)
Active-CMA	14311	1.985e+01	+(0e+00,7e-15)
IPOP-CMA	12518	1.999e+01	+(7e-03,2e-02)
CMA-mES	9421	4.234e-11	+(3e-11,2e-10)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	26317	1.978e+01	+(7e-15,1e-14)
Active-CMA	27456	1.974e+01	+(0e+00,7e-15)
IPOP-CMA	12958	1.999e+01	+(3e-03,1e-02)
CMA-mES	18769	3.337e-11	+(3e-11,2e-10)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	51301	2.115e+01	+(1e-01,2e-01)
Active-CMA	34201	1.980e+01	(2e-05,5e-05)
IPOP-CMA	17101	1.999e+01	+(7e-04,2e-03)
CMA-mES	36139	3.979e-11	+(2e-11,8e-11)

Table 8. The comparison results of f6.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	2591	9.160e-11	+(9e-11,3e-10)
Active-CMA	2731	9.344e-11	+(2e-10,3e-10)
IPOP-CMA	2701	4.231e-11	+(9e-11,4e-10)
CMA-mES	1321	5.096e-11	+(7e-09,5e-08)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	5269	8.123e-11	+(9e-11,2e-10)
Active-CMA	4897	7.507e-11	+(1e-10,2e-10)
IPOP-CMA	5329	6.469e-11	+(9e-11,2e-10)
CMA-mES	2257	7.917e-11	+(1e-08,2e-07)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	12886	9.833e-11	+(6e-11,1e-10)
Active-CMA	11956	8.865e-11	+(4e-11,8e-11)
IPOP-CMA	14116	6.230e-11	+(4e-11,1e-10)
CMA-mES	4846	5.656e-11	+(3e-08,2e-07)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	27932	8.924e-11	+(3e-11,5e-11)
Active-CMA	25076	9.457e-11	+(5e-11,8e-11)
IPOP-CMA	28595	8.411e-11	+(3e-11,8e-11)
CMA-mES	10422	8.3450e-11	+(9e-09,6e-08)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	51301	3.519e-09	+(6e-10,1e-09)
Active-CMA	53125	9.899e-11	+(1e-11,4e-11)
IPOP-CMA	59813	8.190e-11	+(2e-11,5e-11)
CMA-mES	22630	9.871e-11	+(4e-09,2e-08)

Table 9. The comparison results of f7.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	2841	5.771e+01	+(7e-15,4e-14)
Active-CMA	3041	4.577e+01	+(0e+00,2e-14)
IPOP-CMA	31487	9.950e-01	+(4e-10,1e-09)
CMA-mES	1061	1.717e-11	+(9e-11,3e-10)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	5257	7.860e+01	+(0e+00,4e-14)
Active-CMA	7261	2.985e+01	+(7e-15,7e-15)
IPOP-CMA	13719	1.896e+01	+(6e-02,2e-01)
CMA-mES	1885	2.567e-11	+(1e-10,2e-10)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	11371	2.647e+02	+(1e-13,3e-13)
Active-CMA	13891	8.756e+01	+(3e-14,6e-14)
IPOP-CMA	65390	2.790e+01	+(4e+00,1e+01)
CMA-mES	4396	1.619e-11	+(6e-11,9e-11)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	26861	3.801e+02	+(6e-14,2e-13)
Active-CMA	20401	3.415e+02	+(2e-05,6e-05)
IPOP-CMA	15301	8.162e+02	+(1e+02,2e+02)
CMA-mES	9555	1.621e-11	+(3e-11,6e-11)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	42580	8.457e+02	+(1e-13,8e-13)
Active-CMA	38001	1.052e+03	+(8e-04,1e-03)
IPOP-CMA	20901	1.256e+03	+(8e-01,2e+00)
CMA-mES	20844	1.585e-11	+(2e-11,5e-11)

Table 10. The comparison results of f8.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	2781	6.857e-11	+(2e-10,3e-10)
Active-CMA	2581	2.356e-11	+(3e-10,6e-10)
IPOP-CMA	2781	6.228e-11	+(2e-10,4e-10)
CMA-mES	1451	1.736e-11	+(4e-10,5e-09)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	5173	8.777e-11	+(5e-11,2e-10)
Active-CMA	5257	9.807e-11	+(1e-10,2e-10)
IPOP-CMA	5569	2.869e-11	+(9e-11,2e-10)
CMA-mES	2473	8.515e-11	+(2e-09,1e-08)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	15676	8.596e-11	+(5e-11,8e-11)
Active-CMA	14566	9.979e-11	+(6e-11,1e-10)
IPOP-CMA	15136	7.295e-11	+(5e-11,9e-11)
CMA-mES	6796	1.213e-11	+(1e-09,7e-09)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	37469	9.156e-11	+(3e-11,5e-11)
Active-CMA	36109	9.490e-11	+(2e-11,5e-11)
IPOP-CMA	36755	7.349e-11	+(2e-11,4e-11)
CMA-mES	13686	1.923e-11	+(3e-10,4e-09)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	100758	9.912e-11	+(1e-11,2e-11)
Active-CMA	92398	9.977e-11	+(9e-12,3e-11)
IPOP-CMA	97433	8.546e-11	+(2e-11,4e-11)
CMA-mES	29774	1.869e-11	+(4e-11,5e-10)

Table 11. The comparison results of f9.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	811	4.000e+00	+(3e+00,9e+00)
Active-CMA	791	0.000e+00	+(4e+00,6e+00)
IPOP-CMA	961	0.000e+00	+(3e+00,3e+00)
CMA-mES	171	0.000e+00	+(3e+01,1e+02)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	1657	3.000e+00	+(1e+00,3e+00)
Active-CMA	1693	0.000e+00	+(4e+00,6e+00)
IPOP-CMA	5031	0.000e+00	+(3e+00,7e+00)
CMA-mES	289	0.000e+00	+(5e+01,1e+02)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	4081	2.000e+00	+(1e+00,3e+00)
Active-CMA	4291	0.000e+00	+(2e+00,4e+00)
IPOP-CMA	4156	0.000e+00	+(4e+00,9e+00)
CMA-mES	451	0.000e+00	+(5e+01,3e+02)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	7209	2.000e+00	+(1e+00,5e+00)
Active-CMA	8501	1.000e+00	+(1e+00,3e+00)
IPOP-CMA	41689	0.000e+00	+(5e+00,1e+01)
CMA-mES	783	0.000e+00	+(4e+01,2e+02)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	14536	9.000e+00	+(2e+00,5e+00)
Active-CMA	20901	1.100e+01	+(1e+00,2e+00)
IPOP-CMA	39466	0.000e+00	+(4e+00,8e+00)
CMA-mES	1540	0.000e+00	+(2e+01,1e+02)

Table 12. The comparison results of f10.

10D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	5371	7.967e-11	+(6e-11,2e-10)
Active-CMA	3611	9.326e-11	+(3e-10,4e-10)
IPOP-CMA	14675	1.337e-11	+(3e-10,7e-10)
CMA-mES	1181	5.182e-11	+(5e-10,6e-09)
20D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	15385	9.910e-11	+(3e-10,5e-10)
Active-CMA	28309	5.709e+03	+(8e+02,5e+03)
IPOP-CMA	30161	1.610e-11	+(1e-10,4e-10)
CMA-mES	2641	4.262e-11	+(4e-09,5e-08)
50D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	99001	3.447e+04	+(8e+03,2e+04)
Active-CMA	29446	9.042e+04	+(3e+04,7e+04)
IPOP-CMA	19983	4.198e+08	+(1e+08,5e+08)
CMA-mES	6106	7.308e-11	+(4e-09,4e-08)
100D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	112201	4.291e+05	+(1e+05,4e+05)
Active-CMA	78201	3.907e+05	+(8e+04,4e+05)
IPOP-CMA	57346	3.364e+09	+(7e+08,3e+09)
CMA-mES	12887	2.795e-11	+(3e-09,1e-08)
200D			
Algorithm	**Fevals**	**Function Value**	**(median, worst)**
CMA-ES	163401	8.824e+05	+(1e+05,6e+05)
Active-CMA	89301	8.361e+05	+(3e+05,5e+05)
IPOP-CMA	32301	2.733e+10	+(7e+09,2e+10)
CMA-mES	27665	2.598e-11	+(2e-09,4e-09)

5. CASE STUDY

In order to demonstrate that the proposed CMA-mES algorithm can be applied in the network security situation prediction, a case is studied in this Section.

5.1. Problem Formulation

As mentioned in section 1, the network security situation is very important to a network platform. In order to obtain the real network security situation, a network platform is designed, as shown in Fig. (**18**).

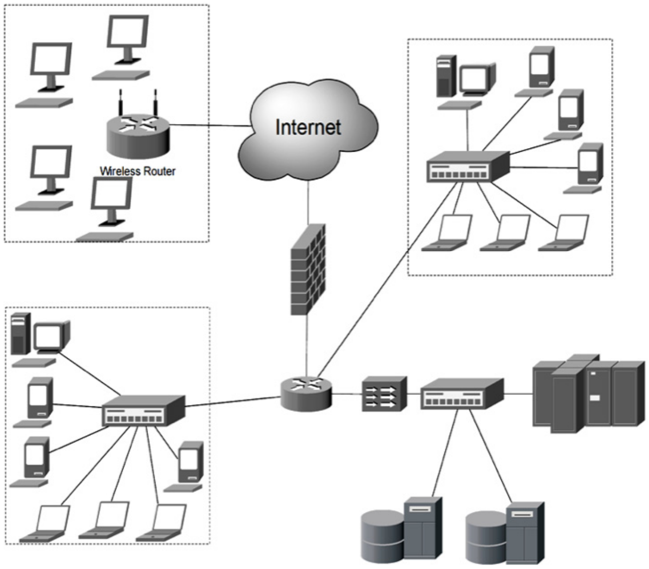

Fig. (18). A network platform.

The attack events of the network platform are collected during 51 days, and the hierarchical evaluation method [21] is used to get the network security situation values. Then the values are normalized by:

$$x(t) = \frac{x(t) - x_{\min}}{x_{\max} - x_{\min}} \tag{15}$$

The collected network security situation is described in Fig. (**19**).

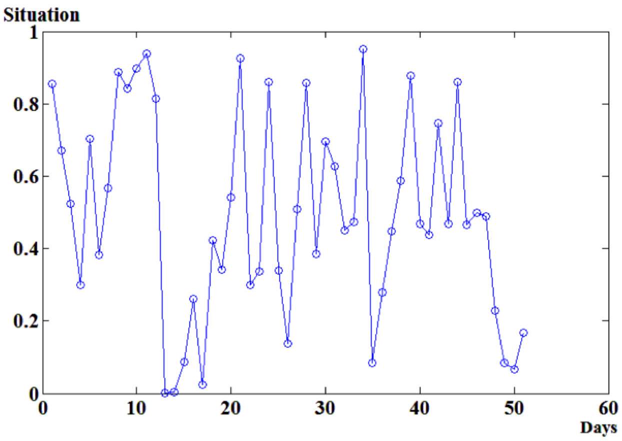

Fig. (19). The network security situation values during 51 days.

A BP neural network is established as the forecasting model of network security situation. The structure of the BP neural network is as shown in Fig. (**20**).

The input training data can be generated by the following rule [22]:

$$\begin{aligned} x(1)_{input} &= [x(1), x(2), \cdots, x(k)] \\ x(2)_{input} &= [x(2), x(3), \cdots, x(k+1)] \\ &\cdots\cdots \\ x(T)_{input} &= [x(T), x(T+1), \cdots, x(T+k-1)] \end{aligned} \tag{16}$$

Where x(t) denotes the situation in the first 51 days, T=49, k=3. x(t) denotes the input training data.

The output training data can be generated by the following rule:

$$y(i)_{output} = x(k+i) \qquad i \in [1, \cdots, T] \tag{17}$$

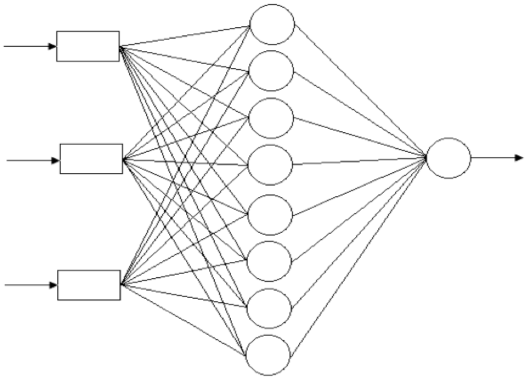

Fig. (20). The BP neural network model for forecasting the network security situation.

5.2. Predict Network Security Situation

The forecasting model shown in Fig. (**20**) and the training data shown in Eqs. 16 and 17 are used to predict the network security situation, the result is shown in Fig. (**21**), the mean squared error, MSE=4.0 × 10^{-3}.

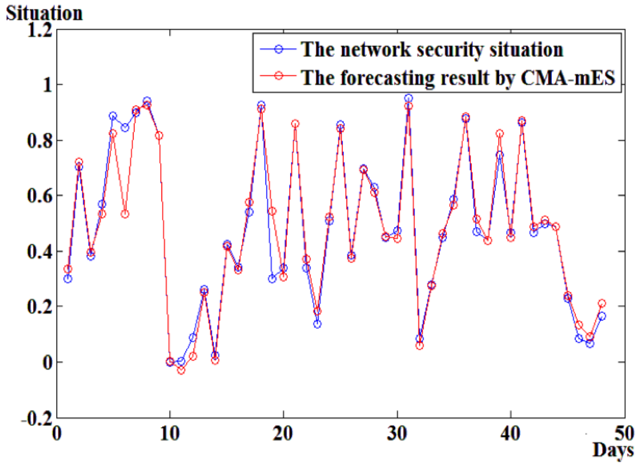

Fig. (21). The forecasting results by the CMA-mES algorithm.

CONCLUSION

In this paper, an efficient improvement of the CMA-ES algorithm is proposed, which is called CMA-mES, and it is used to optimize the forecasting model of the network security situation. The CMA-mES algorithm changes the updating strategy of the mean by adding a component to the recombination. It improves the convergence speed and the high dimensional search ability of the original algorithm.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

Pei-Li Qiao thanks the partial support by NSFC under Grant 61103149.

REFERENCES

[1] T.Bass, “Intrusion detection system and multi-sensor data fusion: Creating cyberspace situation awareness,”*Communications of the ACM*,vol.3, no. 4,pp.99-105, 2000.

[2] B. Chris , and H. Cameron , “Active Defense: A comprehensive guide to network security,” *SYBEX Alameda*, 2001.

[3] K.D. Hoon, L Taek, O.J.Sung, I.P. Hoh, and L.H. Jo, “Cyber threat trend analysis model using HMM,” Third International Symposium on Information Assurance and Security, pp. 177-182,2009.

[4] B.Y. Zhang, Z.G. Chen, S.L. Wang, X.A. Yan, D.X. Zhang, and Q. Fan, “Network security situation assessment based on HMM,”*Advanced Intelligent Computing Theories and Applications ,*vol. 6839, pp.387-394,2012.

[5] P. Baruah and R. B.Chinnam, “HMMs for diagnostics and prognostics in machining processes,” *Failure Prevention Technol. Conference Virginia Beach*, pp.14–18, 2003.

[6] Z.J. Zhou, C.H. Hu, D.L. Xu, M.Y. Chen, and D.H. Zhou, “A model for real-time failure prognosis based on hidden Markov model and belief rule base,” *European Journal of Operational Research,* vol.207, pp. 269-283, 2010.

[7] J.B. Lai, H.Q. Wang, X.W. Liu, Y. Liang, R.J. Zheng,and G. S. Zhao, “WNN-Based network security situation quantitative prediction method and its optimization,” *Journal of Computer Science and Technology*,vol.23, no. 2, pp. 222-230, 2008.

[8] M. Zhou, and S.D. Sun, “*The principle and application of genetic algorithm*,” National Defense Industry Press, 2001.

[9] J. Kennedy, “Particle swarm optimization,” *In: Encyclopedia of Machine Learning, Springer,*pp. 760-766, 2010.

[10] R. Storm, and K. Price, “Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces,”CSI Berkeley, 1995.

[11] N. Hansen, J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, “Towards a new evolutionary computation,” *Advances on Estimation of Distribution Algorithms*, pp. 75-102,2006.

[12] N. Hansen,and S. Kern, “Evaluating the CMA evolution strategy on multimodal test functions,” *Parallel Problem Solving from Nature - PPSN VIII*, vol.3, pp. 282-291,2004.

[13] N. Hansen, S.D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the deran-domized evolution strategy with covariance matrix adaptation (CMA-ES),” *Evolutionary Computation,* vol.11, no. 1, pp.1-18, 2003.

[14] A. Auger, and N. Hansen, “*Benchmarking the (1+1)-CMA-ES on the BBOB-2009 function tested*,” Proceedings of the Genetic and Evolutionary Computation Conference (GECCO ACM)Press, 2009.

[15] S. Kern, S.D. Müller, N. Hansen, D. Büche, J. Ocenasek and P. Koumoutsakos, “Learning probability distributions in continuous evolutionary algorithms-a comparative review,” *Natural Computing,* vol.3, no.77-112, 2004.

[16] A.Auger, and N. Hansen, “A restart CMA evolution strategy with increasing population size,” *Proceedings on IEEE Congress on Evolutionary Computation,* pp.1769-1776, 2005.

[17] C. Igel, T. Suttorp, and N.Hansen, “A computational efficient covariance matrix update and a (1+1)-CMA for Evolution Strategies,”*Proceedings of the Genetic and Evolutionary Computation Conference (GECCO*), pp.453-460, 2006.

[18] G.A.Jastrebski, and D.V. Arnold, “Improving evolution strategies through active covariance matrix adaptation,” *IEEE World Congress on Computational Intelligence, Proceedings*, pp.9719-9726, 2006.

[19] C. Igel, N. Hansen, and S. Roth, “*Covariance matrix adaptation for multi-objective optimization*,”Evolutionary Computation M-IT press, vol. 15, no.1,pp.1-28, 2007.

[20] N. Hansen, “Benchmarking a BI-population CMA-ES on the BBOB-2009 function tested,” *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, 2009.

[21] X.Z. Chen, Q.H. Zheng, X.H. Guan, and C.G. Lin, “Quantitative hierarchical threat evaluation model for network security,” *Journal of Software*, vol.17, no. 4, pp. 885-897,2006.

[22] W. Ren, X.H. Jiang, and T.F. Sun, “RBFNN-based prediction of network security situation,”*Computer Engineering and Applications,* vol.31,pp.136-144, 2006;

Received: May 26, 2015 Revised: July 14, 2015 Accepted: August 10, 2015