An Optimized FPN Network Attack Model Based on Improved Ant Colony Algorithm

Huilin Wu[1,a], Wenjuan Wu [2,b*]

[1]Hebei Academy of Sciences Institute of Applied Mathematics, Shijiazhuang 050081 China
[2]School of Information, Renmin University of China, Beijing , 100872 China
[a]wuhuilin@heb-math.org, [b]wuwj@ruc.edu.cn

Keywords: Fuzzy Petri net; net attack model; ant colony algorithm; BP algorithm

Abstract: FPN attack model can be widely applied to a variety of large and complex network environments. In this paper, we present an optimized FPN network attack model based on the improved ant colony algorithm. First, We apply the ant colony algorithm to the FPN network attack model to optimize the training procedure of weight parameters. Second, we introduce hybridizing and aberrance gene to the algorithm to improve the converging rate and global search capability. Experiments show that our algorithm achieves higher accuracy and faster convergent rate.

Introduction

Petri net-based network attack model was firstly proposed by Mc Demott. It's suitable to describe and analysis the asynchronous, concurrent, resource competition and other issues in the large and complex systems. As the network attack has the characteristics of uncertainty and concurrent, the Petri net is extended to fuzzy Petri net (FPN). However, the parameters of FPN, such as weight, certainty and threshold factors depends on the experience of experts at a large extent, so it is hard to obtain accurate results.

To solve this problem, the research results on parameter optimization in the artificial intelligence domain is worth learning. For example, Huang propose a PID control method, which utilize the property of the BP neural network algorithm that can approximate any continuous bounded nonlinear function[4]. The PID control method can adapt to the dynamic nature of uncertain systems. Peng improve the method and presents a method of optimized PID parameter self-adapted ant colony algorithm with hybridizing and aberrance gene, based on ant colony algorithm[5]. This method overcomes ordinary ant colony algorithm's defects of slow convergence speed, easy to get stagnate, and low ability of full search, and can realize the optimization of PID control parameter perfectly. Currently, some researchers attempt to introduce the neural network algorithm to the parameter optimization problem of FPN network attack model. However, the results are not good. There are 2 reasons. Because of the special structure character of FPN network attack model, the original neural network algorithms can't adapt to the model perfectly. In addition, these algorithms' own parameter optimization strategies have defects.

Aiming these problems , in this paper we present an optimized FPN network attack model based on the improved ant colony algorithm. First, we analysis the inference algorithm of FPN thoroughly, and apply the ant colony algorithm to FPN. Second, giving full consideration to the distinguishing characteristics of FPN attack model, we introduce hybridizing and aberrance gene to the algorithm to improve the converging rate and global search capability.

The rest of the paper is organized as follows. Chapter 2 introduce the attack model. In Chapter 3, we describe the details of the method. The results of experiments are shown in Chapter 4 and this study is concluded in Chapter 5.

Attack Model

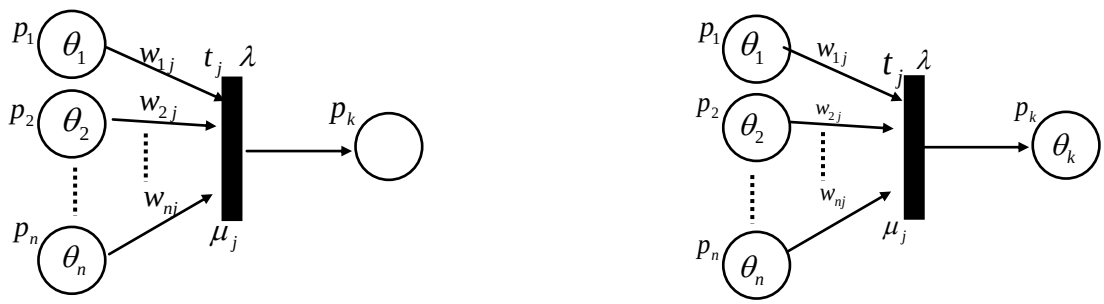
Fuzzy Petri net(FPN)[2] is a library and two kinds of nodes changes bidirectional directed graph. FPN structure is defined as a 9-to Element: $FPN = (P, T, D, I, O, F, \theta^0, W, \Gamma)$ 。

Wherein, $PN = (P, T, D, I, O)$ is a basic Petri net. $F = \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_m)$ is the confidence matrix, μ_j is a fuzzy number between [0,1] interval, and it represents the confidence of fuzzy rule t_j; $\Gamma = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$, $\lambda_i \in [0,1]$, λ_j is the initial threshold of t_j; θ^0 is the initial state of the proposition Dependability, $\theta^0 = (\theta^0_{d1}, \ldots, \theta^0_{dn})T$, θ^0_{di} is the initial logic state of Proposition d_i, $\theta^0_{di} \in [0,1]$, d_i of showing the true extent of the proposition; $W = \{w_{ij}\}, w_{ij} \in [0,1]$, Representation from the library p_i to the changes t_j connection arc given appropriate weight. That is, different p_i importance of different changes t_j trigger.

$$w_{ij} = \begin{cases} 0, & p_i \notin g_{tj} \\ (0,1], & p_i \in g_{tj} \end{cases} \tag{1}$$

Each transition in FPN corresponds to a rule, while the input place and output place of a transition is the precondition and conclusion respectively. A transition occurrence means a corresponding rule matches successfully. The general form of fuzzy production rules are as follows:

Rule 1: If p_1 and p_2 and … and p_n Then p_k $(CF = \mu)$, $\lambda, w_{1j}, w_{2j}, \ldots, w_{nj}$. It can be formally described as Fig. 1.



a) Change Trigger before

b) Changes triggered

Fig. 1 inference rule 1 of FIG.

The confidence of the conclusion of Rule 1 is: $\Theta(p_k) = \mu_j \times \sum_{j=1}^{n} \Theta(p_i) \times w_{ij}$, $\sum_{j=1}^{n} \Theta(p_i) \times w_{ij} \geq \lambda$

Rule 2:If p_1 or p_2 or … or p_n, then p_k $(CF = \mu_i)$, λ formal description shown in Fig. 2.

a) Change Trigger before

b) Changes triggered

Fig. 2 inference rule 2

The confidence of the conclusion of Rule 2 is:

$\Theta(p_k) = \max(\mu_1 \times \Theta(p_1), \ldots, \mu_n \times \Theta(p_n)), \Theta(p_i) \geq \lambda_i$;

Fuzzy Petri net is a good modeling tool to describe the fuzzy production rules in knowledge systems [1], a drawback fuzzy system itself is poor learning ability. FPN inherited graphical depiction ability to represent knowledge, and the representation is simple and clear; it also has the ability for fuzzy reasoning [3], commonly used in the knowledge analysis, testing, and decision support, etc. However, like other fuzzy systems, the parameters of FPN, such as weight, certainty and threshold factors depends on the experience of experts at a large extent, so it is hard to obtain accurate results. As a result, it brings large difficulty to the knowledge inference. To solve this problem, we make use of the improved ant colony algorithm to optimize the parameter estimation procedure of FPN network attack model.

Improved ant colony algorithm to optimize the use of network attack model FPN

Improved Ant Colony Algorithm

The ant colony algorithm is inspired from ants foraging behavior. The mechanism is to mimic ants foraging process, through the social partnership, each ant leaves chemical pheromones, the boot according to the concentration of odor ants probabilistic selection foraging path to achieve global search results. Ant colony algorithm is a global optimization, parallel positive feedback heuristic algorithm, using it to train FPN parameters, may have a stronger binding. But there are ant colony algorithm is easy to fall into the traditional search standstill, the computing time is longer, easy to fall into local optimum problem for large complex and not well suited FPN network.

To solve the above problem, this paper proposes the introduction of Crossover and mutation operator to enhance an ant colony to find solution. This method can make full use of existing pheromone exchange between the ants get information about the route, and crossover and mutation to shorten the search time and reduce the possibility into a local optimum. And while the

introduction of add candidate set thinking, reduce the scope of the search, improve accuracy, so that it can adapt to large-scale cyber attacks FPN complex models.

Algorithm steps

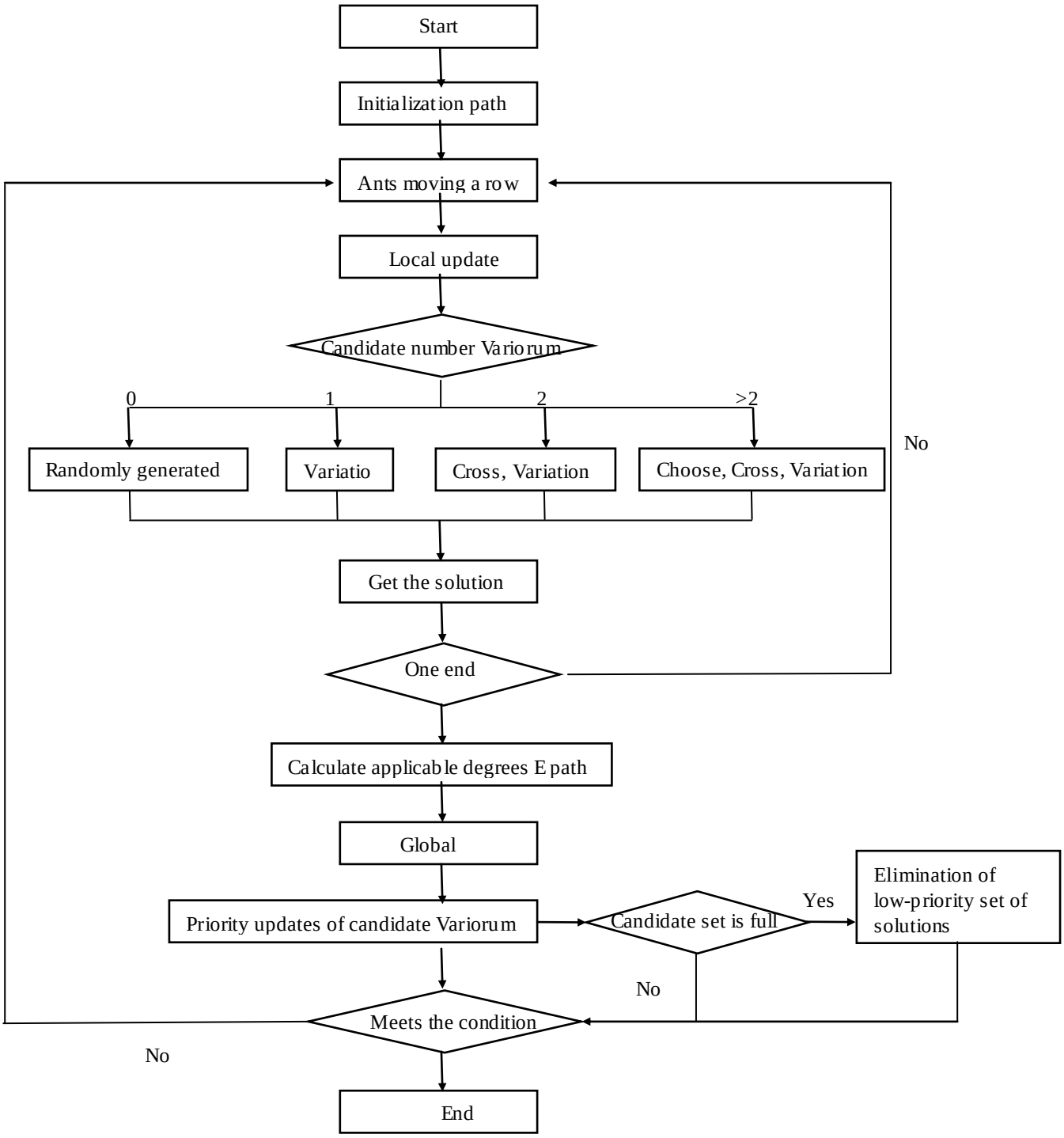


Fig. 3 algorithm flowchart

The first step: initialize

20 randomly generated initial solution, the solution of each component is calculated that 20 belongs to the sub-interval (City), resulting in various cities of the candidate group, calculated on each side (the connection between two cities), the fitness function of information the amount.

Step two: iterative process

While (the number of optimization is more than 400 times or fitness value $<1.0\times10^{-5}$){

```
For(i=1 to 17){          ///17 solution components
    For(j=1 to 20){      ////20 ants
    Determine the value of the i-th component of the falls from 1 to 10 cities and edge
  information based on  q0  roulette method(3.1)
    Local Update on the side of the pheromone (3.2)
    The group carried out the candidate selection, crossover and mutation to produce new
  value - in selected cities. (3.3)
    }
}
For(j=1 to 20){
    Calculate the fitness function of each ant
}
Value update information in accordance with the fitness of each side of (3.4)
Take high fitness value updating the city's candidate set (3.5)
}
```

Structural body (City) on continued algorithm fitness function is an error cost function E, the city's candidate set is a component of the solution can save 20 value and the fitness value, while refers to the neighboring city of about wiring on behalf of i component range selected direction. The continued algorithms (3.1) Select the i-th component in accordance with the following formula where the interval value (City) j, j = 1,2, ..., 10

$$j=\begin{cases}\arg\max\{\tau_{ij}\,|1\le j\le 10\}, & \text{if } q\le q_0\\ j_0, & \text{otherwise.}\end{cases} \tag{2}$$

Argmax which represents the largest concentration of selected information edge. q is a random number [0,1], q_0 is a constant, it is preferable 0.8, in order to avoid premature halt. j_0 probability represented by the following formula between 1 to 10 representing the selected value from 1 to 10 cities, including τ_{ij}(t) represents the time t, i and j number of urban edge information, it dynamically.

$$p_{ij0}^{k}(t)=\tau_{ij0}(t)\,/\sum_{k=1}^{k_i}\tau_{ik}(t) \tag{3}$$

The continued algorithms (3.2) line selected subinterval partial update of information on the selected sub-interval immediately reduce the amount of information as appropriate, so that the probability of other ants in the lower sub-interval selected. Let the k-th individual of the i-th component of the j-th city is selected, press the edge of the partial update type information, which by the test taken after a volatile factor 0.2.

$$\tau_{ij}(\mathrm{t}) = (1-\eta)\square\tau_{ij}(\mathrm{t}) + \eta\square\min\{\tau_{ij}(\mathrm{t}) \mid 1 \le \mathrm{j} \le 10\} \tag{4}$$

Thus, the amount of information updated and relevant information is the original i-th component of the sub-section convex combination of the minimum amount of information. After the most informative sub-interval is repeatedly selected, the amount of information to reduce the amount of information to the average level of 10 sub-section, so that the probability of other ants choose subintervals increase, that increase understanding of diversity, while effectively reducing the stagnation phenomena occur.

The continued algorithm (3.3) of the candidate set of genetic manipulation:

1) Random function corresponding to the operation rand(uper), rnd(int low, double uper), rnd(double low, double uper)。

2) When operating in the candidate selection group, with "roulette wheel" approach to fitness function E value selects two values, the first value j probability of being selected is: $E_j / \Sigma E_i$, which represents the candidate group ΣE_i All values fitness values.

3) In the crossover operation, the setting selected two value $x_{(1)}$ and $x_{(2)}$, the corresponding function values were E_1 , E_2 , and $E_1 < E_2$, we have to cross the probability p_c operation. Randomly generated $p \in [0,1]$,if $p > p_c$, crossover operation is performed. p_c value should be dynamic, Because of randomly generated initial population diversity, in order to improve the convergence rate, crossover rate should As the optimization process increases, in order to avoid early convergence, the probability of crossing should be reduced using the following formula. N is the current number of iterations, M is the total number of iterations.

$$p_c = e^{-0.5\times N/M} \tag{5}$$

In the optimization of the process should take a cross-random number $r \in [0,1]$, cross the resulting value $X_c = x_{(1)} + r \times [\mathrm{x}_{(2)} - x_{(1)}]$; if $p < p_c$, crossover operation is not performed, Take $X_c = x_{(1)}$.

4) In the phase variation can be the probability p_m result of the operation of cross- X_c mutating get X_m . p_m value should be dynamic, initially be relatively small, with the progress of

the optimization process of mutation probability increases to ensure the diversity of the population. p_m the following formula：

$$p_m = e^{0.1\times N/M} - 1 \tag{6}$$

The k-th component of the i-th interval as: $[(k-1)\times 0.1, k\times 0.1)$. Let $d_i = \max\{k\times 0.1 - X_c,\ X_c - (k-1)\times 0.1\}$, generates a random number $\delta \in [-1,1]$, the value of X_m taken following formula:

$$x_m = \begin{cases} x_c + \delta\times d_i & (k-1)\times 0.1 - x_c \le \delta\square d_i \le k\times 0.1 - x_c \\ x_c - \delta\times d_i & \text{otherwise} \end{cases} \tag{7}$$

This will ensure that the results of genetic manipulation is still in the sub-interval.

The continued algorithm (3.4) in the M ants obtain M solution, according to the following formula for the amount of information on each path for updates:

$$\tau_{ij}(\mathrm{t}+1) = \rho\times\tau_{ij}(\mathrm{t}) + \Delta\tau_{ij} \tag{8}$$

Where $\Delta\tau_{ij}$ number for all i and j are numbers through the city on this side of the ant pheromone sum. Here the first i ant after an edge is left $\omega\times Q/f_i$ pheromones, pheromone through the test to take the residual coefficient ρ = 0.5, w = 0.8, because the program is so involved in the floating-point operations take Q = 0.0001, f_i fitness function values.

The continued algorithms (3.5) when 20 ants once get 20 complete optimization and adaptation values. The candidate in each city group fitness sorted by value, determine whether you need to update the candidate group.

This chapter ant colony and genetic combine the advantages of both proposed a new optimization algorithm, and elaborated on the specific implementation steps, the new algorithm performance analysis of time and space, show that the new algorithm is feasible, easy to implement.

Experiments

In this paper, in recent years, the dangers of a large Botnet attacks, for example, to verify the algorithm described above. Fig. 4 Botnet attack with weights FPN process. Wherein p1 to p10 denote different attack state. Wherein, p1: Host dangerous open ports; p2: host does not open the firewall; p3: the host is connected (to allow data exchange); p4: the host does not start the patch management; p5: system has overflow; p6: attacker hosts with User privileges; p7: an attacker to gain access to the host computer; p8: Host allows remote management; p9: hosts are implanted Bot program; p10: Bot program and the host is in the worm infected. t1 to t5 denote different attacks. Where, t1: elevation of privilege; t2: Host exploits; t3: User permissions to other hosts in Bot implantation procedure; t4: Bot remote implantation procedure; t5: Bot host downloads the worm

infection to the machine. And set an ideal weight W = {0.4,0.3,0.3,0.25,0.35,0.4,1,0.5,0.5,1}. 20 batch sample data FPN model train, where b = 5000, the number of initial solution of 20 ant populations of 20, not more than 400 times the number of iterations, the maximum error of accuracy, control 1.0×10^{-6} or more. Above algorithms are used in C ++ programming, algorithms running in the Xeon 2.7 graphics workstations. Through the above data and algorithms of programming, the experimental data obtained can be analyzed and compared the following aspects.

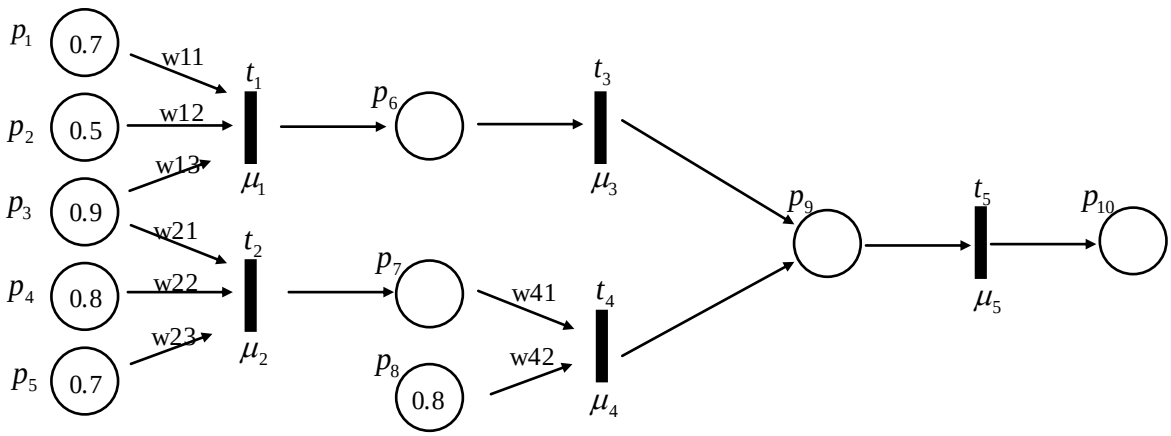
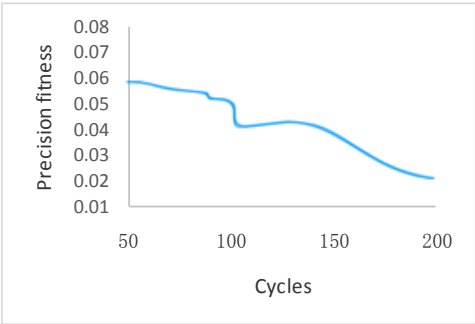
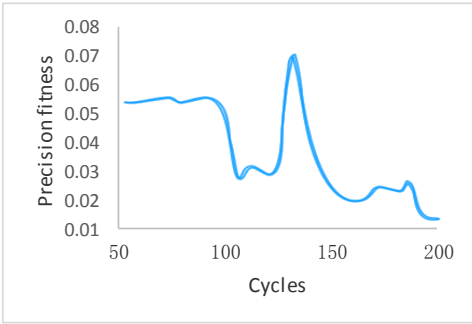


Fig. 4 FPN weighted value represents Botnet attacks of FIG.

Comparison of evolution: the mean square error and (MSE) in this experiment to verify the desired output value and output value by the precision error between after learning through the magnitude of the error value to prove the effectiveness of the learning process. BP algorithm training process reflects the fitness of MSE evolution graph in Fig. 5 (a). Figure fitness function error accuracy 10^{-4}. Fig. 5 (b) is improved ant colony algorithm fitness FIG. Seen from the figure, the fitness function value in leaps and approaching the low error, mainly because of crossover and mutation factor can reduce convergence time, and prevents ant colony into a local optimum, but also can rapidly approaching global convergence direction.

Convergence time analysis: with cross, new ant colony algorithm variation factor prescribed by the 20 ants each iteration participating in the search, set the taboo table records in the program can not take the path of the ants, they can not restrict the same path the repeated optimization, to ensure that the optimization efficiency, while there is also a candidate set of greatly reduced the scope of the search to ensure that the optimization results fast close to the optimal solution in a finite number of times, effectively avoiding the ant colony into local search standstill state, the delay due to genetic operator generated substantially negligible, a substantial increase in the efficiency of the program. Since BP algorithm on programming data structure involves a large number of matrix operations, calculate time-consuming large. Fig. 6 shows the 200 times in the iterative time-consuming comparison of the two algorithms.





(a)BP algorithm (b)Improved ant colony algorithm

Fig. 5 MSE evolution curve

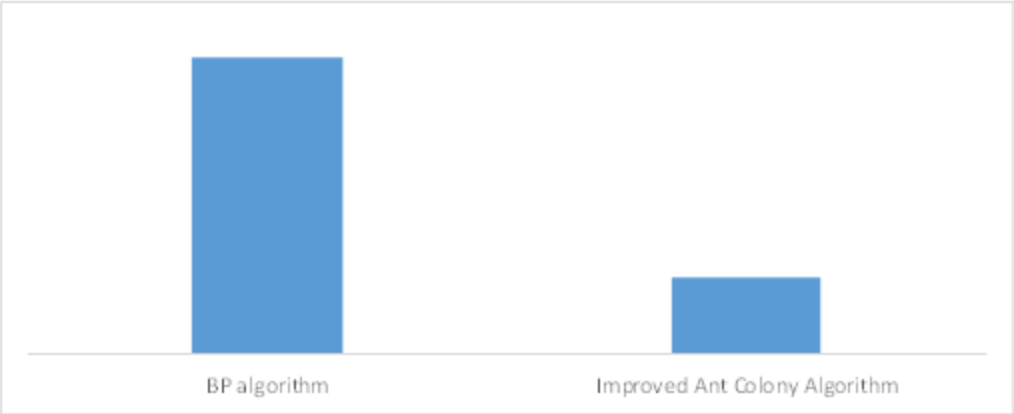


Fig. 6 Comparison of running speed

Conclusion

In this paper, we present an optimized FPN network attack model based on the improved ant colony algorithm. Fully considering the distinguishing nature of FPN, we apply the ant colony algorithm to the FPN network attack model. The method optimize the weight parameter estimation problem and can better adaptive to the complex network attack model. Furthermore, hybridizing and aberrance gene are introduced to the algorithm to improve the converging rate and global search capability. The results of simulation experiments which compare to the result of BP algorithm shows that our method can better adapt to the complex FPN network attack model , achieves higher accuracy and faster convergent rate. This paper is supported by the project of research and implementation of mobile health monitoring information platform, project number: 20150503LR62-4

Reference

[1] Fay A.A fuzzy knowledge2based system for railway traffic control Engineering Applications of Artificial Intelligence .13(2000)178-193.

[2] Looney C. G. Fuzzy Petri nets and application. In : Tzafestas S. G. et al. Fuzzy Reasoning in Information , Decision and Control Systems Norwell ,MA: Kluwer Academic Publishers.(1994) 511-527.

[3] Eugenia Minca Daniel Racoceanu Noureddine Zerhouni. Monitoring Systems Modeling and Analysis Using Fuzzy Petri Nets[J].Studies in Informatics and Control. 11(2002)331-338.

[4] Jinyan Huang . Study PID control method based on BP neural network [J]. Microcomputer Information. 22 (2006).278-280.

[5] Peifu Peng. Adaptive hybrid ant colony algorithm, the optimal PID parameter variation factor [J]. Computer Engineering and Applications.6(2006)88-91.