The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
21-23 September, 2017, Bucharest, Romania

An Improved Grey Wolf Optimizer Algorithm Integrated with Cuckoo Search

Hui Xu, Xiang Liu, Jun Su
Hubei University of Technology, Wuhan 430068, China, xuhui@mail.hbut.edu.cn, www.hbut.edu.cn

***Abstract*—Grey Wolf Optimizer (GWO) is a new meta-heuristic optimization. It is inspired by the unique predator strategy and organization system of grey wolves. Since the GWO algorithm is easy to fall into local optimum especially when it is used in the high-dimensional data, an improved GWO algorithm combined with Cuckoo Search (CS) is proposed in this paper. By introducing the global-search ability of CS into GWO to update its best three solutions that are alpha_wolf, beta_wolf and delta_wolf, the search ability of GWO is strengthened, and the shortcoming of GWO is offset. Preliminary experimental analysis validates that, the propsoed CS-GWO algorithm has a stronger global-search ability, and might avoid to fall into the local optimum and jump out of the local optimum in high-dimension datasets, compared with both the original GWO algorithm and the original CS algorithm.**

Keywords—Grey Wolf Optimizer; Cuckoo Search; integration

I. INTRODUCTION

Although the creatures of nature are not as intelligent as human beings but through the group cooperation they show excellent group intelligence. Seyedali Mirjalili [1] proposed Grey Wolf Optimizer (GWO) in 2014. It was inspired by unique predator strategy and organization system of grey wolves. Recent years, the GWO algorithm has been widely used in many aspects. Ibrahim A. Hameed [2] applied it to offshore crane design area. By using GWO to setup arguments in design procedure, it can enhance the working capacity of the offshore crane and reduce the offshore all-up weight. Totok Ruki Biyanto [3] applied GWO to find the best operating equipment parameters in natural gas process area. Jitendra Kumar Seth [4] used binary GWO for feature selection in intrusion detection area.

Although the GWO algorithm has been widely used in many areas, but it also has some defects. During the iteration, the global-search ability of GWO algorithm is weak and is easy to fall into local optimum. Aijun Zhu [5] used Differential Evolution (DE) to update the position of best three wolfs in order to use DE's strong searching ability to improve GWO. Shahrzad Saremi [6] combined Evolutionary Population Dynamics (EPD) with GWO and used it to solve the function optimization problem. Tien-Szu Pan [7] proposed an improved GWO which used the split group strategy and inter-group communication strategy to improve GWO.

Cuckoo Search (CS) [8] is an optimization algorithm proposed by Xin-She Yang. The main idea of the CS algorithm is inspired from the aggressive reproduction strategy of cuckoo and levy-flight-style. The CS algorithm has the characteristics of less parameters, excellent search path and strong global-search ability. The aim of this paper is then to integrate the GWO algorithm with CS to update the best three grey wolves during the iteration, so as to improve the global-search ability of GWO and avoid it to fall into local optimum[1].

II. THE GWO ALGORITHM

A. Overview

Grey wolf is a kind of social life creature and has a strict hierarchical system shown in Fig. 1.

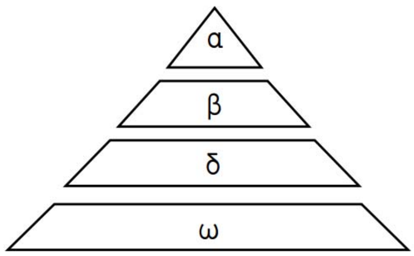

Figure 1. Hierarchical structure of grey wolves

In the nature, grey wolves usually live in groups. As shown in Fig. 1, four kinds of wolves are living in the group. Leader wolf in the group is called as alpha (α), which is located at the top of the pyramid. Alpha may not be the strongest wolf in the group, but it must be the best leadership manager. It is responsible for making important decisions for the group, such as predation behavior or food distribution. Located on the second floor of the pyramid is called as beta (β), which plays the role of alpha assistant helping alpha manage the group. It only needs to respect the alpha and it can command others. The third level wolf is delta (δ), which must obey the instructions of alpha and beta. When alpha and beta become old they will be downgraded to delta. The bottom of the pyramid called

[1] This work has been supported by the National Natural Science Foundation of China (No. 61602162, No. 61440024, No. 61502155), and the Doctoral Scientific Research Fund from Hubei University of Technology (No. BSQD12029).

as omega (ω). Omega has to submit to everyone else in the group.

B. Mathematical model

The level of grey is determined by the fitness function. According to the fitness value, the best fitness solution is the alpha_wolf, the beta_wolf and the delta_wolf. In this paper, these three solutions are set as key-group. The rest of the wolves is omega_wolf.

The process of predation is divided into three processes as follows.

1) Encircling prey

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{2}$$

In these two equations, $t+1$ represents the next iteration. $\vec{X}$ represents the position of one wolf. $\vec{X}_p$ represents the position of the prey, $\vec{A}$ and $\vec{D}$ are coefficient vectors. The calculation method is modeled by the following equations.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{4}$$

Where $\vec{r}_1$ and $\vec{r}_2$ are random number in [0,1]. Vector $\vec{a}$ is with the number of iterations between 2 and 0 to decrease linearly during the iteration process.

2) Hunting prey

When the wolf group has determined the location of prey, the alpha_wolf, the beta_wolf and the delta_wolf lead the wolf group to surround the prey. Assume that they know the position of the prey. Thus, store the best three solutions that are gained so far as key-group and update the position of each wolf in the group according to the key-group. These equations for position updating are shown as follows.

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{5}$$

$$\vec{X}_1 = \left|\vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha\right| \tag{6}$$

$$\vec{X}_2 = \left|\vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta\right| \tag{7}$$

$$\vec{X}_3 = \left|\vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta\right| \tag{8}$$

Where $\vec{X}_\alpha$, $\vec{X}_\beta$, $\vec{X}_\delta$ represent the best three solutions so far during the iteration process, which are the key-group. Other parameters are given by the following equations.

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right| \tag{9}$$

$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right| \tag{10}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right| \tag{11}$$

3) Attacking prey

In nature, grey wolves usually attack the prey when it stops moving. So, the behavior of grey wolves approaching the prey is modeled by the following equation.

$$A = 2 - 2\left(\frac{t}{Max}\right) \tag{12}$$

Where t means the number of times the current algorithm is running, which is an integer value between 0 and Max (max number of iteration).

III. Improving the Original GWO Algorithm

A. Integration with Cuckoo Search

The CS algorithm is inspired from the unique nesting way of cuckoo birds and levy-flight-style.

The reproduction strategies of some cuckoos are very special, by which they do not hatch their own offspring but to lay down their eggs in the host's nest while the host goes out, sometimes they also remove host bird's eggs away. Some host birds can find out the eggs belongs to outsider, and it may then move the outsider's eggs away or abandon the nest and find somewhere else to make a new one.

Levy-flight-style is a typical characteristic of flight behaviors for many animals. It refers to the individual generally in a smaller range of activities, but it may have a small probability of long-range jump. And it may also have a small probability of significant deviation from the mean value of the activities, as the power to CS algorithm jumping out of the local optimum.

CS algorithm needs to fit the following three idealized rules.

First, cuckoo birds choose the nest randomly and they only place one egg at once.

Second, only the best nests will remain to be the next generations.

Third, the number of bird's nests and the probability that the eggs are discovered are fixed. If outsider's egg is discovered by the host bird, host bird will abandon the nest and built a new one.

By complying with these three rules above, the nests are updated by following equations during the iteration.

$$\mathrm{x}_i(t+1) = x_i(t) + \alpha \oplus Levy(\lambda),\ \ i = 1,2,...,n \quad (13)$$

Where the product $\oplus$ representative entry-wise multiplication. $x_i(t+1)$ means the new solutions for cuckoo i . $x_i(t)$ represent the current solutions. Since $\alpha > 0$ control the step size, in most situation, it is set to 1. The levy-flight is provided by following equation.

$$Levy \sim u = t^{-\lambda},\ \ (1 < \lambda < 3) \qquad (14)$$

Therefore, the CS algorithm can search the solution space in an efficient way because its step changes with short distance detection and occasional long distance walking and the length of step is much longer in long run [8].

B. *Proposed CS-GWO algorithm for Improvement*

From Formula (5), it can be seen that the GWO algorithm updates the positions by trend search to the individuals with high fitness values, which is key-group. Thus, it will lead to a weak global-search ability, which may possibly be easy to fall into the local optimum, especially when it uses some high-dimension data sets.

The CS algorithm updates the positions of nest using random walk and levy-flights, while the search path is long or short with almost the same probabilities, and the direction is highly random. Thus it is easier to jump from the current area to other areas. According to this feature of CS, the CS algorithm is then integrated to improve the GWO algorithm.

The improved GWO algorithm integrated with cuckoo search is proposed as the CS-GWO algorithm, with its flow chart shown in Fig. 2.

IV. PRELIMINARY EXPERIMENTAL ANALYSIS

A. *Experimental Environment*

As for preliminary experimental analysis of the proposed CS-GWO algorithm, the NSL-KDD datasets are employed as experimental datasets. In this paper, 10% of the sampling data from the test file of the experimental datasets are selected as the test set and 10% of the sampling data from the training file of the experimental datasets are selected as as the training set.

Each connection record in the experimental datasets can be classified as one of the two categories which is normal and anomaly. Each record contains 41 features, as well as a category label, of which 38 are numeric features, 3 for the character type.

In the pre-processing phase, the three character-type features are mapped to numerical features. Since the "Service" feature in the data set has tens of different values, this feature causes a large false positives based on prior knowledge, hence this feature is not used in the preliminary experimental analysis, in order to reduce the interference of the artificial factors on the experimental results. Finally, all the other 40 features are all retained for preliminary experimental analysis of the proposed CS-GWO algorithm.

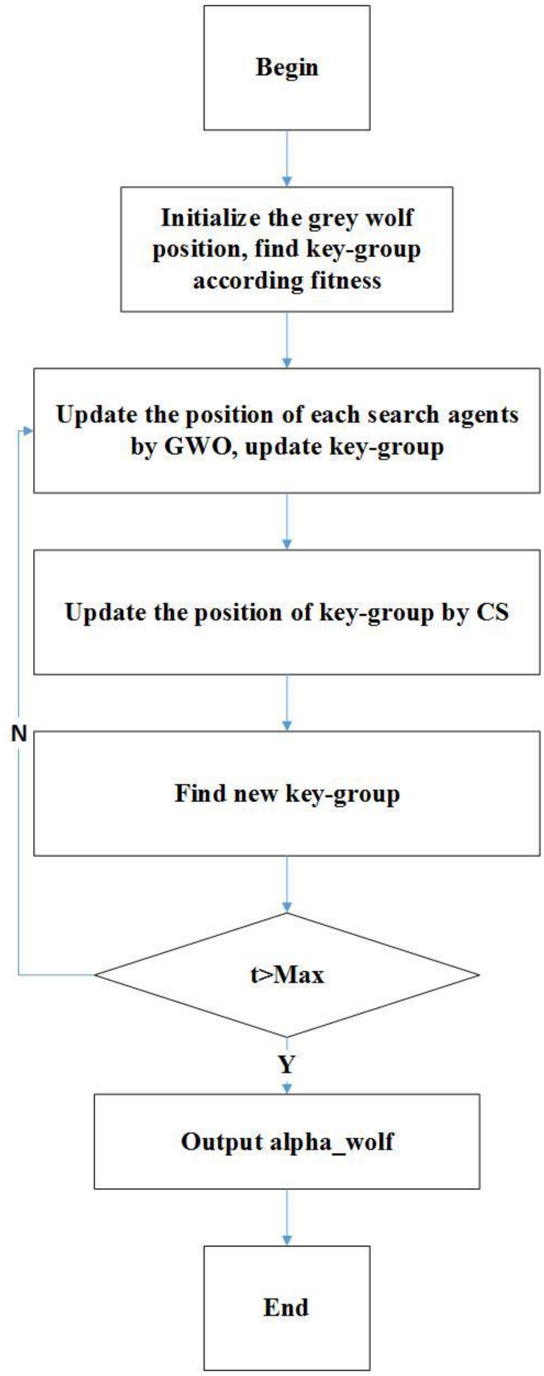

Figure 2. Flow chart of proposed CS-GWO algorithm

B. *Pseudo Code*

As for the purpose of preliminary experimental analysis, the pseudo code of the proposed CS-GWO algorithm is provided in Fig. 3.

```
Initialize the grey wolf position.
Initialize a, A ,C and Pa
Compute the fitness of each search agents in the pack
Set the Xα,Xβ,Xδ  according to the fitness
t=1
While(t<Max)
    for each wolf
        Update the position by Equation(5)
    end for
    Update a, A and C
    Compute the fitness of each search agents in the pack
    Update the Xα,Xβ,Xδ
    for Xα,Xβ,Xδ
        Update the position by Equation(13)
        If random number > Pa
            Random change wolf's position
        Compute the fitness and update it according to fitness
    t=t+1
end while
Output Xα
```

Figure 3. Pseudo code of proposed CS-GWO algorithm

C. *Experimental Result*

By choose the key feature from datasets using GWO, CS, CS-GWO algorithm respectively, we compared the global-search ability of these three algorithms according to the experimental results.

Table I shows the CS-GWO algorithm, GWO algorithm and the CS algorithm the best feature selection results after running 30 times.

TABLE I. COMPARATIVE RESULTS

Algorithm Name	Original Features Number	Subsets Features Number	Accuracy	Best Fitness
CS-GWO	40	6	83.54%	78.811%
GWO	40	11	83.54%	75.261%
CS	40	11	83.54%	75.261%
unselected	40	40	77.15%	55.107%

As shown in Table I, the classification accuracy of CS-GWO algorithm, GWO algorithm and CS algorithm are all improved compared with the no feature selection, and these three algorithms have the same classification accuracy. Note that, as for the proposed CS-GWO algorithm, number of selected subsets is less, and the best fitness is higher, when being compared with GWO algorithm and CS algorithm. The experimental results indicate that, the propsoed CS-GWO algorithm has the strongest global-search ability for these four cases in the NSL-KDD datasets.

In summary, compared with the original GWO algorithm, the proposed CS-GWO algorithm has a stronger global-search ability and might avoid to fall into the local optimum in high-dimension datasets. Furthermore, compared with both the original GWO algorithm and the original CS algorithm, the proposed CS-GWO algorithm might find the best fitness, and during the iteration, the proposed CS-GWO algorithm might jump out of the local optimum.

V. CONCLUSIONS

The main contribution of this paper is to propose an improved GWO algorithm integrated with the CS algorithm, named as the CS-GWO algorithm. The proposed algorithm utilizes the advantages of the CS algorithm for the GWO algorithm to update its best three grey wolves, extends the search area of GWO, and improves the defects of GWO.

Finally, the experimental results indicate that, the proposed CS-GWO algorithm has the strongest global-search ability for these four cases in the NSL-KDD datasets, and might avoid to fall into the local optimum and jump out of the local optimum in high-dimension datasets, compared with both the original GWO algorithm and the original CS algorithm.

In the future, the proposed CS-GWO algorithm will be applied to the domain of network security as is indicated in the preliminary experimental analysis on one hand, and it will be verified by more experimental datasets especially using those high-dimension datasets on the other hand.

REFERENCES

[1] D. Kreutz, F. M. V. Ramos and P. Verissimo, "Towards secure and dependable software-defined networks," *Proceedings of 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, 2013

[2] M. Vizváry and J. Vykopal, "Future of DDoS attacks mitigation in software defined networks," *Lecture Notes in Computer Science*, vol. 8508, pp. 123-127, 2014.

[3] Radware, *NetFlow and SDN based DDoS Attack Defense*, 2017, [Online]. Available: www.radware.com/Products/DefenseFlow/.

[4] H. B. Bae, M. W. Park, S. H. Kim and T. M. Chung, "Zombie PC detection and treatment model on software-defined network," *Lecture Notes in Electrical Engineering*, vol. 330, pp. 837-843, 2015.

[5] B. Schneier, "Attack trees: Modeling security threats," *Dr. Dobb's Journal*, vol. 12, no. 24, pp. 21-29, 1999.

[6] B. Schneier, *Secrets and Lies*, John Wiley and Sons, New York, pp. 318-333, 2000.

[7] C. Y. Yang and W. Cai, *Extenics*, Science Press, Beijing, Simplified Chinese version, 2014.

[8] W. Cai, C. Y. Yang and B. He, *Preliminary Extension Logic*, Science Press, Beijing, Simplified Chinese version, 2003.