Whale Optimization Algorithm with Chaos Strategy and Weight Factor

Yintong Li, Tong Han, Bangjie Han, Hui Zhao, Zhenglei Wei

1 Air Force Engineering University
   Xi’an, Shaanxi province, China
2 Air Force Engineering University
   Xi’an, Shaanxi province, China
3 Air Force representative room in Shijiazhuang Prefecture
   Shijiazhuang, Hebei province, China
4 Air Force Engineering University
   Xi’an, Shaanxi province, China
5 Air Force Engineering University
   Xi’an, Shaanxi province, China

ABSTRACT: Whale optimization algorithm (WOA) is a novel optimization algorithm inspired by humpback whale hunting behavior. Due to the defect of unbalanced exploration and exploitation by using control parameter with linear change, WOA has slow convergence and is easy to fall into local optimum. Thus, a novel whale optimization algorithm with chaos strategy and weight factor (WOACW) is proposed to improve the convergence speed and accuracy. In this work, the chaos strategy is executed to initialize the population to enhance the diversity of the initial population. The weight factor is introduced to adjust the influence degree of the current optimal solution on the generation of new individuals in order to improve the convergence speed and accuracy. At the same time, the convergence factor is adjusted by cosine function to better balance the relationship between exploration and exploitation. In addition, using greedy strategy to fully retain the dominant individuals from the parents and the generated candidates to generate offspring, improves the convergence speed of the algorithm. To verify the performance of our approach, WOACW is benchmarked on 13 classical benchmark functions, and the statistical results are compared with the original WOA and three other WOA variants, IWOA, WOAWC, and two state-of-the-art algorithms, SSA, GWO. The experimental results and Wilcoxon signed ranks test show that WOACW has a higher convergence speed and precision than compared algorithms, which verifies the effectiveness of WOACW in this work.

1. INTRODUCTION
Optimization refers to the process of obtaining the global optimal solution for a particular problem under a given condition. Most of the practical problems in scientific fields such as engineering design, economic planning and so on are complex optimization problems, and these high-dimensional complex problems can’t be solved well in a certain time range by using traditional methods. Thus,
researchers have put forward lots of efficient meta-heuristic algorithms inspired by natural phenomenon and animal behavior characteristics to solve complex practical optimization problems, but most meta-heuristic algorithms generally have defects that searching is easy to fall into the local optimum. How to balance the exploration and exploitation in the process of optimization and how to avoid the algorithm falling into local optimum or premature convergence are hot in algorithm research.

WOA is a novel meta-heuristic algorithm proposed in 2016 by Australian scholar Seyedali Mirjalili imitating the hunting behavior of humpback whales [1]. It has the advantages of simple structure, few control parameters, and considerable avoiding local optimum. It has been proved that WOA is superior to particle swarm algorithm (PSO)[2], Gravity search algorithm (GSA)[3], differential algorithm (DE)[4], Rapid evolutionary Planning (FTP)[5] and adaptive covariance matrix Evolutionary Strategy (CMA-ES)[6] in terms of convergence accuracy and speed [1]. However, it has the defects that exploration is difficult to coordinate with exploitation which makes WOA easily fall into local optimum.

Nowadays, the relevant theoretical research on WOA has been carried out. Kaur G introduces chaos theory into WOA (called CWOA) to tune the main parameter of WOA which helps in controlling exploration and exploitation to improve the performance of WOA [7]. Simulated Annealing (SA) algorithm [9] is embedded in WOA algorithm to enhance the exploitation property of the algorithm in literature [8]. Sinusoid, cosine, tangential, logarithmic and quadratic curves are introduced to control parameter $a$ to balance exploration and exploitation in literature [10]. In literature [11], inertia weight is introduced to tune the influence on the current best solution, which improves the exploration of WOA (called IWOA). Literature [12] (called WOAWC) introduces adaptive weight to improve the exploration, and introduces Cauchy variation strategy to improve the exploitation of the algorithm.

Although above variants of WOA has optimized its performance to a certain extent, there are still some shortcomings that searching is easy to fall into local optimum or slow convergence, which need to be continuously improved and innovated. In order to improve the convergence speed and precision of WOA in solving complex problems, this paper introduces chaos strategy, weight factor and greedy strategy respectively to propose a novel variant of WOA named WOACW. The experimental results based on classical benchmark function set show that the improved algorithm has excellent performance, which provides technical support for the next step to solve the practical problems.

The rest of this paper is structured as follows: Section 2 describes brief introduction of WOA. In Section 3 the improved version of WOA named WOACW has been proposed. Simulation experiment and interpretation of results on classical benchmark functions have been presented in Section 4. In Section 5 we conclude this paper and suggest directions for future work.

2. WHALE OPTIMIZATION ALGORITHM

WOA is a novel meta-heuristic algorithm proposed in 2016 by an Australian scholar Seyedali Mirjalili inspired by the hunting behavior of humpback whales [1]. Humpback whale is one of the biggest whales, mainly preying on krill and small fish herds near the sea surface. After discovering the prey, Humpback whales dive to the bottom of the prey about 12 meters deep, and then create distinctive bubbles along a circle or ‘9’-shaped path as shown in Fig. 1. At the same time, humpback whales upstream to the sea surface, through the bubbles to encircle the prey in a smaller range to devour. A detailed description of the predatory behavior of humpback whales can be found in reference [13]. Based on the author’s analysis and modeling of bubble-net feeding behavior, a new population-based intelligent algorithm, whale optimization algorithm, is proposed to solve the practical problems in engineering. The mathematical model based on the predation behavior of humpback whales is as follows.
Fig. 1 Bubble-net feeding behavior of humpback whales

2.1 Search for prey
The variation of the $A$ is utilized to search for prey (exploration) or Bubble-net attacking method (exploitation). In WOA, it uses $|A|$ with the random values greater than 1 to force search agent to move far away from the reference individual. In order to improve the exploration ability of the global search, the reference individual is randomly chosen. When $|A| > 1$, the new individual will be far away from the reference individual as shown in Fig. 2, the larger the $|A|$ is, the greater step size of the individual update position is. The mathematical model is as follows:

\begin{align}
C &= 2 \cdot r \\
a &= 2 - 2 \cdot t / \text{Maxiter} \\
A &= 2 \cdot a \cdot r - a \\
D &= |C \cdot X_{\text{rand}}(t) - X(t)| \\
X(t+1) &= X_{\text{rand}}(t) - A \cdot D
\end{align}

where $r$ is a random value in [0, 1], $t$ indicates the current iteration, Maxiter is the maximum number of iterations, $A$ and $C$ are coefficient variations, $X$ is the position vector, $X_{\text{rand}}$ is a random position vector chosen from the current population, $a$ is the convergence factor, as the number of iteration increased from 2 linear to 0, $|$ $|$ is the absolute value, and $\cdot$ is an element-by-element multiplication.

2.2 Encircling prey
Humpback whales will encircle their prey once confirming the location of prey. Since the ‘prey’ position is unknown in practical problem during the optimization process, WOA assumes that the current optimal solution is the "prey" position or close to the optimal solution. And then other search individuals update their location information toward the current optimal solution. The mathematical model of process is established by Eq. (6) and (7). Theoretically, by adjusting the value of $A$ and $C$, the search agent can reach any position around the reference individual. This behavior is represented by the following equations:

\begin{align}
D &= |C \cdot X^*(t) - X(t)| \\
X(t+1) &= X^*(t) - A \cdot D
\end{align}

where $X^*$ is the vector of the best solution obtained so far.

2.3 Bubble-net attacking method
In WOA, there are two methods designed to mimics bubble-net attack mechanism of the humpback whales. The mathematical model of two methods are as follows.

2.3.1 Shrinking encircling mechanism
Shrinking encircling mechanism is achieved by decreasing the value of $a$ in the Eq. (3). From Eq. (3), it is easy to know that $A$ is a random variable in the interval $[-a, a]$. The whales update the range of
their position in the 2D space according to the Eq. (7) as shown in Fig. 2. When $|A|<1$, the individual approaches to the current optimal solution.

2.3.2 Spiral updating mechanism

The spiral updating mechanism first calculates the distance between the whale position and prey position. And then a spiral equation is employed to mimic the helix-shaped movement of humpback whales as follows:

$$D^* = |X^*(t) - X(t)|$$ (8)

$$X(t+1) = D^* e^{|l|} \cdot \cos(2\pi l) + X^*(t)$$ (9)

where $b$ is a constant for defining the shape of the logarithmic spiral, $l$ is a random value in $[-1,1]$. Note that humpback whales hunt for prey with Bubble-net attacking method by a shrinking circle and along a spiral-shaped path simultaneously. So WOA model assumes that there is a probability of 50% to choose between the shrinking encircling mechanism and spiral updating mechanism to update the position of whales during optimization. The specific mathematical model is as follows.

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, & p < 0.5 \\ D^* e^{|l|} \cdot \cos(2\pi l) + X^*(t), & p \geq 0.5 \end{cases}$$ (10)

where $p$ is a random value in $[0,1]$.

3. PROPOSED WOACW ALGORITHM

Although it has a good convergence speed, WOA is still unable to achieve this performance better when looking for the global optimal solution. Thus, in order to improve the convergence speed and accuracy, WOACW is proposed by introducing chaos strategy, weight factor and greedy strategy to WOA in this paper. The detail of WOACW is as follows.

3.1 Chaos Strategy Initializes Population

The convergence efficiency and quality of the global optimal solution are greatly influenced by the diversity of the initial population, and the low convergence efficiency of the WOA is related to a certain extent that its randomly initialized population can’t guarantee the uniform distribution in the search space. Chaos refers to the nature of complex systems with unpredictable behavior, and mapping refers to the use of functions to map or correlate chaotic behavior in an algorithm with a parameter. By using the randomness and traversal characteristics of chaotic systems, chaos strategy initializing population can improve the distribution quality of the initial population in the search space to improve the diversity of the population which improves the convergence efficiency of the algorithm.

Based on the topological conjugate relationship between logistic chaotic mapping and Tent chaotic mapping, a definition form of segmented logistic chaotic mapping is given by combining segmented tent chaotic mapping in Literature [14]. It is proved that the segmented logistic chaotic mapping
defined has good efficiency and security. In this paper, the segmented logistic chaotic mapping strategy proposed in Literature [14] is adopted to initialize Population. The model is as follows.

\[
a_{i+1} = \begin{cases} 4\mu a_i (0.5 - a_i), & 0 \leq a_i < 0.5 \\ 1 - 4\mu a_i (0.5 - a_i)(1 - a_i), & 0.5 \leq a_i \leq 1 \end{cases} \tag{11}
\]

where \(3.569946 \leq \mu \leq 4\), \(\mu = 4\), and \(a_r = \text{rand} \in (0, 1)\), chaotic sequences are generated and then mapped to search space to generate initial populations.

3.2 Weight Factor

In Particle swarm algorithm (PSO) [2], the inertial weight \(w\) is a controlling parameter in the speed update formula which keeps the particles moving inertia. If \(w=0\), the particle velocity is not memory and the particle swarm will lose itself which lead to produce a divergence effect. Inspired by this, the weight factor \(w\) is introduced in WOA to adjust the influence degree of the current optimal solution on the generating offspring in the process of optimization. The influence degree of the current optimal solution on the generation of descendant individuals is unchanged during the whole optimization process in WOA from Eq. (10). Thus, the exploration ability in the later stage of optimization and the exploitation ability in the early stage of optimization should be strengthen, that is to say that \(w\) value should be monotone increasing function. Through the comparison and analysis of simulation experiments, the use of sinusoidal function can achieve better results. The improved model of Eq. (10) is as follows.

\[
w = \sin\left(\frac{\pi - t}{2}\cdot\text{Maxiter} - \pi / 2\right) + 1 \tag{12}
\]

\[
X(t+1) = \begin{cases} w \cdot X'(t) - A \cdot D, & p < 0.5 \\ D \cdot c^{x} \cdot \cos(2\pi t) + w \cdot X'(t), & p \geq 0.5 \end{cases} \tag{13}
\]

From literature [1] and Eq.(3), the control parameter \(a\) is a major factor to affect the algorithm’s convergence precision and speed. Exploration and exploitation of WOA are guaranteed by the values of convergence factor \(a\). The value of convergence factor \(a\) is linearly decreased from 2 to 0 over the course of iterations. The linearly varying convergence factor strategy has shown faster convergence and high accuracy in the early stage for most cases by empirical investigations with some well-known problems [10]. However, the actual optimization process of WOA is nonlinear and complex, so it can’t be truly reflected by the linearly decreasing convergence factor \(a\). Literature [10] proposes corresponding improved WOA algorithm with different nonlinear adjustment strategy by adopting sinusoid, cosine, tangential, logarithmic and quadratic curves to adjust convergence factor \(a\). The results of literature [10] show that the cosine function can obtain better results. In this paper, the cosine function is adopted and the parameter \(u=3\) is determined by experiments. The equation is as follows:

\[
a = \cos(3\cdot \pi \cdot t / \text{Maxiter} - \pi / 2) + 1 \tag{14}
\]

In order to fully retain the dominant individuals, the fitness values of the parent and new individuals are sorted, and the dominant individuals are selected to generate offspring using greedy strategy to improve the convergence speed of the algorithm.

Based on the above considerations, the pseudo-code of WOACW is illustrated in Algorithm 1.

\begin{center}
Algorithm 1: Pseudo-code of WOACW
\end{center}

- Initialize the population \(X_i\) by chaos strategy
- Calculate the fitness of each search agent
- \(X^*\) = the best search agent
- \(\text{while}\ t < \text{Maxiter}\)
- \(\text{for}\ \text{each search agent}\)
- \(\quad \text{Update} \ w, \ a, \ A, \ C, \ l, \ and \ p\)
- \(\quad \text{if} \ (p < 0.5)\)
\textbf{if2} (|A| \geq 1)
Select a random search agent
Update the position of search agent by the Eq. (5)
\textbf{else if2} (|A|<1)
Update the position of search agent by the Eq. (13)
\textbf{end if2}
\textbf{else if1} (p \geq 0.5)
Update the position of search agent by the Eq. (13)
\textbf{end if1}
Check if goes beyond the search space and emend it
\textbf{end for}
Calculate the fitness of each search agent
Retain dominant individuals by greedy strategy
Update X* if there is a better solution
t=t+1
\textbf{end while}
Return X*, best fitness value

\section{NUMERICAL EXPERIMENTS}
In this section, in order to illustrate the effectiveness of the proposed WOACW, the performance of WOACW is evaluated based on a series of 13 classical benchmark functions selected from literature [1]. By comparing WOACW with its basic version WOA [1], two other WOA variants IWOA [11], WOAWC [12], and two state-of-the-art algorithms SSA [15], GWO [16], the performance of WOACW proposed in this paper is analyzed. All these numerical experiments have been performed on computer with Inter(R)Core(TM)i7-4770K CPU @3.50GHz 8GB RAM using MATLAB 2013a.

\subsection{Experiment Results and Analysis}
The series of 13 benchmark functions selected from classical benchmark functions consist of unconstrained optimization problems which can be divided into two main molds: unimodal functions (f1–f6) and multimodal functions (f7–f13). Unimodal functions are mainly used to evaluate the exploitation ability of algorithm. Meanwhile, multimodal functions are usually used to evaluate the exploration and avoidance local optimum ability of algorithm. Table 1 presents a brief description about the different types of benchmarks.

\begin{table}[h]
\centering
\caption{The description of Benchmark function}
\begin{tabular}{|c|c|c|c|c|}
\hline
Function & Name & Dimension & Range & \textit{f}_{\text{min}} \\
\hline
\text{f}_1(x)=\sum_{i=1}^{n} x_i^2 & Sphere & 30 & [-100,100] & 0 \\
\hline
\text{f}_2(x)=\sum_{i=1}^{n} |\prod_{i=1}^{n} x_i| & Schwefel2.22 & 30 & [-10,10] & 0 \\
\hline
\text{f}_3(x)=\left(\sum_{i=1}^{n} x_i \right)^2 & Schwefel 1.2 & 30 & [-100,100] & 0 \\
\hline
\text{f}_4(x)=\max \left\{ |x_i|, 1 \leq i \leq n \right\} & Schwefel 2.21 & 30 & [-100,100] & 0 \\
\hline
\text{f}_5(x)=\sum_{i=1}^{n} \left[100(x_i-x_i^2)^2+(x_i-1)^2\right] & Rosenbrock & 30 & [-30,30] & 0 \\
\hline
\text{f}_6(x)=\sum_{i=1}^{n} (x_i+0.5)^2 & Step & 30 & [-100,100] & 0 \\
\hline
\text{f}_7(x)=\sum_{i=1}^{n} x_i^4 + \text{random}(0,3) & Quartic & 30 & [-1.28,1.28] & 0 \\
\hline
\end{tabular}
\end{table}
In order to guarantee the fairness of the experiments, the population size of the 6 algorithms is set to 30 and the maximum number of iterations is 500 on the basis of the same conditions of the experimental platform. In addition, the parameters specific to each comparison algorithm are set in reference to the original literature. In order to avoid the occasionality of experimental data, each test is run independently 30 times, and Table 2 lists the mean and standard deviation (SD) of the statistical results obtained by algorithms through 30 independent runs for benchmark functions.

In order to observe the convergence accuracy of WOACW on a series of representative functions. As can be seen from fig.3, the convergence speed of each algorithm more intuitively, fig.3 especially in solving the problems of high real-time requirement. Thus, in order to observe the contrasting algorithms on all functions.

Table 2. Statistical results obtained by algorithms through 30 independent runs for benchmark functions.

| Function | Mean(SD) | Mean(SD) | Mean(SD) | Mean(SD) | Mean(SD) | Mean(SD) |
|----------|----------|----------|----------|----------|----------|----------|
| f1 = \(\sum_{i=1}^{n} x_i \sin(\sqrt{x_i})\) | Schwefel2.26 | 30 | [-500,500] | 418.992 | 90D |
| f2 = \(\sum_{i=1}^{n} x_i^2 - 10 \cos(2\pi x_i) + 10\) | Rastrigin | 30 | [-5,12.5,12] | 0 |
| f3 = \(-20 \exp \left( -0.2 \prod_{i=1}^{n} x_i^2 \right) - \exp \left( \prod_{i=1}^{n} \cos(2\pi x_i) \right) + 2\) | Ackley | 30 | [-32,32] | 0 |
| f4 = \(\frac{1}{4000} \sum_{i=1}^{n} x_i^2 \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1\) | Griewank | 30 | [-600,600] | 0 |
| f5 = \(\prod_{x_i} 10 \sin(\pi x_i) + \sum_{x_i} (x_i - 1)^2 \) | Penalized | 30 | [-50,50] | 0 |
| f6 = \(\prod_{x_i} 10 \sin(\pi x_i) + \sum_{x_i} (x_i - 1)^2 \) | Penalized 2 | 30 | [-50,50] | 0 |

As we all know, the convergence rate of search algorithm is more important than the final results especially in solving the problems of high real-time requirement. Thus, in order to observe the convergence speed of each algorithm more intuitively, fig.3–fig.9 list the convergence curves of a series of representative functions. As can be seen from fig.3–fig.8, although the convergence accuracy of WOACW on f1 is worse than that of SSA, the convergence speed of WOACW is faster than 6 contrasting algorithms on all functions.
The Wilcoxon signed ranks test is a simple, safe and robust, nonparametric test for pairwise statistical comparisons, and it is usually used to detect the significant differences between two samples. In order to compare the performance of algorithms as a whole on the 13 classic benchmark functions, the pair-wise comparison between the WOACW and the other competitors based on the Wilcoxon signed ranks test [18] with a confidence level of 0.95 is employed. The symbol R in Table 2 represents the results of Wilcoxon signed ranks test, where ‘+’ in the Table 2 means the compared algorithm performs better than WOACW, ‘−’ indicates it performs worse than WOACW and ‘=’ means the algorithm perform equally with WOACW. In the last row in Table 2 we list the number of ‘+/−’ counts compared with WOACW. It means that WOACW exhibits the statistically superior performance than the five compared algorithms in the pair-wise Wilcoxon signed ranks test.

5. CONCLUSION AND FUTURE SCOPE
In this work, we proposed an improved version of WOA called WOACW by incorporating chaos strategy and weight factor to improve convergence speed and accuracy of WOA based on WOA and the achievements of predecessors. A series of 13 classical benchmark functions consisting of unimodal and multimodal functions have taken to check the performance of the WOACW. The performance of WOACW is compared with basic WOA, two WOA variants and other meta-heuristic algorithms, which show that WOACW is competitive with the competitors in terms of convergence speed and
accuracy. From the analysis of the results in this paper it can verify the effectiveness of the improved strategy proposed in this paper.

In addition, the proposed WOACW can hardly converge to theoretical optimal solutions for some complex functions such as $f_5, f_6, f_{13}$. In the future work, how to improve the performance of WOACW for more complex problems will be our work. Meanwhile, we would try to employ WOACW for solving different optimization problems like constrained optimization problems, integer programming problems, such as unmanned combat aerial vehicles task assignment and trajectory planning.

ACKNOWLEDGMENTS
The study is supported by Aeronautical Science Foundation of China (No.20175196019) and Shaanxi nature science foundation (No.2017JM6078).

REFERENCES
[1] Mirjalili, S, & Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 95, 51-67. DOI: https://doi.org/10.1016/j.advengsoft.2016.01.008
[2] Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning (pp. 760-766). Springer, Boston, MA. DOI: https://doi.org/10.1007/978-0-387-30164-8_630
[3] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248. DOI: https://doi.org/10.1016/j.ins.2009.03.004
[4] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.
[5] Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. IEEE Transactions on Evolutionary computation, 3(2), 82-102. DOI: 10.1109/4235.771163
[6] Hansen, N, Müller, S. D., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary computation, 11(1), 1-18. DOI: https://doi.org/10.1162/10636560321828970
[7] Kaur G, Arora S. (2018). Chaotic Whale Optimization Algorithm. Journal of Computational Design & Engineering, 5(3). 275–284. DOI: https://doi.org/10.1016/j.jcde.2017.12.006
[8] Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. Neurocomputing, 260, 302-312. DOI: https://doi.org/10.1016/j.neucom.2017.04.053
[9] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680. DOI: 10.1126/science.220.4598.671
[10] Zhong, M H, & Long, W. (2017). Whale optimization algorithm with nonlinear control parameter. In MATEC Web of Conferences (Vol. 139, p. 00157). EDP Sciences. DOI: https://doi.org/10.1051/matecconf/201713900157
[11] Hu, H P, Bai, Y P, & Xu, T. (2016). A whale optimization algorithm with inertia weight. WSEAS Trans. Comput, 15, 319-326.
[12] Guo Z. Z, Wang P, & Yun-Feng, M. A. (2017). Whale optimization algorithm based on adaptive weight and cauchy mutation. Microelectronics and Computer, 34(9), 20-25. DOI:10.19304/j.cnki.issn1000-7180.2017.09.005
[13] Goldbogen, J. A., Friedlaender, A. S., Calambokidis, J., Mckenna, M. F., Simon, M., & Nowacek, D. P. (2013). Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. BioScience, 63(2), 90-100. DOI: https://doi.org/10.1525/bio.2013.63.2.5
[14] Fan, J. L., & Zhang, X. F. (2009). Piecewise logistic chaotic map and its performance analysis. Acta Electronica Sinica, 37(4), 720-725. DOI: 10.3321/j.issn:0372-2112.2009.04.009
[15] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software, 114*, 163-191. DOI: https://doi.org/10.1016/j.advengsoft.2017.07.002

[16] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software, 69*, 46-61. DOI: https://doi.org/10.1016/j.advengsoft.2013.12.007

[17] Li, M. D., Zhao, H., Weng, X. W., & Han, T. (2016). A novel nature-inspired algorithm for optimization: Virus colony search. *Advances in Engineering Software, 92*, 65-88. DOI: https://doi.org/10.1016/j.advengsoft.2015.11.004

[18] Derrac, J., Garcia, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3-18. DOI: https://doi.org/10.1016/j.swevo.2011.02.002