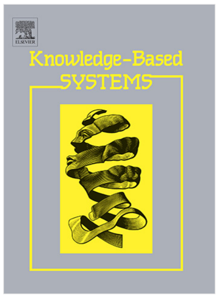
Accepted Manuscript



The defect of the Grey Wolf optimization algorithm and its verification method

Peifeng Niu, Songpeng Niu, Nan liu, Lingfang Chang

PII: S0950-7051(19)30018-8
DOI: https://doi.org/10.1016/j.knosys.2019.01.018
Reference: KNOSYS 4647

To appear in: *Knowledge-Based Systems*

Received date : 8 September 2018
Revised date : 8 December 2018
Accepted date : 11 January 2019

Please cite this article as: P. Niu, S. Niu, N. liu et al., The defect of the Grey Wolf optimization algorithm and its verification method, *Knowledge-Based Systems* (2019), https://doi.org/10.1016/j.knosys.2019.01.018

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

*Revised Manuscript (Clean Version)
Click here to view linked References

ACCEPTED MANUSCRIPT

The defect of the grey wolf optimization algorithm and its verification method

Peifeng Niu[a], Songpeng Niu[a], Nan liu[a], Lingfang Chang[a]

[a]Institute of electric engineering, Yanshan university, 066004 Qinhuangdao, China

Abstract: Grey wolf optimization algorithm (GWO) is a new meta-heuristic optimization technology. Its principle is to imitate the behavior of grey wolves in nature to hunt in a cooperative way. GWO is different from others in terms of model structure. It is a large-scale search method centered on three optimal samples, and which is also the research object of many scholars. In the course of its research, this paper find that GWO is flawed. It has good performance for the optimization problem whose optimal solution is 0, however, for other problems, its advantage is not as obvious as before or even worse. Then it is further found that when GWO solves the same optimization function, the farther the function's optimal solution is from 0, the worse its performance, and this flaw also appears in other optimization algorithms. Through the study of this defect, the analysis is carried out, and the reason is determined. Finally, although there is no way to make GWO normal, this paper provides a verification method to avoid the same problem, and hopes to help the development of the optimization algorithm.

Keywords: Meta-heuristic; Gray Wolf optimization algorithm; Optimization algorithm; Defect; New verification method

1. Introduction

Meta-heuristic algorithm is a kind of optimizing techniques. It is inspired by the principles or structures of nature and is used to solve optimization problems [1-3]. In the past 20 years, the optimization algorithm has received extensive attention and has been well developed. After all, it is a new type of optimization method, and can describe all aspects of nature to create a different optimization model. There are many aspects in this area, such as particle swarm optimization (PSO) [4], genetic algorithm (GA) [5], Teaching and Learning Based Optimization (TLBO) [6]. Its characteristics are that simplicity, flexibility, derivation-free mechanism, and local optima avoidance [7,8]. It is described as one of the most promising areas for solving real-world optimization problems [9,10].

The optimization algorithm is also known as swarm intelligence algorithm. Although it has a good development prospect, but its research is still in the initial stage, and there are many problems need to be solved [10,11]. For example, How to effectively avoid local optimum? How to perfectly combine the advantages of different optimization algorithms? How to effectively set the parameters of an algorithm? What are the effective iteration stop conditions? and so on. The most important problems is that it lacks a unified and complete theoretical system. That is to say at present, swarm intelligent algorithm is developed in exploration, and its model tends to exploit innovation. This is an expedient measure without theoretical guidance at the present stage. In practical applications, this method will inevitably cause some problems [12,13]. In this case, how to effectively regulate its development direction? It is a problem that must be taken seriously. This method can be considered to help swarm intelligence algorithm: discover an error in the optimization algorithm, analyze it in detail, find out the cause, and draw a conclusion or propose a method to avoid, which is used to standardize the development of the optimization algorithm.

GWO is a new meta-heuristic optimization technology, which was proposed by Australian scholar Mirjalili in 2014. It simulates the hierarchical mechanism and predation behavior of the grey wolf pack, that under the leadership of the head grey wolf, the wolves capture the prey through a series of processes, such as surrounding, hunting and attacking. Judging from the actual results, this is a large-scale search algorithm centered on 3 best grey wolves. There is no elimination mechanism in GWO, that is to say, if a gray wolf is going to a worse place than it is now, it still must arrive, which makes this optimization algorithm more fluid and has a stronger global search capability. On the whole, it is easy to operate, has few parameters, and is easy to implement. It is the object of

ACCEPTED MANUSCRIPT

many scholars' research, and has achieved many results [14-18].

However, a problem was discovered in the course of its research, which shown that GWO was defective. This paper through repeated experiments, step-by-step analysis, and finally draw a conclusion: only for an optimization problem whose optimal solution is 0, GWO is obviously better than others; for other problems, the performance of it is general or even poor. Then it is further discovered that for the same function, as its optimal solution move away from 0, the performance of GWO is gradually degraded. This is very bad for an optimization algorithm, because whether the optimal solution of an optimization problem is 0 or not, it could be known by experiment once. If it is, there is no need to solve it again; if not, the performance of GWO is uncertain, which is not as good as other optimization algorithms. Therefore, GWO is a flawed optimization algorithm. After in-depth research, the reason was found. It was also found that other algorithms may have this defect, and the verification method of this paper can be used to test. Finally, the paper hopes that it can help scholars no longer make the same mistake and looks forward to the further development of optimization algorithms.

2. Grey Wolf Optimizer

GWO is a bionic optimization algorithm. It mimics the behavior of gray wolves to capture prey with a clear division of labor and mutual cooperation. At the top of the food chain, gray wolves mostly prefer to live in a pack [7]. Usually, there are 5-12 wolves in each group. They have a strict hierarchical management system that constitute a hierarchical pyramid as shown in Fig.1. This hierarchy allows the grey wolf pack to efficiently kill the prey.

α layer is the head wolf, which is the strongest and most capable individual. It is also the only leader in a wolf pack, who directs the team's predation actions, food distribution, and other decision-making tasks. β and δ layer are two wolves that are second only to α, their responsibility is mainly to assist α in the behavior of group organizations. ω is at the bottom of the pyramid, which occupies the majority of the total, and is mainly responsible for balancing the internal relationship of the population and looking after the young.

2.1. *GWO algorithm description*

The predation process of gray wolf pack could be divided into 3 stages: encircling, hunting and attacking.

2.1.1. *Encircling*

After determining the location of its prey, the gray wolves began to surround it.

$$D_p = | C \cdot X_p(t) - X(t) | \tag{1}$$

$$X(t+1) = X_p(t) - A \cdot D_p \tag{2}$$

where t is the number of iteration, $X(t)$ is one grey wolf, $X(t+1)$ is the next position it arrives, $X_p(t)$ specifically refers to one of α, β, δ. Where A and C are coefficient vectors, expressed as follows.

$$A = 2ar_1 - a \tag{3}$$

$$C = 2r_2 \tag{4}$$

Where r_1, r_2 are random vectors in [0,1], a is a decreasing value in [0, 2], typically a = 2-2t/I (I is the maximum number of iterations).

2.1.2. *Hunt*

After encircling its prey, under the guidance of α, β, δ, gray wolves hunted the prey. Its update principle in this process was shown in Fig. 2, update equation as follow.

ACCEPTED MANUSCRIPT

$$\begin{cases} D_\alpha = | C_1 X_\alpha - X(t) | \\ D_\beta = | C_2 X_\beta - X(t) | \\ D_\delta = | C_3 X_\delta - X(t) | \end{cases} \tag{5}$$

$$\begin{cases} X_1 = X_\alpha(t) - A_1 D_\alpha \\ X_2 = X_\beta(t) - A_2 D_\beta \\ X_3 = X_\delta(t) - A_3 D_\delta \end{cases} \tag{6}$$

$$X_p(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{7}$$

2.1.3. *Attack*

Gray wolves had surrounded the prey and began to prepare for capture (convergence and get results). because of $A \in [-2a, 2a]$, This process was mainly achieved by the decrement of a in Eq. (3). When $|A| \geq 1$, the gray wolves would stay away from the prey to achieve global search; when $|A|<1$, the gray wolf pack would approach the prey and finally complete it.

3. The performance of GWO

In order to analyze the defects of GWO, it is necessary to first understand its performance. PSO, DE, CS, CSO and GSA are selected to compare with GWO. Their sample size(N) and maximum number of iterations(I) are set to N=100, I=1000. The parameters of GWO is in Section 2, and other algorithms are shown in Table 1.

Benchmark test function is the most basic standard for testing the performance of an algorithm. The performance of an algorithm detected by it will also revealed in other applications. The optimization problems chosen are shown in Table 2. They are mainly derived from the literature on GWO research, which contain single peaks, multiple peaks, mixed or complex types, etc [7,19,20]. The optimization is programmed in MATLAB 7.1 using a window 7 with Core(TM)i5-3210M 2.50 GHz CPU.

For every standard test function, each optimization algorithm run continuously 50 times. The results obtained are shown in Table 3, including the average and standard deviation of the results of each algorithm. For convenience, when the accuracy of a result is exceeds 10^{-100}, it is recorded as the optimal.

As can be seen from Table 3, the performance of GWO is very excellent. At f_1-f_6, it has satisfactory results, is the best compared to other algorithms: its average value is the closest to the optimal solution, and the standard deviation is also the smallest. However, for other optimization algorithms, only in f_6, CSO can get a better result, the rest are relatively poor regardless of average or variance. In f_7-f_9, although the performance of GWO is not as great as before, it is also quite good. At f_7, it is better than CSO, DE; at f_8, it is second only to DE and GSA; at f_9, it is better than CS, PSO and GSA. Among these functions, GWO does not show strong performance, its performance is more general. But on the whole, GWO has good rapidity and robustness. But is this really the case?

4. The defect of GWO

Through the above experiments, it can be know that GWO has excellent performance. But a problem has been found that, in f_1-f_6, the accuracy of its result is at least 10^{-55}, however in f_7-f_9, it is at most 10^{-8}. Although there is no comparability between the results of the different functions, it is still lead us to think about the reason for the larger differences between these two parts. Comparing these two parts, it can be found that their optimal solution(X^*) is different. To verify whether this is the cause, the functions in Table 2 are modified. The specific rule is that if X^* of an original function is 0, then it is changed to X^*=1; if $X^* \neq 0$, it is changed to X^*=0. The

ACCEPTED MANUSCRIPT

modified test functions are shown in Table 4, where k is a parameter that can be used to change X^*. Then the paper repeat the experiment, analyze the results, and observe the changes of these two results.

In the same experimental environment, the results obtained are shown in Table 5. It can be see that GWO is only better than CSO in f_1, f_3, and is better than CSO and DE in f_2, f_6, even, it is the worst in f_5. However, In f_7-f_9, it is obvious that GWO shows excellent performance. The performance of GWO also can be divided into 2 parts: f_1-f_6 and f_7-f_9, the former performs is more common, while the latter is very good.

Based on the above two experiments, it can be found that their conclusions are completely opposite. it can be affirmed that GWO can get better results on the optimization function whose $X^*=0$, and its performance has greater uncertainty for other problems. So for the same test function, when its X^* changes, what happens to the performance of GWO? Set $k\in[-5, 5]$ in Table 4, other conditions are unchanged, and the optimal value curve of GWO is shown in Fig. 3. Fig. 4 is a variation curve of other algorithms in f_1.

From Fig. 3, it can be seen that as $|k|$ increases, these curves are increasing. In f_7-f_8, they grow as a concave function, and the rest is a convex function. But what the magnitude of growth is, or whatever the reason made it grow, it can be sure that the performance of GWO is decreasing, when X^* is away from 0.

From Fig. 4, it can be find that DE, PSO, CS, GSA, their curves are irregular, and their variation can be accepted, while CSO has similar changes to GWO. That is to say, the defect mentioned in paper do not only appear on GWO. This is very bad, the innovations they propose do not really make the development of optimization algorithms, then it should be avoided, and the verification method in this paper can be used.

4.1. *Instance verification*

When different algorithms get the same result for the same function, their cost(N, I) can also be used to measure their performance. In a similar way, in order to verify whether GWO has the above problems in the actual work, there are 4 engineering problems selected from literature [21]. They are used for experiments, and get N and I when max(Std)<0.01. Then these problems are modified so that their solutions are all close to 0, the experiment is repeated. All results are shown in Table 6. It can be seen that when $X^* \neq 0$, GWO costs at least 2 times more than the other. That is to say, in this case, the performance of the GWO is degraded, which is the same as the previous conclusion.

5. Cause Analysis

Because the optimization algorithm is a method to control random changes, it is very complicated, cumbersome, or even impossible to fully explain its change pattern. However, because GWO is simple to operate, Unlike other algorithms that have this type of defect, this makes it possible to analyze the reason.

The code of GWO is very concise, so Eq. (2) is locked. For this formula, the interpretation of GWO is shown in Fig. 5. But, when analyze this formula carefully, it could find that Eq. (2) is not as described. Suppose $X_p=0$, then $X(t+1)=A\cdot|X(t)|$, when $|A|<1$, this is a variable that converges to 0. That is, in the later stage of GWO operation, $X(t)$ will tend to 0 regardless of its previous value. Of course, this is the same when $Xp=\varepsilon$ (a value very close to 0). If $Xp>\varepsilon$, the position of $X(t+1)$ cannot be determined, at this time, GWO is always in the stage of exploration, its result is random and not robust. Xp is the current optimal solution, representing the current best result. It can lead to sample changes, but it should not cause such an effect. In summary, when the optimal solution of a problem is 0, GWO can quickly find its optimal, but it can not solve other problems well. Unfortunately, Many methods have been tried, and they have not made GWO normal.

6. Conclusion

This paper takes GWO as the research object, and then gradually discusses it through experiments, and

ACCEPTED MANUSCRIPT

finally determines that its characteristic is that when solving an optimization problem whose $X^*=0$, its performance is very good, but as X^* farther away from 0, its performance is worse. This is very bad for an algorithm. Whether it is an optimization algorithm or a traditional method, the reason for using them is because the optimal solution of a problem is unknown or uncertain, and which algorithm is chosen based on better applicability and stability among them. The characteristics of GWO make it limited, therefore, in the application, it can not be favored, even the research based on GWO has yet to be verified.

The problem with GWO is not just a special case. In Section 4, it can be seen that CSO has the same defects as GWO. The defect of GWO is because of Eq.(2), so every algorithm which has it is need to verified, such as the Whale Optimization Algorithm (WOA) [22], A Sine Cosine Algorithm (SCA) [23]. Whether other algorithms have this problem is not clear, but it is like a problem of the entire optimization algorithm.

The reason for this may be due to the standard test function. Although they can effectively test the performance of an algorithm, but too many of them are use 0 as the optimal solution. This will make scholars more refer to their results, when building the model, so that the optimization algorithm has the characteristics like GWO.

At present, there is no complete theory to guide the development of optimization algorithms and in this process, it is inevitable that problems may arise. GWO is an example presented in paper, and its problem is discovered. Because the optimization algorithm is a method of controlling randomness, it is difficult to analyze the cause of its problem, so only the test method is proposed in paper. It can be seen from the above experiments that it works well and can be used to test other algorithms. Finally, the optimization algorithm is expected to be further developed.

References

[1] M.G. Gong, L.C. Jiao, D.D. Yang, et al., Research on evolutionary multi-objective optimization algorithms, Journal of Software. 20 (2009) 271-289.

[2] I. Rechenberg, Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, 15 (1973).

[3] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press. (2008).

[4] J. Kennedy, R. Eberhart, Particle swarm optimization, International Conference on Neural Networks, IEEE, 2002, pp. 1942-1948 vol.4.

[5] D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machinelearning. (1989).

[6] A. Mortazavi, V. Toğan, A. Nuhoğlu, Interactive search algorithm: A new hybrid metaheuristic optimization algorithm, Engineering Applications of Artificial Intelligence. 71 (2018) 275-292.

[7] S. Mirjalili, S,M. Mirjalili, A. Lewis, Grey Wolf Optimizer. Advances in Engineering Software. 69 (2014) 46-61.

[8] R. Salgotra, U. Singh, Application of Mutation Operators to Flower Pollination Algorithm, Expert Systems with Applications. 79 (2017).

[9] E. Engeler, Introduction to the Theory of Computation, Introduction to the theory of computation, PWS Pub, Co, 1997, pp. 2[illegible]-29.

[10] W.X. Xing, J.X. Xie, Modern Optimization Algorithms, Tsinghua University press. (2006).

[11] G.N. Xuan, R.W. Cheng, Genetic Algorithms and Engineering Optimization, Tsinghua University press. (2004).

[12] S. Fong, X. Wang, Q. Xu, et al., Recent advances in metaheuristic algorithms: Does the Makara dragon exist?. Journal of Supercomputing. 72 (2016) 1-23.

[13] K. Sörensen, Metaheuristics—the metaphor exposed, International Transactions in Operational Research. 22

ACCEPTED MANUSCRIPT

(2013) 3-18.

[14] W. Long, J. Jiao, X. Liang, et al., Inspired grey wolf optimizer for solving large-scale function optimization problems, Applied Mathematical Modelling. 60 (2018).

[15] C. Lu, L. Gao, J. Yi, Grey Wolf Optimizer with Cellular Topological Structure, Expert Systems with Applications. 107 (2018).

[16] S. Gupta, K. Deep, A novel Random Walk Grey Wolf Optimizer, Swarm & Evolutionary Computation. (2018).

[17] A.A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, Applied Soft Computing. 60 (2017) 115-134.

[18] S. Saremi, S.Z. Mirjalili, S.M. Mirjalili, Evolutionary population dynamics and grey wolf optimizer, Neural Computing & Applications. 26 (2015) 1257-1263.

[19] R.A. Ibrahim, M.A. Elaziz, S. Lu, Chaotic Opposition-Based Grey-Wolf Optimization Algorithm based on Differential Evolution and Disruption Operator for Global Optimization, Expert Systems with Applications. (2018).

[20] X. Zhang, Q. Kang, J. Cheng, et al., A Novel Hybrid Algorithm Based on Biogeography-Based Optimization and Grey Wolf Optimizer, Applied Soft Computing. (2018).

[21] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, Neural Computing & Applications. 27(2016) 495-513.

[22] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Advances in Engineering Software. 95 (2016) 51-67.

[23] S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, Knowledge-Based Systems. 96 (2016) 120-133.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Grey wolf optimization algorithm is the object of study.

Defect of GWO is pointed out and the reason is determined.

Other optimization algorithms may have a similar defect.

The test method is proposed, and it works well.

Looking forward to the further development of optimization algorithms

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 1

Experimental parameters setting.

Optimization algorithm	Parameters
CS	Pa=0.25
CSO	G = 10; rPercent = 0.15
	hPercent = 0.7; mPercent = 0.5
DE	F0 = 0.5; CR = 0.6
PSO	c1=2; c2=2
GSA	G_0=100; *a*=20

Table 2

Original test functions used in the experiments of this paper.

Formula	Dim	Range	Optimum *f*
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	*f(0,0……0)=0*
$f_2(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	$[-10, 10]^n$	*f(0,0……0)=0*
$f_3(x)=\sum_{i=1}^{n} i x_i^2$	30	$[-100, 100]^n$	*f(0,0……0)=0*
$f_4(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	$[-100, 100]^n$	*f(0,0……0)=0*
$f_5=\sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	30	$[-100, 100]^n$	*f(0,0……0)=0*
$f_6(x)=2x_1^2-1.05x_1^4+\frac{x_1^6}{6}+x_1x_2+x_2^2$	2	$[-100, 100]^n$	*f(0,0……0)=0*
$f_7(x)=(x_i+2x_{i+1}-7)^2+(2x_i+x_{i+1}-5)^2$	2	$[-10, 10]^n$	*f(1,3)=0*
$f_8(x)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	30	$[-100, 100]^n$	*f(π, π)=-29*
$f_9(x)=\sum_{i=1}^{n}100(x_{i+1}-x_i^2)^2+(x_i-1)^2$	30	$[-100, 100]^n$	*f(1,1……1)=0*

ACCEPTED MANUSCRIPT

Table 3

Experimental results of original test functions obtained by all optimization algorithms.

Function	Type	GWO	CS	CSO	DE	PSO	GSA
f_1	Mean	1.82242E-99	0.32662919	7.31899E-06	1.83256542	0.02151131	0.00116140
	Std	4.80930E-99	0.06574119	1.79623E-05	3.62262683	0.0374675[illegible]	0.00[illegible]39245
f_2	Mean	0	52.64727696	0.93168147	1.92802356	13.430[illegible]9111	2.98991811
	Std	0	7.67425729	1.68178371	2.86590318	4.0[illegible]036319	4.37781985
f_3	Mean	1.19784E-100	0.04169872	1.90195E-07	0.16574280	0.[illegible]154368[illegible]	0.64752292
	Std	3.81359E-100	0.01024475	3.57029E-07	0.22270017	0.54[illegible]	0.10726552
f_4	Mean	1.193636E-55	6.66745656	7.57196E-09	4.48063088	1[illegible]7168635	4.12938133
	Std	1.657736E-55	3.96129526	4.76758E-09	2.883397[illegible]7	32[illegible]8459275	5.77833750
f_5	Mean	9.44113E-72	10.79713944	0.25548092	15.230553[illegible]	40[illegible]4900203	30.88245954
	Std	1.52458E-71	15.60696731	0.43766553	25[illegible]44881[illegible]	68.89827562	36.94228843
f_6	Mean	0	4.43864E-32	1.13451E-82	0.01[illegible]9250	8.04322E-12	1.66561E-22
	Std	0	9.41655E-32	5.35398E-82	0.02[illegible]9756	5.64992E-11	1.66558E-21
f_7	Mean	2.29168E-08	0	2.94292E-06	0.03418[illegible]67	1.60855E-12	4.77788E-23
	Std	1.96387E-08	0	4.31629E-06	0.07653591	6.16840E-12	5.89145E-21
f_8	Mean	-8.35965732	-5.67355857	-4.867[illegible]	-26.3179683	-6.13932234	-11.18951660
	Std	2.62445599	0.54958701	4.61916391	3.4148354	1.44251824	5.60182021
f_9	Mean	25.74613723	25.78475420	0.02[illegible]454[illegible]5	20.20469515	74.63474585	26.30679270
	Std	0.80164596	1.46999629	[illegible]07194[illegible]70	11.50480378	68.97631752	7.55555553

Table 4

Modified test functions used in the experiments of this paper.

Formula	k	Dim	Range	Optimum *f*
$f_1(x)=\sum_{i=1}^{n}(x_i-k)^2$	*k=1*	30	$[-100, 100]^n$	*f(k,k⋯)=0*
$f_2(x)=\sum_{i=1}^{n}[(x-k)^2-10\cos(2\pi x_i-2k\pi)+10]$	*k=1*	30	$[-10, 10]^n$	*f(k,k⋯)=0*
$f_3(x)=\sum_{i=1}^{n}i\cdot(x-k)^2$	*k=1*	30	$[-100, 100]^n$	*f(k,k⋯)=0*
$f_4(x)=\sum_{i=1}^{n}\lvert x_i-k\rvert+\prod_{i=1}^{n}\lvert x_i-k\rvert$	*k=1*	30	$[-100, 100]^n$	*f(k,k⋯)=0*
$f_5=\sum_{i=1}^{n}[\sum_{j=1}^{i}(x_j-k)]^2$	*k=1*	30	$[-100, 100]^n$	*f(k,k⋯)=0*
$f_6(x)=2(x_1-k)^2-1.05(x_1-k)^4+\frac{(x_1-k)^6}{6}+(x_1-k)(x_2-k)+(x_2-k)^2$	*k=1*	2	$[-100, 100]^n$	*f(k,k⋯)=0*
$f_7(x)=(x_i+2x_{i+1}-7k)^2+(2x_i+x_{i+1}-5k)^2$	*k=0*	2	$[-10, 10]^n$	*f(0,0)=0*
$f_8(x)=-\cos(x_1+\pi-k)\cos(x_2+\pi-k)\exp[-(x_1-k)^2-(x_2-k)^2)$	*k=0*	30	$[-100, 100]^n$	*f(0,0⋯)=-29*
$f_9(x)=\sum_{i=1}^{n-1}100(x_{i+1}-x_i^2)^2+(x_i-k)^2$	*k=0*	30	$[-100, 100]^n$	*f(0,0⋯)=0*

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 5

Experimental results of modified test functions obtained by all optimization algorithms.

Function	Type	GWO	CS	CSO	DE	PSO	GSA
f_1	Mean	1.67004178	0.32757422	4.94980716	1.37231998	0.0[illegible]2504[illegible]	0.00102402
	Std	2.54096737	0.0720197	4.40772419	2.01550646	[illegible]3745519	0.00174770
f_2	Mean	18.56306835	53.94949389	0.44761702	2.00825271	[illegible].12003[illegible]5	2.98487717
	Std	2.28897523	8.54437796	1.12120796	2.45219376	1[illegible].02400206	5.37277890
f_3	Mean	4.89822451	0.03952762	9.65683637	0.189672[illegible]	[illegible]0599244	0.57488503
	Std	5.36358008	0.00861146	17.20659446	0.2570[illegible]494	[illegible]53302931	0.12370356
f_4	Mean	0.81821518	1.36352596	9.85775563	3.672894[illegible]	10.08298381	4.07536407
	Std	0.75615353	4.95975155	6.19056526	[illegible].8790[illegible]24	10.65353004	6.21646251
f_5	Mean	74.59519606	11.09906404	11.58865013	15.879972[illegible]	38.33442806	30.48281321
	Std	36.84247475	17.30235126	18.94072251	24.[illegible]307386	44.23048166	35.37041570
f_6	Mean	3.28758E-09	2.91632E-31	8.63839E-07	0.016[illegible]5417	3.28758E-11	2.11962E-21
	Std	3.21734E-09	1.86708E-30	1.73300E-06	0.02687118	3.21734E-11	2.24235E-21
f_7	Mean	0	0	8.384[illegible]E-53	0.02565173	8.21536E-13	1.70788E-22
	Std	0	0	2.[illegible]0219E-52	0.03613052	5.76747E-12	5.43777E-21
f_8	Mean	-29	-5.5981919	-28.[illegible]9998	-25.74256084	-6.19889764	-10.77220171
	Std	0	0.64304489	[illegible]45066[illegible]-08	3.51184731	1.26099066	5.92565743
f_9	Mean	7.08270E-96	25.32505897	2[illegible]79796031	17.86135228	71.55002074	29.46532259
	Std	2.81381E-95	1.35683962	19.56277686	10.97086413	57.6267928	29.46532259

Table 6

The cost of GWO in practical applications.

Test function	$X^*=0$		$X^*\neq0$	
	N	I	N	I
Three-bar truss design problem	198	188	23	13
Pressure vessel design problem	169	159	34	24
Gear train design problem	23	25	8	12
Cantilever beam design	70	100	40	50

ACCEPTED MANUSCRIPT

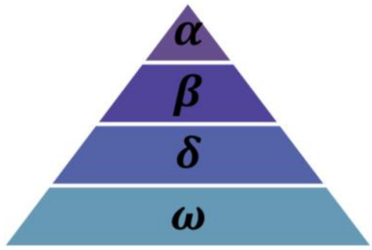


Fig. 1. Different grades of grey wolves.

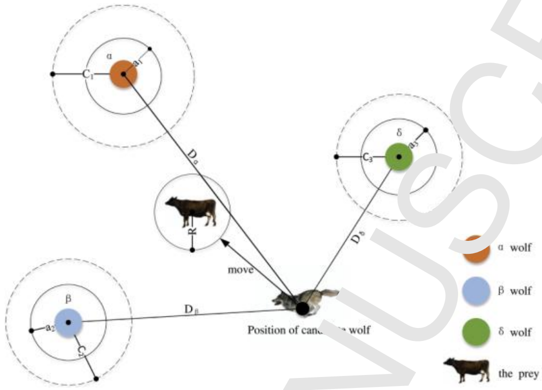
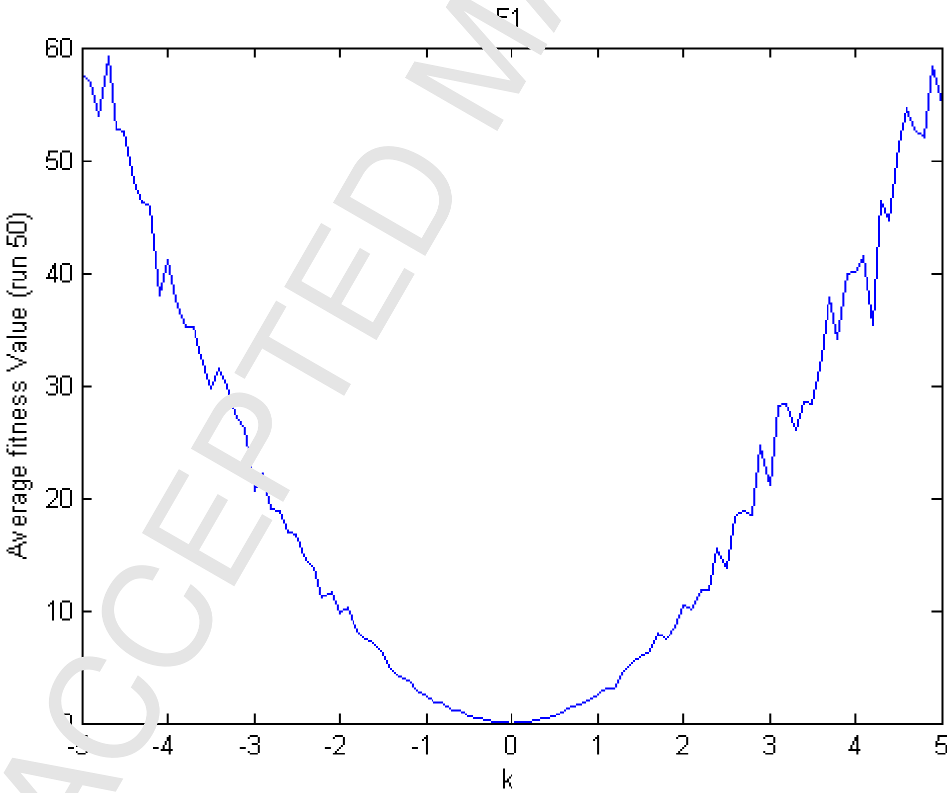
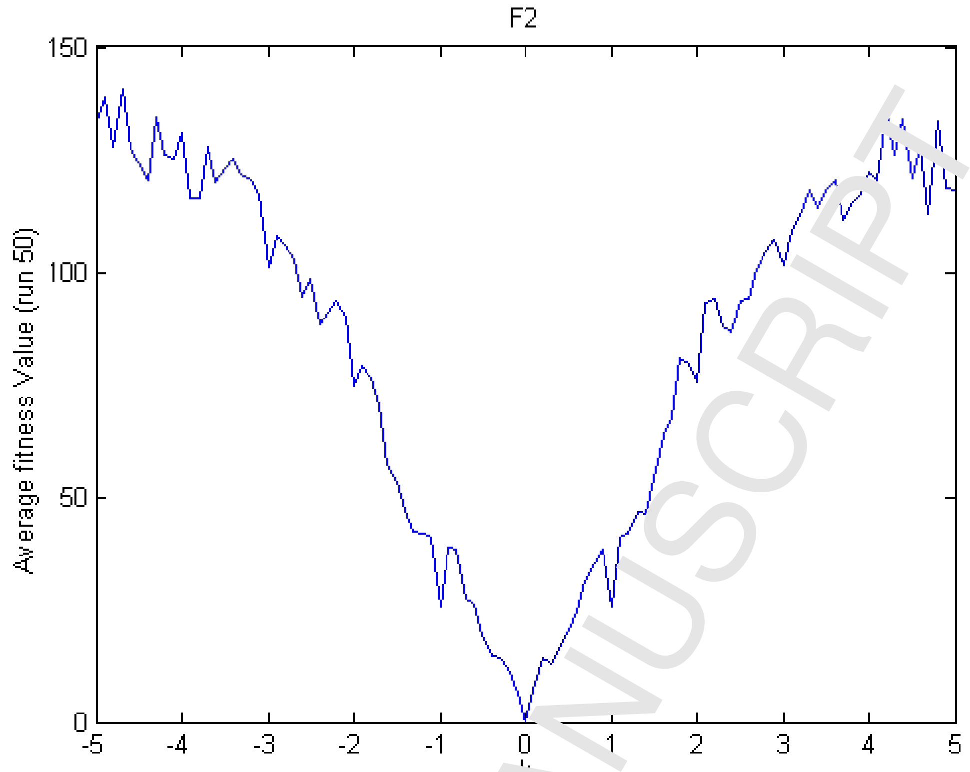


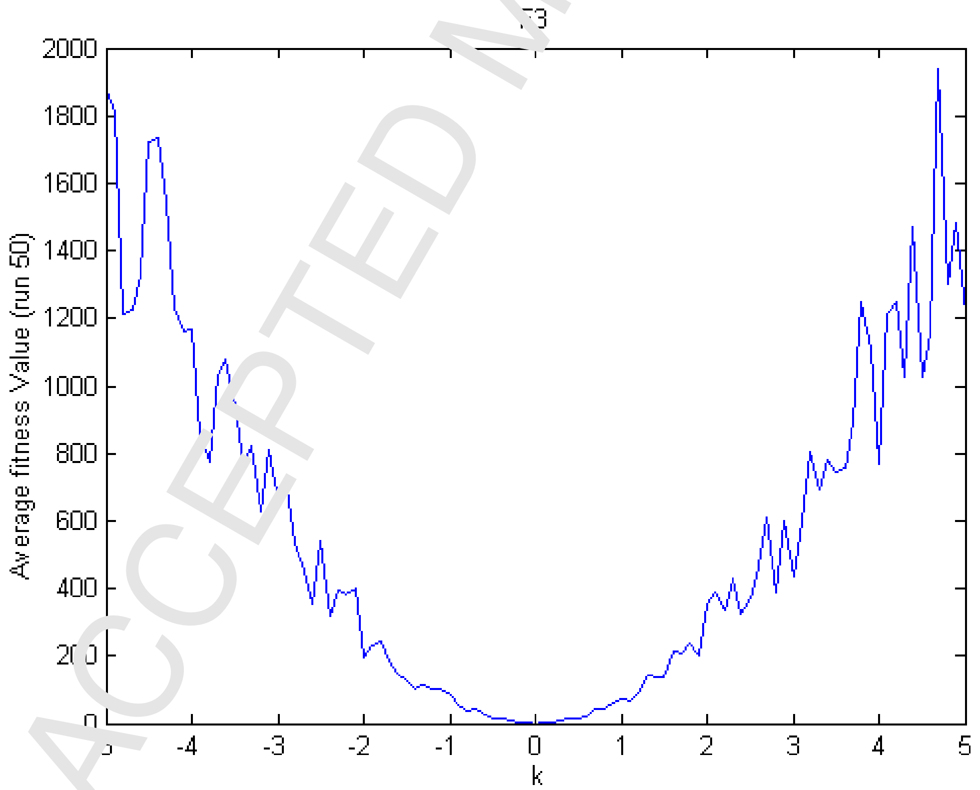
Fig. 2. Position updating in GWO.



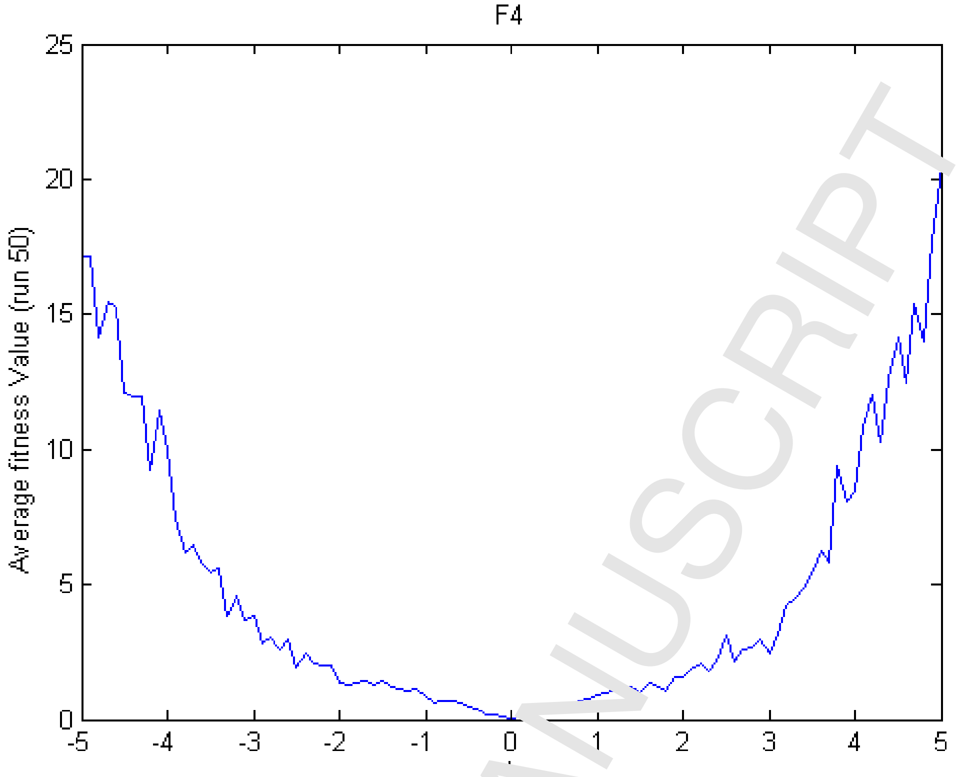
ACCEPTED MANUSCRIPT

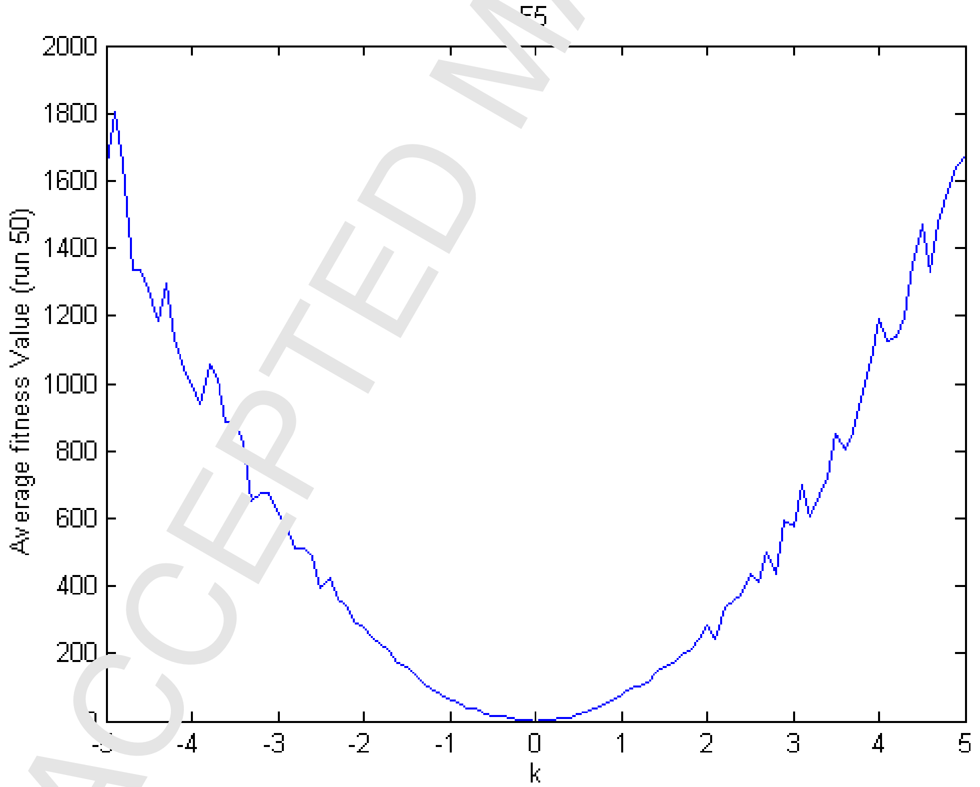
ACCEPTED MANUSCRIPT



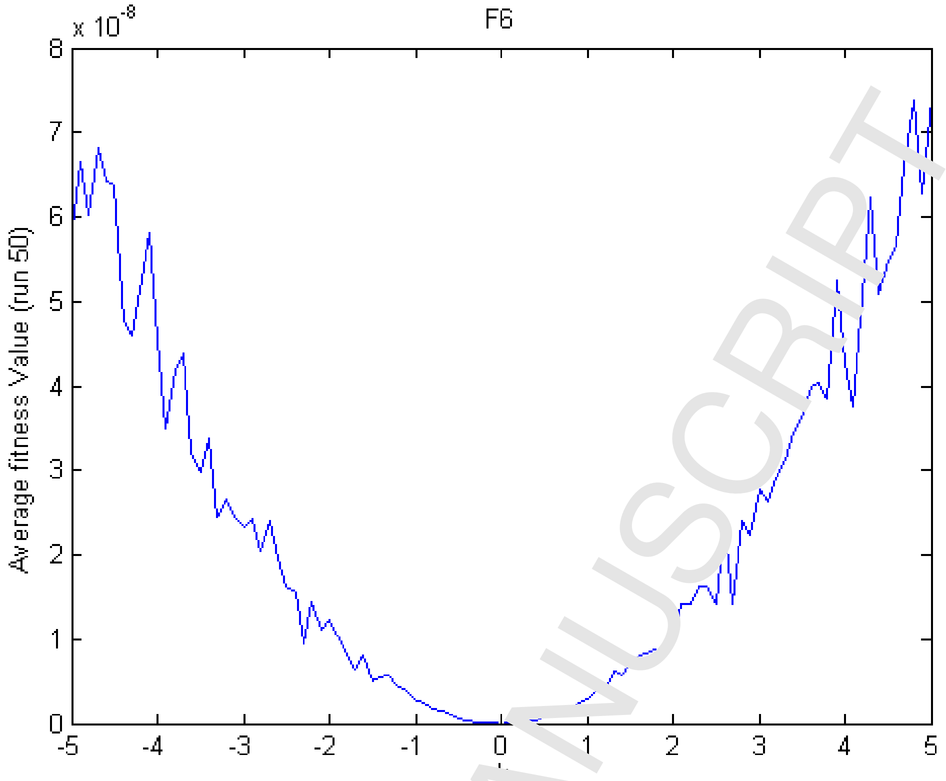


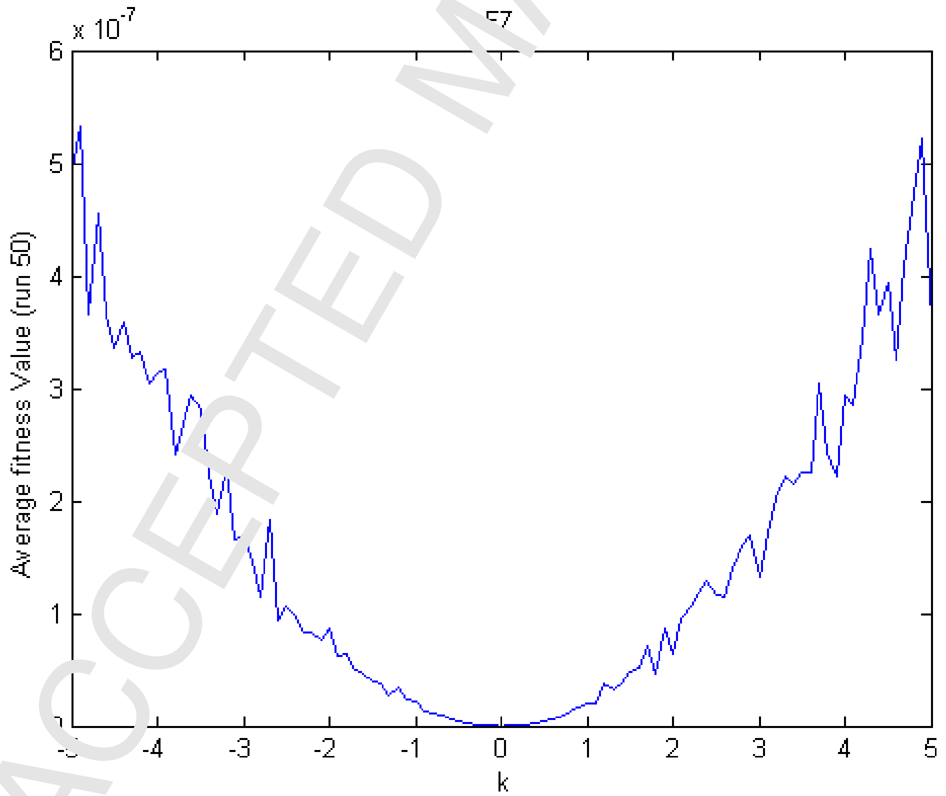
ACCEPTED MANUSCRIPT

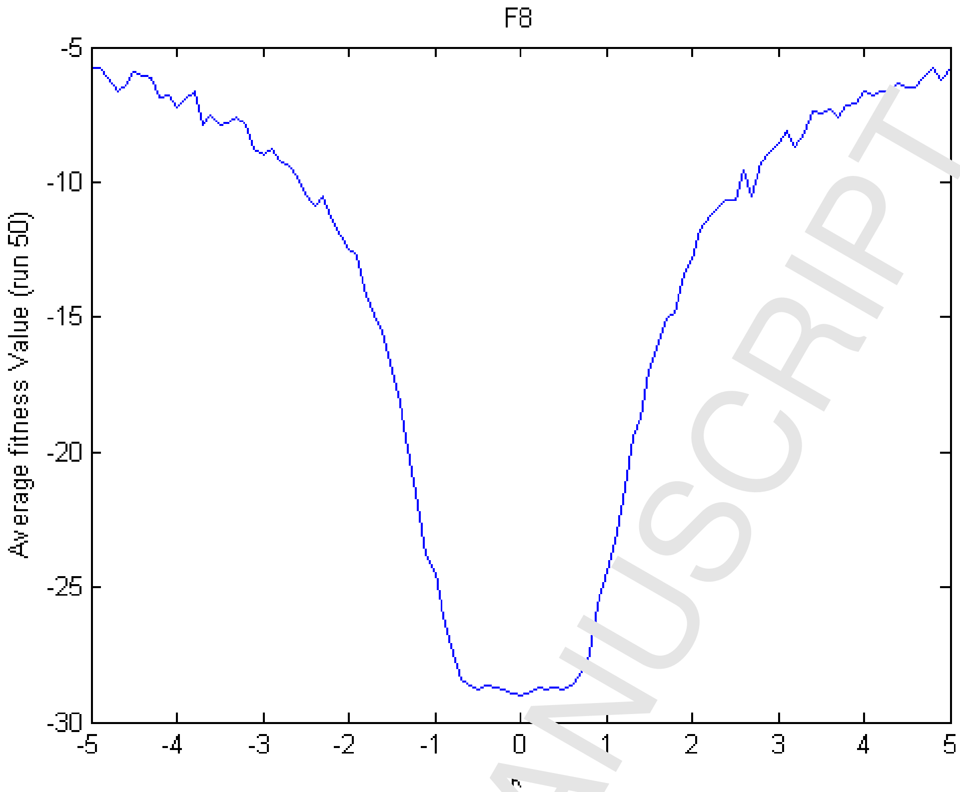




ACCEPTED MANUSCRIPT







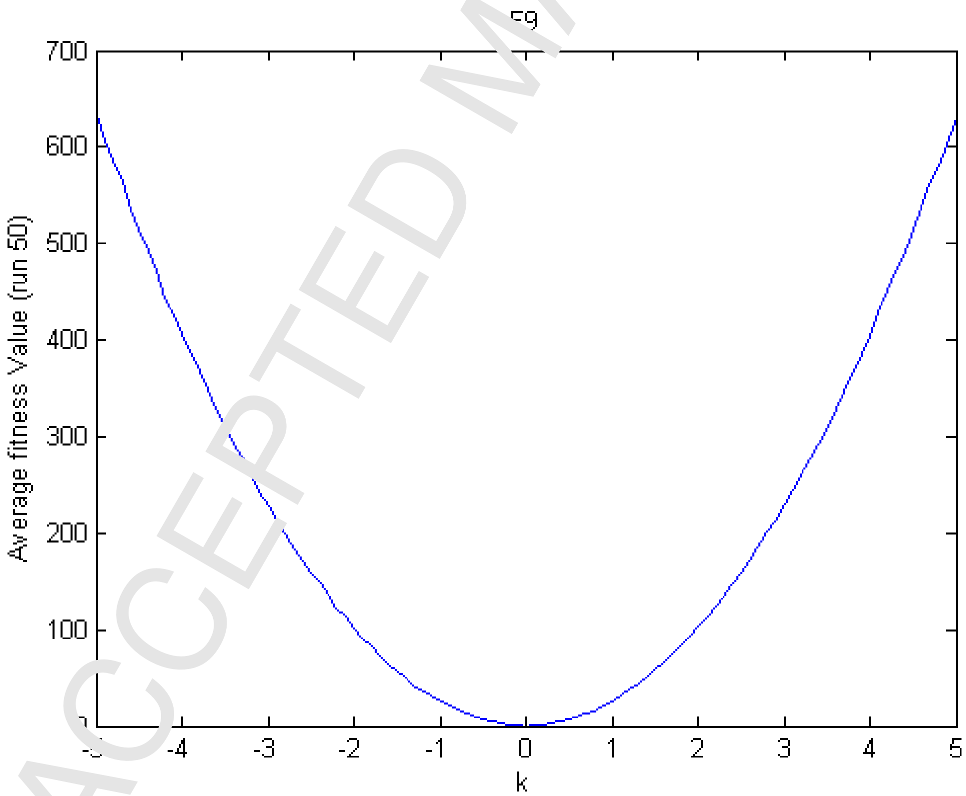


Fig.3. Best-average curve of GWO changed with *k*.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

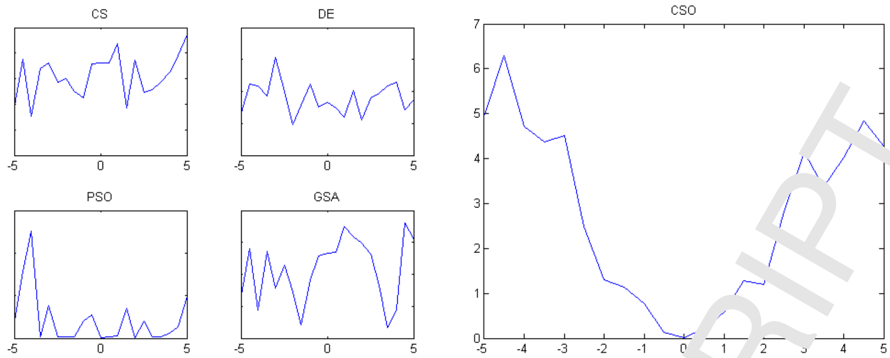


Fig.4. Best-average curve of other algorithms on f_1 changed with k.

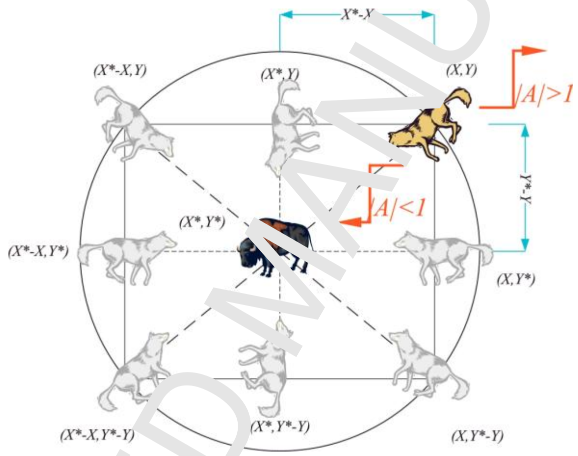


Fig.5. Position updating mechanism of search agents and effects of A on it. (Mirjalili et al., 2014).

ACCEPTED MANUSCRIPT