16

Grey Wolf Optimizer – Modifications and Applications

Ahmed F. Ali
Department of Computer Science
Suez Canal University, Ismaillia, Egypt

Mohamed A. Tawhid
Department of Mathematics and Statistics
Faculty of Science, Thompson Rivers University, Kamloops, Canada

CONTENTS

16.1 Introduction

The grey wolves live in groups and they have a special dominant social structure. In 2014, Mirjalili et al. proposed a swarm intelligence algorithm [4] which was called grey wolf optimizer (GWO). The GWO algorithm mimics the social behavior of the grey wolves. Many researchers have applied the GWO algorithm due to its efficiency in various applications, for example, for CT liver segmentation [1], minimizing potential energy function [8], feature selection [2], [11], minimax and integer programming problems [9], global optimization problem [10], flow shop scheduling problem [3], optimal reactive power dispatch problem [7], and the casting production scheduling [12]. The rest of the organization of our chapter is as follows. We present the original GWO algorithm in Section 16.2. We modify the GWO algorithm in Section 16.3. In Section 16.4, we solve the engineering optimization problem by the chaotic grey wolf optimization (CGWO) algorithm. Finally, we summarize the conclusion in Section 16.5.

16.2 Original GWO algorithm in brief

In this section, we present the main steps of the GWO algorithm. These steps are shown in Algorithm 18.

Algorithm 18 Grey wolf optimizer algorithm.

Set the initial values of the population size n, parameter a and the maximum number of iterations Max_{itr}
Set $t := 0$
for $(i = 1 : i \leq n)$ **do**
    Generate an initial population $\vec{X_i}(t)$ randomly
    Evaluate the fitness function of each search agent (solution) $f(\vec{X}_i)$
end for
Assign the values of the first, second and the third best solutions $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$, respectively
repeat
    for $(i = 1 : i \leq n)$ **do**
        Decrease the parameter a from 2 to 0
        Update the coefficients $\vec{A}$ and $\vec{C}$ as shown in Equations (16.4)–(16.5)
        Update each search agent in the population as shown in Equations (16.1)–(16.3)
        Evaluate the fitness function of each search agent $f(\vec{X}_i)$

```
    end for
    Update the vectors X_α, X_β and X_δ.
    Set t = t + 1
until (t ≥ Max_itr)                    ▷ Termination criteria are satisfied
Produce the best solution X_α
```

16.2.1 Description of the original GWO algorithm

The standard GWO starts by setting the initial parameters of the population size n, the parameter a and the maximum number of iterations Max_{itr}. The iteration counter t is initialized where $t = 0$. The initial population n is randomly generated and each search agent (solution) $\vec{X_i}$ is evaluated by calculating its fitness function $f(\vec{X_i})$. The overall best three solutions are assigned according to their fitness values which are alpha α, beta β and the delta δ solutions $\vec{X_\alpha}$, $\vec{X_\beta}$ and $\vec{X_\delta}$, respectively. The main loop is repeated until the termination criterion is satisfied. Each search agent (solution) in the population is updated according to the position of the α, β and δ solutions as shown in Equations 16.1–16.3.

$$\begin{aligned} \vec{D_\alpha} &= |\vec{C_1} \cdot \vec{X_\alpha} - \vec{X}|, \\ \vec{D_\beta} &= |\vec{C_2} \cdot \vec{X_\beta} - \vec{X}|, \\ \vec{D_\delta} &= |\vec{C_3} \cdot \vec{X_\delta} - \vec{X}|, \end{aligned} \tag{16.1}$$

$$\begin{aligned} \vec{X_1} &= \vec{X_\alpha} - \vec{A_1} \cdot (\vec{D_\alpha}), \\ \vec{X_2} &= \vec{X_\beta} - \vec{A_2} \cdot (\vec{D_\beta}), \\ \vec{X_3} &= \vec{X_\delta} - \vec{A_3} \cdot (\vec{D_\delta}), \end{aligned} \tag{16.2}$$

$$\vec{X}(t+1) = \frac{\vec{X_1} + \vec{X_2} + \vec{X_3}}{3}, \tag{16.3}$$

The parameter a is gradually decreased from 2 to 0 and the coefficients $\vec{A}$ and $\vec{C}$ are updated as shown in Equations 16.4–16.5.

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} \cdot \vec{a} \tag{16.4}$$

$$\vec{C} = 2 \cdot \vec{r_2} \tag{16.5}$$

where components of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations and $\vec{r_1}$, $\vec{r_2}$ are random vectors in $[0, 1]$.

Each search agent (solution) in the population is evaluated by calculating its fitness function $f(\vec{X_i})$. The first three best solutions are updated $\vec{X_\alpha}$, $\vec{X_\beta}$ and $\vec{X_\delta}$, respectively. The iteration counter is increased where $t = t + 1$. Once the termination criteria are satisfied, the algorithm is terminated and the overall best solution $\vec{X_\alpha}$ is produced.

16.3 Modifications of the GWO algorithm

Recently, many metaheuristic algorithms have been developed and suggested for global search. These developed algorithms can improve computational efficiency and solve various optimization problems [1], [8], [2], [11], [9], [10], [3], [5], [7], [12]. Among these algorithms, grey wolf optimizer (GWO) was developed in 2014 by Mirjalili et al. as a swarm intelligence algorithm [4]. The GWO algorithm simulates the social behavior of the grey wolves. On the other hand recent studies in nonlinear dynamics such as chaos, have attracted more attention in many areas [16] . One of these areas is the integration of chaos into optimization algorithms to improve and enhance certain algorithm-dependent parameters [17]. Researchers have used chaotic sequences to tune up parameters in metaheuristic optimization algorithms such as genetic algorithms [18], particle swarm optimization [19], ant and bee colony optimization [20], and simulated annealing [21]. Empirical studies show that chaos can improve the capability of standard metaheuristics when a chaotic map exchanges a fixed parameter. It turns out the solutions generated from such exchange may have higher diversity. For this reason, it is desirable to perform more studies by introducing chaos to recent metaheuristic algorithms. In this section, we replace the random vectors $\vec{r_1}$, $\vec{r_2}$ in Equations 16.4–16.5 by the chaotic maps $C1, C2$. The proposed algorithm is called Chaotic Grey Wolf Optimization (CGWO) algorithm. Invoking the two chaotic map in the standard GWO algorithm can help it to escape from being trapped in local minima due to increasing the range of the random numbers from [0,1] to [-1,1].

16.3.1 Chaotic maps

Although the chaotic maps have no random variables, they show a random behavior. In Table 16.1, we present the mathematical forms of these maps. The GWO suffers from being trapped in local minima like other meta-heuristics algorithms. We invoke two chaotic maps in GWO instead of using the standard random parameters to increase the diversity of the search and avoid being trapped in local minima. We use the initial point $x^0 = 0.7$ as chosen in [6].

16.3.2 Chaotic grey wolf operator

We replace the random vectors $\vec{r_1}$, $\vec{r_2}$ in Equations 16.4–16.5 by two chaotic maps with range [-1,1] as shown in Table 16.1. The two new equations are

TABLE 16.1
Chaotic maps.

No	Name	Chaotic map	Range
C1	Chebyshev [13]	$x_{i+1} = \cos(i \cos^{-1}(x_i))$	(-1,1)
C2	Iterative [15]	$x_{i+1} = \sin(\frac{a\pi}{x_i}, a = 0.7$	(-1,1)

presented as follows.

$$\vec{A} = 2\vec{a} \cdot \vec{C_1} \cdot \vec{a} \tag{16.6}$$

$$\vec{C} = 2 \cdot \vec{C_2} \tag{16.7}$$

16.4 Application of GWO algorithm for engineering optimization problems

In this section, we test the efficiency of the proposed CGWO by solving five engineering optimization problems as follows.

16.4.1 Engineering optimization problems

In this subsection, we give a brief description for five engineering optimization problems. These problems are the welded beam design problem, pressure vessel design problem, speed reducer design problem and three-bar truss design problem.

16.4.1.1 Welded beam design problem

The main objective of a welded beam design problem is minimizing the cost of its structure which consists of a beam A and the required weld to hold it to member B subject to constraints on shear stress τ, beam bending stress θ, buckling load on the bar P_c, beam end deflection δ. The four design variables $h(x_1), l(x_2), t(x_3)$ and $b(x_4)$ are shown in Figure 16.1. The mathematical form of the welded beam problem is shown in Equation 16.8.

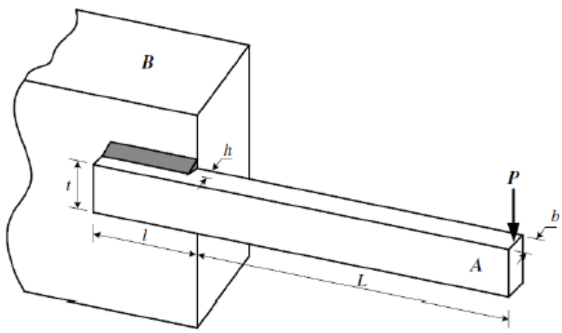

FIGURE 16.1
Welded beam design problem.

$$
\begin{aligned}
Minimize \quad f(x) = &\quad 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \qquad (16.8)\\
Subject\ to & \\
g_1(x) = &\quad \tau(x) - \tau_{max} \leq 0\\
g_2(x) = &\quad \sigma(x) - \sigma_{max} \leq 0\\
g_3(x) = &\quad x_1 - x_4 \leq 0\\
g_4(x) = &\quad 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 0.5 \leq 0\\
g_5(x) = &\quad 0.125 - x_1 \leq 0\\
g_6(x) = &\quad \delta(x) - \delta_{max} \leq 0\\
g_7(x) = &\quad P - P_c(x) \leq 0
\end{aligned}
$$

where the other parameters are defined as follows.

$$
\begin{aligned}
\tau(x) = &\quad \sqrt{((\tau')^2 + (\tau'')^2 + \frac{2\tau'\tau''x_2}{2R}}, \quad \tau' = \frac{p}{\sqrt{2}x_1x_2}\\
\tau'' = &\quad \frac{MR}{J}, \quad M = P(L + \frac{x_2}{2}), \quad R = \sqrt{\left(\frac{x_1 + x_3}{2}\right)^2 + \frac{x_2^2}{4}}\\
J = &\quad 2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \quad \sigma(x) = \frac{6PL}{x_4x_3^2}\\
\delta(x) = &\quad \frac{4PL^3}{Ex_4x_3^3}, \quad P_c(x) = \frac{4.013\sqrt{EGx_3^2x_4^6/36}}{L^2}\left(1 - \frac{x^3}{2L}\sqrt{\frac{E}{4G}}\right) \qquad (16.9)
\end{aligned}
$$

where $P = 6000$lb, $L = 14$, $\delta_{max} = 0.25$,in, $E = 30,106$ psi, $G = 12,106$ psi, $\tau_{max} = 13,600$psi, $\sigma_{max} = 30,000$ psi and $0.1 \leq x_i \leq 10.0$ $(i = 1, 2, 3, 4)$.

16.4.1.2 Pressure vessel design problem

The objective of this problem is to minimize the total cost of material, forming and welding. Figure 16.2 shows the following four parameters, T_s (x_1, thickness

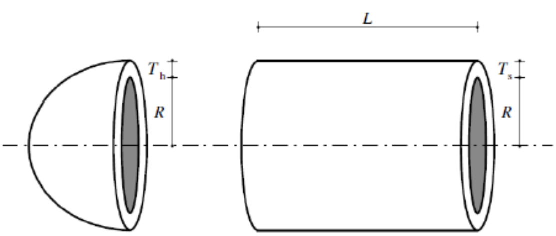

FIGURE 16.2
Pressure vessel design problem.

of the shell), T_h (x_2, thickness of the head), R (x_3, inner radius) and L. (x_4, length of the cylindrical section of the vessel without the head). T_s and T_h are integer multiples of 0.0625in, R and L are continuous variables. The mathematical form of this problem is shown in Equation 16.10.

$$\begin{aligned}
Minimize \quad f(x) = &\ 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\
&+3.1661x_1^2x_4 + 19.84x_1^2x_3 \qquad (16.10)\\
Subject \quad to & \\
g_1(x) = &\ -x_1 + 0.0193x_3 \\
g_2(x) = &\ -x_2 + 0.00954x_3 \leq 0 \\
g_3(x) = &\ -\pi x_3^2x_4 - 4/3\pi x_3^3 + 1296000 \leq 0 \\
g_4(x) = &\ x_4 - 240 \leq 0
\end{aligned}$$

where $1 \leq x_1 \leq 99$, $1 \leq x_2 \leq 99$, $10 \leq x_3 \leq 200$ and $10 \leq x_4 \leq 200$.

16.4.1.3 Speed reducer design problem

In this problem, we minimize a gear box volume and weight subject to the following constraints as shown in Equation 16.11. In this problem, we have seven design variables $x_1 - x_7$, which can be described as the following. x_1 is a width of the gear face (cm), x_2 teeth module (cm), x_3 number of pinion teeth, x_4 shaft 1 length between bearings (cm), x_5 shaft 2 length between bearing (cm), x_6 diameter of shaft 1 (cm) and x_7 diameter of shaft 2 (cm). The speed reducer design problem is shown in Figure 16.3.

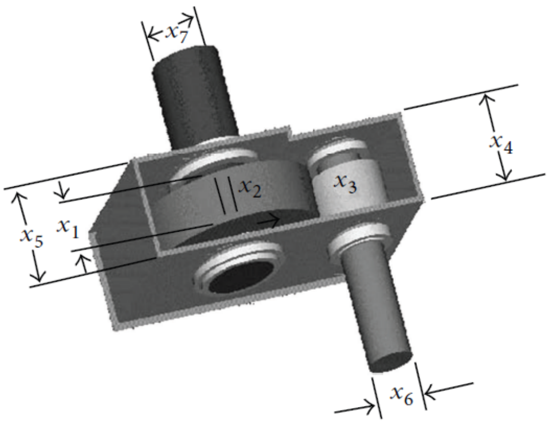

FIGURE 16.3
Speed reducer design problem.

$$\begin{aligned} Minimize \quad f(x) = \quad & 0.7854x_1x_2^2(3.3333x_3^2 \\ & +14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) \\ & +7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \end{aligned} \tag{16.11}$$

Subject to

$$\begin{aligned} g_1(x) = \quad & \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \\ g_2(x) = \quad & \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \\ g_3(x) = \quad & \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0 \\ g_4(x) = \quad & \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0 \\ g_5(x) = \quad & \frac{[(745x_4/x_2x_3)^2 + 16.9 \times 10^6]^0.5}{110.0x_6^3} - 1 \leq 0 \\ g_6(x) = \quad & \frac{[(745x_5/x_2x_3)^2 + 157.5.9 \times 10^6]^0.5}{85.0x_7^3} - 1 \leq 0 \\ g_7(x) = \quad & \frac{x_2x_3}{40} - 1 \leq 0 \\ g_8(x) = \quad & \frac{5x_2}{x_1} - 1 \leq 0 \\ g_9(x) = \quad & \frac{x_1}{12x_2} - 1 \leq 0 \\ g_{10}(x) = \quad & \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \\ g_{11}(x) = \quad & \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0 \end{aligned} \tag{16.12}$$

where $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8 \leq$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4 \leq 8.3$, $7.3 \leq x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, $5 \leq x_7 \leq 5.5$

16.4.1.4 Three-bar truss design problem

In the three-bar truss design problem, we try to minimize the weight $f(x)$ of three bar trusses subject to some constraints as shown in Equation 16.13.

$$Minimize \quad f(x) = \quad (2\sqrt{2}x_1 + x_2) \times l \tag{16.13}$$

Subject to

$$\begin{aligned} g_1(x) = \quad & \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\ g_2(x) = \quad & \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \\ g_3(x) = \quad & \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0 \end{aligned}$$

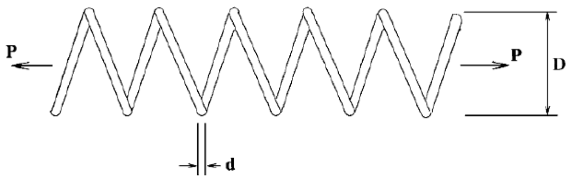

FIGURE 16.4
Tension compression spring problem.

where $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$, $l = 100$ cm, $P = 2kN/cm^2$, and $\sigma = 2kN/cm^2$. The minimum weighted structure should be achieved by determining the optimal cross-sectional areas x_1, and x_2.

16.4.1.5 Tension compression spring problem

The last problem is the tension compression spring problem. In this problem, we need to minimize the weight $f(x)$ of a tension compression spring design subject to three constraints as shown in Equation 16.14. The mean coil diameter $D(x_2)$, the wire diameter $d(x_1)$ and the number of active coils $P(x_3)$ are the design variables as shown in Figure 16.4.

$$\begin{aligned} &Minimize \quad & f(x) =& \quad (x_3+2)x_2x_1^2 \\ &Subject \;\; to \\ & & g_1(x) =& \quad 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \\ & & g_2(x) =& \quad \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\ & & g_3(x) =& \quad 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0 \end{aligned} \tag{16.14}$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$ and $2 \leq x_3 \leq 15$.

16.4.2 Description of experiments

The proposed algorithm was programmed in Matlab. We set the population size $n = 20$ and the $Max_{itr} = 1000$. We handled the constraints by transforming the constrained optimization problems to an unconstrained optimization problem as shown in [14].

16.4.3 Convergence curve of CGWO with engineering optimization problems

In Figure 16.5, we show the performance and the convergence curve of the proposed CGWO by plotting the number of iterations versus the function

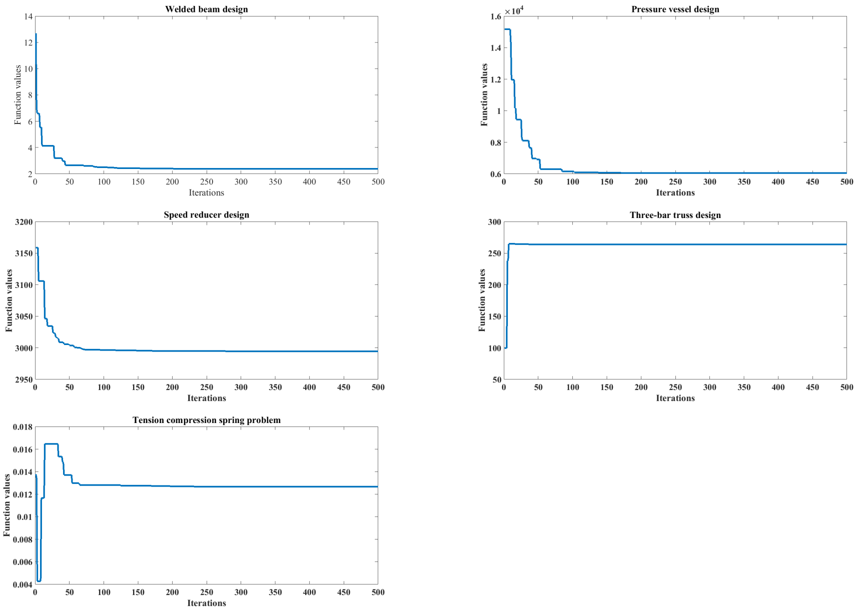

FIGURE 16.5
Convergence curve of the proposed CGWO algorithm with engineering optimization problems.

values for the engineering optimization problems. We can see that the CGWO can reach to near optimal global minimum for all problems in a small number of iterations.

16.4.4 Comparison between CGWO and GWO with engineering optimization problems

We verify the efficiency of the proposed CGWO algorithm by comparing it with the GWO algorithm. We report the results in Table 16.2 and the best

TABLE 16.2
Comparison between CGWO and GWO with engineering optimization problems.

Design problem	GWO	CGWO
Welded beam design	2.382447652	**2.380956951**
Pressure vessel design	6076.292065	**6059.714335**
Speed reducer design	3000.385887	**2994.471066**
Three-bar truss design	263.896704	**263.8958434**
A tension/compression	0.012717623	**0.012665233**

results in **bold text**. The results in Table 16.2 show that the proposed CGWO outperforms the GWO which means that invoking the chaotic maps in CGWO can enhance and obtain better results than the standard GWO.

16.5 Conclusions

In this chapter, we present a proposed algorithm to solve five engineering optimization problems. The proposed algorithm is called Chaotic Grey Wolf Optimization CGWO. Replacing the random vectors in the standard GWO by chaotic maps with range [-1,1] increases the diversity of the proposed algorithm and avoids being trapped at the local minima. In order to verify the efficiency of CGWO, we compare it with the standard GWO algorithm. The results show that the proposed CGWO is better than GWO in all cases.

References

1. A.F. Ali, A. Mostafa, G.I, Sayed, M.A. Elfattah and A.E. Hassanien. “Nature inspired optimization algorithms for ct liver segmentation”. In *Medical Imaging in Clinical Applications* (pp. 431–460), 2016. Springer.
2. E.Emary, H.M. Zawbaa and A.E. Hassanien. “Binary grey wolf optimization approaches for feature selection”. *Neurocomputing*, vol. 172, pp. 371–381, 2016.
3. G. M. Komaki and V. Kayvanfar. “Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time”. *Journal of Computational Science*, vol. 8, pp. 109-120, 2015.
4. S. Mirjalili, S. M. Mirjalili, and A. Lewis. “Grey Wolf Optimizer”. *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
5. A. Pasquet. “Cooperation and prey capture efficiency in a social spider, Anelosimus eximius (Araneae, Theridiidae)”. *Ethology*, vol. 90, pp. 121—133, 1991.
6. S. Saremi,S. Mirjalili, and A. Lewis. “Grasshopper optimisation algorithm: Theory and application”. *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
7. M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman. “Using the gray wolf optimizer for solving optimal reactive power

dispatch problem". *Applied Soft Computing*, vol. 32, pp. 286-292, 2015.

8. M.A. Tawhid and A.F. Ali. "A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function". *Memetic Computing*, vol. 9, no. 4, pp. 347–359, 2017.
9. M.A. Tawhid and A.F. Ali. "A simplex grey wolf optimizer for solving integer programming and minimax problems". *Numerical Algebra, Control & Optimization*, vol. 7, no. 3, pp. 301–323, 2017.
10. M.A. Tawhid and A.F. Ali. "Multidirectional Grey Wolf Optimizer Algorithm for solving global optimization problems". *International Journal of Computational Intelligence and Applications*, pp. 1850022, 2018.
11. Q. Tu, X. Chen, and X. Liu. "Multi-strategy ensemble grey wolf optimizer and its application to feature selection". *Applied Soft Computing*, vol. 76, pp. 16–30, 2019.
12. H. Qin, P. Fan, H. Tang, P. Huang, B. Fang and S. Pan. "An effective hybrid discrete grey wolf optimizer for the casting production scheduling problem with multi-objective and multi-constraint". *Computers & Industrial Engineering*, vol. 128, pp. 458–476, 2019.
13. N. Wang, L. Liu. "Genetic algorithm in chaos". *OR Trans*, vol. 5, pp. 1–10, 2001.
14. J.M. Yang, Y.P. Chen, J.T. Horng and C.Y. Kao. "Applying family competition to evolution strategies for constrained optimization". In *Lecture Notes in Mathematics* vol. 1213, pp. 201–211, New York, Springer, 1997.
15. G. Zhenyu, C. Bo, Y. Min, C. Binggang. "Self-adaptive chaos differential evolution Zhenyu". *Advances in Natural Computation*, pp. 972–975, 2006.
16. L.T. Pecora. "Carroll, synchronization in chaotic system". *Physical Review Letters* vol. 64, no. 8, pp. 821–824, 1990.
17. D. Yang, G. Li, G. Cheng. "On the efficiency of chaos optimization algorithms for global optimization". *Chaos, Solitons & Fractals*, vol. 34, pp. 136–1375, 2007.
18. G. Gharooni-fard, F. Moein-darbari, H. Deldari, A. Morvaridi. "Scheduling of scientific workflows using a chaos-genetic algorithm". *Procedia Computer Science*, vol. 1, pp. 1445–1454, 2010.
19. A.H. Gandomi, G.J. Yun, X.S. Yang, S. Talatahari. "Chaos-enhanced accelerated particle swarm algorithm". *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 2, pp. 327–340, 2013.

20. B. Alatas. “Chaotic bee colony algorithms for global numerical optimization”. *Expert Systems with Applications*, vol. 37, pp. 5682–5687, 2010.

21. J. Mingjun, T. Huanwen. “Application of chaos in simulated annealing”. *Chaos, Solitons & Fractals*, vol. 21, pp. 933–941, 2004.