An Improved Gradient based Optimizer Combined with Random Spare Strategy and Adaptive Weight Strategy

Ji Wu, Bihui Bao

Department of electrical and Electronic Engineering, Wenzhou Polytechnic, Wenzhou 325035, China.

Abstract

This paper proposes an improved gradient based optimizer (GBO) named RWGBO that combines two strategies. One is the random spare strategy and the other is the adaptive weight strategy. 23 test benchmark functions selected from IEEE CEC 2017 test functions are utilized to validate the effectiveness of the proposed RWGBO and compare and analyze the optimization capacity of RWGBO versus GBO.

Keywords

Gradient based Optimizer; Adaptive Weight Mechanism; Random Spare Strategy.

1. Introduction

With the increasing complexity and dimensionality of data, all information of feature space is unknown in various fields such as machine learning and industrial design, so it is impossible to consider the accurate mathematical model to solve all problems as the best solution [1]. In the era of big data [2], we need to develop more scalable and efficient optimization solvers for new complex problems. Recently, the application of meta-heuristic algorithms (MAs) to handle the problems of parameter identification in PV models has attracted the focus of researchers gradually. In conclusion, this paper mainly completes the following works. An improved GBO with adaptive weight mechanism and the random spare strategy is named RWGBO, which is tested in 23 IEEE CEC 2017 test functions [3]. The IEEE CEC 2017 test functions include unimodal functions, multimodal functions and composite functions, is used to verify the capability of the proposed RWGBO. The proposed algorithm RWGBO is compared with conventional and advanced algorithm to evaluate the performance. And it is applied in three classic practical engineering fields. The effectiveness of RWGBO is verified by describing and analyzing the classic practical problem in detail.

2. Methods

2.1 Gradient based optimizer (GBO)

Meta-heuristic methods such as GBO tend to generate random operators without the need to slope data based on stable performance. GBO is a new algorithm based on gradient and population. It mainly uses a set of vectors and two operators to explore the entire search space.

2.1.1 Initialization

each individual in the population is represented as a vector. Therefore, GBO contains N individuals in the D-dimensional search space. The vector is expressed as follows.

$$X_{n,d} = [X_{n,1}, X_{n,2}, \dots, X_{n,D}], \quad n = 1,2,\dots,N \quad d = 1,2,\dots,D \tag{1}$$

The initial population of GBO has randomly generated N individuals in the Ddimensional search space, which can be expressed by Eq. (2).

$$X_n = X_{min} + rand(0,1) \times (X_{max} - X_{min}) \tag{2}$$

Where X_{min} is the minimum value of the X bound, X_{max} is the maximum value of the X bound, and $rand(0,1)$ is a random number between 0 and 1.

2.1.2 Gradient search rule (GSR)

The gradient search rule (GSR) is extracted from the method based on Newton's gradient, which is used to control the exploration of the vectors in the search space and improve the convergence speed of GBO. The specific GSR equation is as follows

$$GSR = randn \times \rho_1 \times \frac{2\Delta x \times x_n}{(x_{worst} - x_{best} + \varepsilon)} \tag{3}$$

Where *randn* is a normally distributed random number, x_{worst} and x_{best} are the worst and best solutions in the optimization process, Δx is the value of delta, and ε is a random number between 0 and 0.1. In order to further improve the search ability of GBO, a random parameter ρ_1 is introduced into GSR; the specific expression is as follows.

2.1.3 Local escaping operator (LEO)

The purpose of LEO is to add the possibility of GBO escaping the local optimum. The solution to X_n^{m+1} is updated by the common effect of the best solution x_{best}, the solution $X1_n^m$ and $X2_n^m$, the interaction of two random solution x_{r1}^m and x_{r2}^m, and a random gained solution x_k^m. The execution of GSR meets certain conditions. The definition of pr decides the execution probability of GSR. The equation of LEO is expressed as follows.

$$\begin{gathered} X_n^{m+1} = X_n^{m+1} + f_1 \times (u_1 \times x_{best} - u_2 \times x_k^m) \\ +f_2 \times \rho_1 \times (u_3 \times (X2_n^m - X1_n^m) + u_2 \times (x_{r1}^m - x_{r2}^m))/2 \quad \text{rand} < 0.5 \end{gathered} \tag{4}$$

$$\begin{gathered} X_n^{m+1} = x_{best} + f_1 \times (u_1 \times x_{best} - u_2 \times x_k^m) \\ +f_2 \times \rho_1 \times (u_3 \times (X2_n^m - X1_n^m) + u_2 \times (x_{r1}^m - x_{r2}^m))/2 \quad \text{rand} \geq 0.5 \end{gathered} \tag{5}$$

where f_1 is a random number between -1 and 1, f_2 is a random from a normal distribution, and u_1, and u_3 and u_1, u_2 and u_3 are three random number, which is expressed as follows.

2.2 Proposed RWGBO

2.2.1 An adaptive weight factor

Any stochastic process can give us more chance to reach higher reliability and stability for the searching process due to its random spreading patterns [4]. In order to further improve the ability of the original GBO to get rid of local search, the adaptive weight mechanism is designed to improve the performance of GBO, which increase the possibility of getting rid of the local optimum. At the same time, this method is introduced into the GBO algorithm to achieve a proper balance between exploration and exploitation. in this paper, we introduce a key weight w which acts on new position. It is helpful for the algorithm to expand global search area in the early stage and avoid falling into a local optimum prematurely.

Among them, FES is the current calculation amount and MaxIt is the max calculation amount respectively. All the experiments in this paper, MaxIt is 300,000. The variables are changing continuously. When the optimal solution position of the population is not updated, the value is increased by 1 automatically. Once the Xnew can’t jump out in the local scope, it is easy to cause the algorithm unable to carry out a larger global search, unable to find a better solution.

Many researchers began to introduce adaptive weights into the optimization algorithms to obtain good results. Such as, the pr in this paper is equal to 0.5, and we introduce a crucial weight pr which acts on LEO. If rand is smaller than pr, the LEO can be defined by a equation. If rand is bigger than pr, the LEO will be defined by another expression. The mathematical formula of pr is expressed as follows.

$$\text{pr} = \text{FES}/\text{MaxIt} \tag{6}$$

$$\text{Pr} = w \tag{7}$$

FES is the current evaluation and MaxIt is the max evaluation number respectively. All the experiments in this paper, MaxIt is 300,000. While escaping the local optimum, the variables are continuously changing.

2.2.2 Random spare strategy

The random spare strategy is to replace the position vector of the current individual in the n-th dimension with the value of the position vector of the optimal individual in the current dimension. In the process of searching, the original algorithm may have appropriate position vectors in some dimensions, while there are inappropriate position vectors in others. However, the position vector on the optimal individual dimension is excellent. Therefore, we propose a random spare strategy to reduce the occurrence of the above situation. Since not every position vector on the individual dimension is bad, this strategy should be executed with some probability. From the starting value M to the end of the iteration, m is set to 0, the iteration result is the best. Finally, we choose the Cauchy random number to compare with the ratio of the current number of evaluations to the total number of evaluations to decide whether to implement the random standby strategy in the global scope.

2.2.3 The structure of the proposed RWGBO

In any design, it is very significant how to arrange the architecture of the skeleton and body of the model to reach the best quality of results [5]. Considering that the original GBO still has the possibility of falling into the local optimum, adaptive weight *pr* and random spare strategy are introduced into GBO for better convergence precision. The pseudo-code execution process of RWGBO is shown as follows, and the flowchart of RWGBO is shown in Fig.1.

Algorithm: RWGBO

Initialization parameters ε, *m*, *FES*, *MaxIt*, *nP*, *nV* and the initial population $X_0 = [x_{0,1}, x_{0,2}, \ldots, x_{0,D}]$;

Evaluate the objective function $f(x_0), n = 1,2,\ldots,N$, specify the best and worst solution x_{best}^m, x_{worst}^m;

FES=*FES*+*nP*;

While *FES*≤*MaxIt*

for each search agent

if (tan(pi∗(rand-0.5))<(1-FEs/MaxFEs))

Replace the current search location with the optimal search location

end if

end for

Calculate the fitness of each search agent and update x_{best}^m

Set the parameter of α , β and pr;

for *i*=1:*nP*

pr=*FES/MaxIt;*

Gradient search rule (GSR)

if *rand* <pr;

Local escaping operator (LEO)

Further calculate the position x_n^{m+1} by Eqs. (18)-(24)

end if

Evaluate the position x_n^{m+1} and Update the position x_{best}^m and x_{worst}^m;

FES=*FES*+1;

end for

m=*m*+1

End while

Return x_{best}^m

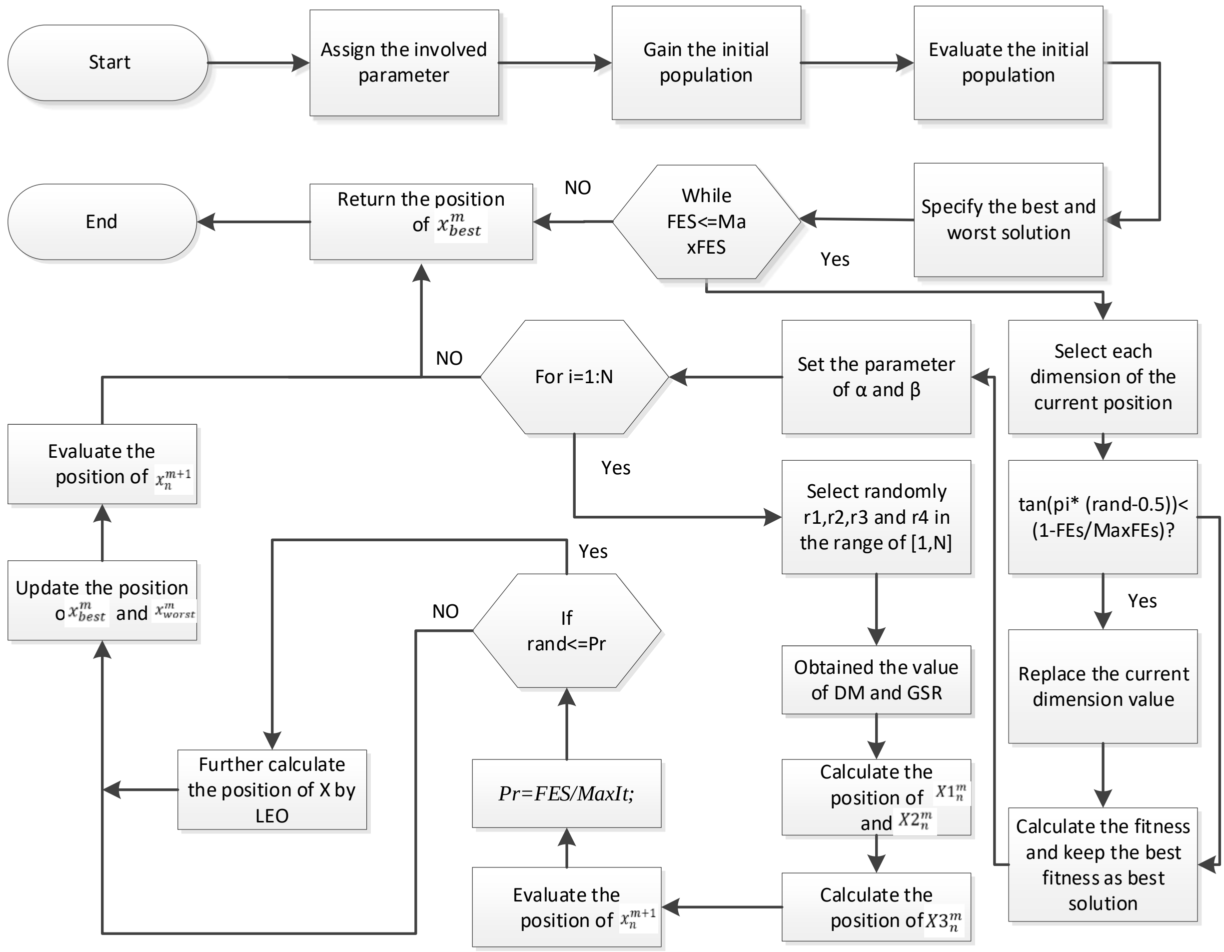

Fig. 1 The flowchart of RWGBO

3. Experimental results and analysis

The effectiveness of proposed RWGBO was verified on IEEE CEC 2017 test functions. Several experiments are performed to compare the proposed RWGBO algorithm and its competitors in the same test environment.

3.1 Component analysis

In this experiment, we used the test set from IEEE CEC 2017 test functions, where the search range is [-100,100]. In order to ensure the fairness and control small error of the experimental results, these comparison algorithms are run independently for 30 times, and the population size is set to 30. The value of the maximum number of evaluations (MaxIt) is set to 300 000, and the dimension of the search space is set to 30. The Friedman test is a comparative test of non-parametric statistical test, which is used to find the difference between multiple test results. Besides, the Friedman test is utilized to evaluate the results of all algorithms on the benchmark functions and give their rankings. In the end, the Friedman test will rank the average performance of all the selected methods for further statistical comparison and report the ARV (average ranking value) in the comparison results. The algorithm with a smaller ARV will have a better performance.

RGBO stands for the original algorithm containing a random selection strategy. Table 2 shows the P values of the Wilcoxon test [6] obtained for RWGBO, RGBO and GBO on 23 functions selected from IEEE CEC 2017 test functions. When the p-value is higher than 0.05, there is no significant difference between the two methods. When the p-value is less than 0.05, it implies that the two methods are significantly different.

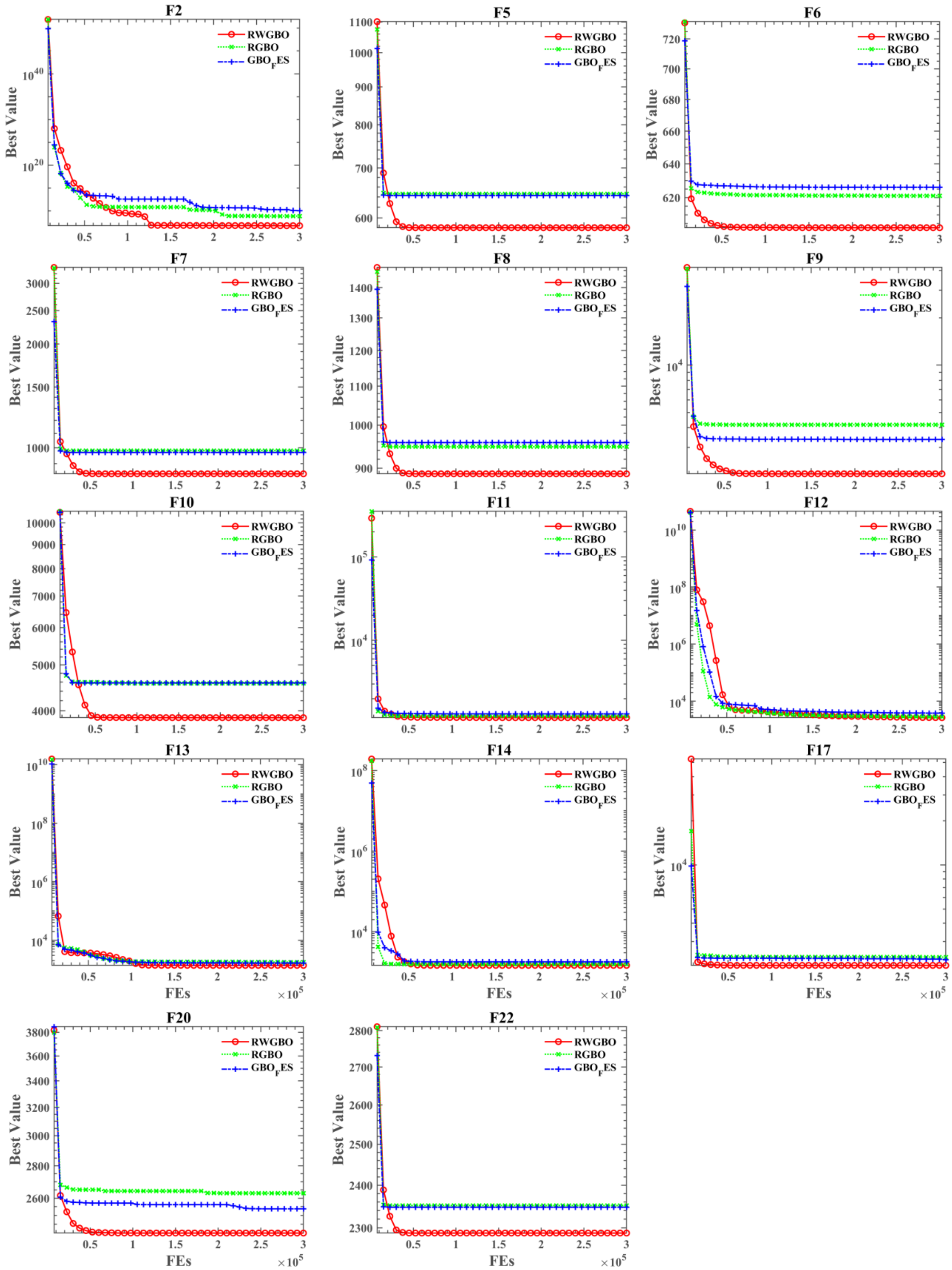

Fig. 2 Convergence curves on selected test functions

As can be seen from the p-value of the experimental results, the performance of RWGBO is the best among them. According to the average ranking value (ARV) obtained by the 23 benchmark functions selected from IEEE CEC 2017 test functions, RWGBO achieves the minimum average ranking value of 1.502899. Obviously, according to these ranking results, It can be found that RWGBO has higher precision than RGBO and GBO in Unimodal functions, Multimodal functions, hybrid functions and composite functions. RWGBO has higher precision than GBO in multi-module benchmark functions of F5, F6, F7, F8, F9, F10, Which can be seen in the Fig 2. Furthermore, RWGBO has higher precision than GBO in Hybrid Function benchmark functions, Which can be seen in the F11, F12, F13, F14, F17, F20. Therefore, Compared with GBO and RGBO it verifies that RWGBO has the best performance in resolving these test problems.

Table 2. P-values of CEC2017 test between RWGBO and GBO

	RWGBO	RGBO	GBO
F1	N/A	9.91E-01	4.11E-03
F2	N/A	1.41E-01	6.42E-03
F3	N/A	1.60E-04	7.52E-02
F4	N/A	9.75E-01	1.32E-02
F5	N/A	4.73E-064.73E-06	
F6	N/A	1.73E-06	1.73E-06
F7	N/A	2.35E-06	1.73E-06
F8	N/A	2.88E-06	1.92E-06
F9	N/A	1.97E-05	6.32E-05
F10	N/A	4.86E-05	1.29E-03
F11	N/A	1.60E-04	5.75E-06
F12	N/A	1.59E-01	4.45E-05
F13	N/A	2.13E-06	3.18E-06
F14	N/A	2.60E-06	1.73E-06
F15	N/A	6.98E-06	1.73E-06
F16	N/A	9.26E-01	2.62E-01
F17	N/A	2.60E-05	2.96E-03
F18	N/A	2.45E-01	4.41E-01
F19	N/A	4.17E-01	1.89E-04
F20	N/A	3.11E-05	7.73E-03
F21	N/A	7.81E-01	1.40E-02
F22	N/A	1.73E-06	1.36E-05
F23	N/A	1.00E+00	1.00E+00
+/-/=	N/A	13/1/917/2/4	
ARV	1.502899	2.16087	2.33632

References

[1] Goldfarb, D.; Qin, Z. Robust Low-Rank Tensor Recovery: Models and Algorithms. SIAM Journal on Matrix Analysis and Appli-cations.2014. 35(1). 225-253.

[2] Zhu, Q. Research on Road Traffic Situation Awareness System Based on Image Big Data. IEEE Intelligent Systems.2020. 35(1). 18-26.

[3] Mohamed H,H.; Essam H, H.; Mohamed A,M.; Salah, K. An improved Manta ray foraging optimizer for cost-effective emission dispatch problems. Engineering Applications of Artificial Intelligence.2021.100.104155.

[4] Dong, Q.; Cui, L.; Si, S. Reliability and availability analysis of stochastic degradation systems based on bivariate Wiener processes. Applied Mathematical Modelling. 2020. 79. 414-433.

[5] Ni, T. Architecture of Cobweb-Based Redundant TSV for Clustered Faults. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2020. 28(7). 1736-1739.

[6] Herrmann, D. Wilcoxon-test, Wahrscheinlichkeitsrechnung und Statistik — 30 BASIC-Programme, 1984, p. 51–52.