LSTM Neural Network with Parallel Swarm Optimization Algorithm for Multiple Regression Prediction

Jeng-Shyang Pan[1,2], Wei Wang[1], Shu-Chuan Chu[1], Zhi-Yuan Shao[1], Hong-Mei Yang[1]*

[1] *College of Computer Science and Engineering, Shandong University of Science and Technology, China*
[2] *Department of Information Management, Chaoyang University of Technology, Taiwan*
jspan@cc.kuas.edu.tw, weiwang@sdust.edu.cn, scchu0803@sdust.edu.cn, zhiyuanshao97@gmail.com, Yhm1998@163.com

Abstract

Multiple regression prediction can be applied to many applications, including predicting market prices, weather conditions, and so on. In recent years, the use of neural network models to deal with some problems related to multiple regression prediction has been proposed. The accuracy of prediction with long short-term memory (LSTM) neural network will be affected by the model parameters. If the parameters are adjusted according to human experience, there will be some limitations. In this paper, we propose a method to optimize multiple regression prediction of LSTM neural networks with a gannet optimization algorithm (GOA) modified by parallel communication strategies. The method optimizes the three parameters of the count of nodes in the hidden layer, training epochs, and learning rate of the LSTM neural network by the parallel gannet optimization algorithm (PGOA) to improve the accuracy and reliability of the prediction. The data results from the 28 benchmark functions tested by CEC2013 show that PGOA is more capable of finding the optimal solution compared to other algorithms. The accuracy of the PGOA-LSTM model and other models is tested with two datasets. The experimental results show that the PGOA-LSTM model predicts the data with higher accuracy than other models.

Keywords: Gannet optimization algorithm, Parallel, Communication strategy, Long short-term memory, Multiple regression prediction

1 Introduction

To better solve optimization problems in various application areas, metaheuristic algorithms have been proposed by simulating the evolutionary behavior of living organisms or based on the rules of physics. They usually solve complex problems in engineering sciences by imposing some requirements in the search process and are the product of combining stochastic algorithms with local search algorithms. Different metaheuristic algorithms use different processes to explore and exploit the search optimal space, and keep approaching the optimal solution by learning strategies. Many swarm intelligence optimization algorithms have been proposed earlier. Examples are: Particle swarm optimization (PSO) [1], cat swarm optimization (CSO) [2], ant colony optimization (ACO) [3], differential evolution (DE) [4], bat algorithm (BA) [5], whale optimization algorithm (WOA) [6], gray wolf optimization (GWO) [7] and other recently proposed efficient swarm optimization algorithms. Among them, the butterfly optimization algorithm (BOA) [8] was based on the foraging strategy of butterflies and used their sense of smell to locate nectar or mating partners in order to solve the global optimization problem. The fish migration algorithm (FMO) [9-10] was inspired by fish migration and integrated migration and swimming models into the optimization process to solve numerical optimization problems. The phasmatodea population evolution algorithm (PPE) [11] mimicked the features of convergent evolution, path dependence, population growth and competition in the evolutionary process of stick insect population in nature, so that stick insect population tend to be the nearest dominant population in the evolutionary process. These algorithms generally do not rely on information about the solution structure for optimization and can be applied to many different classes of combinations or functions. They combine stochastic algorithms with local search algorithms to obtain optimal solutions by using their own unique population evolution mechanisms that allow the population individuals to improve during the iterative process.

The Gannet Optimization Algorithm (GOA) [12] is more applicable than most swarm optimization algorithms to many constrained engineering design problems and provides better solutions in most cases. The GOA mathematizes the various predatory behaviors of gannet populations in their natural life, and uses the two diving and predatory behaviors of gannet for exploration and sudden turns and random walks for exploitation, which ensures that the optimal solution is found. Under the benchmark function test of CEC2013, the GOA is compared with other algorithms in experiments with the same number of iterations and population size conditions specified. Under the single-peaked function, GOA finds a relatively better solution at the beginning of the iteration and maintains this advantage until the final stage. And in some tests under the combinatorial function, even though GOA's convergence speed is not the fastest in the early stages, it converges very well and shows a great advantage in the later stages. The usability of GOA is better than other algorithms under time testing. So in this paper, GOA is used as an improved algorithm for application. However, when

*Corresponding Author: Shu-Chuan Chu; E-mail: scchu0803@sdust.edu.cn
DOI: https://doi.org/10.70003/160792642024122507008

GOA is used to optimize problems with complex conditions, it may fail to converge to the optimal solution or converge slowly. On the other hand, when it comes to solving some problems on a small or medium scale, GOA may find an effective solution late in the process, which takes more time. To this end, the GOA is improved by learning the parallel communication strategy [13-15], and the improved algorithm is called Parallel Gannet Optimization Algorithm (PGOA). Based on the summary of other studies, PGOA has a novel parallel grouping approach and two new communication strategies to compensate the shortcomings of GOA. Testing with 28 benchmark functions from CEC2013, we obtained the result that PGOA is more efficient in finding the optimal solution than GOA and other algorithms.

Multiple regression prediction is a method to establish a prediction model by analyzing the correlation between two or more influencing factors and a target outcome [16-17]. Compared to univariate regression prediction, it can use multiple influencing factors to predict the relevant data, thus allowing better accuracy and reliability of the predicted data [18-19]. Optimization methods with respect to mathematical classes [20-22] can be used to predict some complex and dynamic application fields, but with the progress of various fields and industries, the required influencing factors will gradually increase, leading to a large error in the prediction results [23-25]. Artificial intelligence technology is very hot in recent years, and researchers often use it to solve complex tasks that humans can't do. Compared with the classical technique of mathematical model, artificial intelligence technique is more effective in solving the problem of regression prediction [26-28]. Neural network model that can solve prediction problems has been much studied in artificial intelligence techniques [29-31], its application has been involved in various fields, and has achieved great success. Among them, LSTM neural network model has obvious advantages in the application of multiple regression prediction. Many studies [32-34] have used optimization algorithms to optimize LSTM neural networks. Optimization algorithms can, to some extent, optimize the weights, architectures, hyperparameters or other relevant coefficients in a neural network model [35-37]. Lu et al. [38] used the DE algorithm to optimize the LSTM to predict the price of electricity consumption. One of his studies shows that LSTM is better than earlier proposed mathematical methods in electricity price prediction because it can better handle some irregular changes in electricity prices. Chang et al. [39] used a hybrid method of adam and wavelet transform to optimize the LSTM, which has a more stable variance in predicting electricity prices and can accurately obtain the fluctuating changes of electricity prices. Pal et al. [40] uses a Bayesian optimization framework to optimize the shallow LSTM neural network, which can be applied to long-term forecasting. In terms of prediction results, the shallow network is better than the deep neural network. In general, all these methods can improve the accuracy of prediction models to some extent.

In this paper, we conclude from our experiments that PGOA has a stronger ability to find the best solution and it can better handle problems with different levels of complexity compared to other algorithms. This facilitates a more accurate optimization process for LSTM neural networks. Therefore, we propose a method to optimize multiple regression prediction of LSTM neural networks with a gannet optimization algorithm modified by a parallel communication strategy. The method optimizes the three parameters of the count of nodes in the hidden layer, training epochs and learning rate of the LSTM neural network by the parallel gannet optimization algorithm to improve the accuracy and reliability of the prediction. Two publicly available datasets are used to perform multiple regression prediction experiments on the proposed PGOA-LSTM model and are compared with other algorithmically optimized LSTM neural network models. The results show that the PGOA-LSTM model is more accurate in prediction compared to other models. In addition, the PGOA1-LSTM model with the first communication strategy is suitable for larger number of datasets, while the PGOA2-LSTM model with the second communication strategy is suitable for smaller number of datasets.

The structure of this paper is as follows: In Section 2, we introduce the GOA and LSTM neural networks. In Section 3, a parallel grouping strategy and two PGOAs with communication strategies are proposed. In Section 4, we conduct relevant experiments on the proposed PGOAs and analyze the experimental results. In Section 5, we construct the PGOA-LSTM model and test it against other models using two publicly available datasets. In Section 6, we provide a summary of the entire article and briefly describe the future work needed to be done.

2 Related Works

2.1 Gannet Optimization Algorithm

Gannets often live in groups in the wild, their eyes are very sharp, and very good at swimming and flying. Even as they fly through the air, they can spot fish in the water and hunt their targets with great speed. When it catches a fish, it flaps its wings quickly on the surface of the water, paddling its feet through the water. With a tremendous amount of thrust, the gannet gradually accelerates, and then, slowly reaching takeoff speed, it leaves the water and slowly rises into the air. The GOA algorithm was studied by the above mentioned gannets habits. GOA has both u and v shaped dive forms for the exploration phase and development has sudden turns and random walks.

During the exploration phase, gannets fly in the air to find prey targets in the water, observe the depth of the target from the water surface, and choose two diving modes according to the depth. The authors use Equation (2) for deep u shaped dives and Equation (3) for shallow v shaped dives,

$$t = 1 - \frac{it\_iter}{\max\_iter} \tag{1}$$

$$u = 2 \times \cos(2 \times \pi \times r1) \times t \tag{2}$$

$$v = 2 \times W(2 \times \pi \times r2) \times t \tag{3}$$

$$W(x)=\begin{cases}-\dfrac{1}{\pi}\times x+1, x\in(0,\pi)\\ \dfrac{1}{\pi}\times x-1, x\in(\pi,2\pi)\end{cases} \quad (4)$$

where *it_iter* represents how many iterations have been made, max_*iter* represents the largest amount of iterations, and r1, r2 are two random numbers in the range (0,1).

These two diving strategies are then used for position updates. The probability of choosing these two diving strategies is the same, so q is used to denote the random selection of a dive strategy, and q is a random number in the range (0,1). Define a storage matrix *MX*, and use MX_i instead of X_i if the current X_i is not as good as the individuals of the matrix MX_i after fitness function evaluation.

Among the population, $X_i(t)$ represents the ith individual, denote the randomly chosen individuals by $X_r(t)$, $X_m(t)$ is the mean position of the individuals, and $X_{Best}(t)$ represents the best individual so far. The position update formula is shown in Equation (5),

$$MX_i(t+1)=\begin{cases}X_i(t)+u1+u2, q\ge 0.5\\ X_i(t)+v1+v2, q<0.5\end{cases} \quad (5)$$

$$u2=U\times(X_i(t)-X_r(t)) \quad (6)$$

$$v2=V\times(X_i(t)-X_m(t)) \quad (7)$$

$$U=(2\times r3-1)\times u \quad (8)$$

$$V=(2\times r4-1)\times v \quad (9)$$

where r3 and r4 both range from a random number between 0 and 1, u1 ranges from -u and u, and v1 ranges from -v and v. Equation (10) represents the calculation of $X_m(t)$.

$$X_m(t)=\frac{1}{N}\sum_{i=1}^{N}X_i(t) \quad (10)$$

During the exploitation phase, when gannets encounter fish that suddenly turn around, they also need to take two actions to develop further. Here, capturing capability is defined as Equation (11),

$$Capturability=\frac{1}{R\times t2} \quad (11)$$

$$t2=1+\frac{it\_iter}{\max\_iter} \quad (12)$$

$$R=\frac{M\times v^2}{L} \quad (13)$$

$$L=1.8\times r5+0.2 \quad (14)$$

where r5 is a random number in the range (0,1), the gannet's weight is noted as M=2.5kg, and the gannet's marching speed in the water is v=1.5m/s (ignoring the resistance underwater). If the fish's escape position is within the gannet's ability to catch, the position will suddenly change to chase the fish; Otherwise, the gannets cannot catch the nimble fish and perform Levy moves to randomly find the next fish, referring to Equation (15),

$$MX_i(t+1)=\begin{cases}t\times delta\times(X_i(t)-X_{Best}(t))+X_i(t), Capturability\ge c\\ X_{Best}(t)-(X_i(t)-X_{Best}(t))\times P\times t, Capturability<c\end{cases} \quad (15)$$

$$delta=Capturability\times\left|X_i(t)-X_{Best}(t)\right| \quad (16)$$

$$P=Levy(Dim) \quad (17)$$

where c=0.2 is a specific value obtained from several surveys conducted by the author of the study GOA. Equation (18) is the *Levy*() flight function,

$$Levy(Dim)=\frac{\sigma\times\mu\times 0.01}{|v|^{\frac{1}{\beta}}} \quad (18)$$

$$\sigma=\left(\frac{\sin\left(\frac{\pi\beta}{2}\right)\times\Gamma(1+\beta)}{\Gamma\left(\frac{1+\beta}{2}\right)\times\beta\times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \quad (19)$$

where μ and σ correspond to random numbers in the range (0,1) and an already set constant of β =1.5.

The GOA algorithm not only saves execution time, but also has a great advantage in constraining engineering design problems. It can find the optimal solution better in some high dimensional cases. However, when GOA is used to optimize problems with complex conditions, the problem of not converging to the optimal solution or slow convergence may occur due to the presence of multiple parameters. In this paper, we propose a new grouping strategy that divides the entire population into smaller populations according to appropriate groupings, allowing them to explore and develop together. Each subpopulation has its own progress, which in turn improves the development of the whole population. It compensates the drawback that the original GOA is too slow to converge in some complex cases. With the use of parallel grouping for GOA, there is also a high probability that the optimal solution cannot be found. Therefore, it is necessary to add communication strategies to further optimize GOA. the proposed two PGOAs with communication strategies can greatly reduce the probability of falling into local optimum and effectively accelerate the convergence probability.

2.2 Long Short-term Memory Neural Networks

Long short-term memory (LSTM) [41] neural network is an improvement on recurrent neural networks (RNNs), which compensates the shortcoming of RNNs that cannot perform long-term dependence and effectively avoids the problem of gradient disappearance. Currently, it is commonly used for sequential tasks (including prediction problems, classification judgments, and machine translation), and it works much better than other models when dealing with large data sets. Its structure is shown in Figure 1.

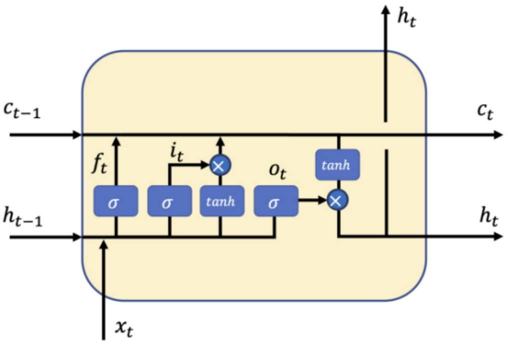


Figure 1. Basic structure of LSTM

Figure 1 shows the internal infrastructure of the LSTM, which operates on cell states through the structure of gates. The module in LSTM has three sigmoids and a tanh layer, which interact with each other in a special way. Firstly, LSTM should screen out which information needs to be forgotten, and this process is processed by the sigmoid layer of the forgetting gate. The forgetting gate reads the previous output h_{t-1} and the current input x_t, maps it to sigmoid, and prints a vector f_t (where the dimension value is a number between 0 and 1, where 1 is fully preserved and 0 is completely forgotten), multiplied by a cell state C_{t-1}. The next step is to process the new input information, use the input gate layer (sigmoid layer) to select the information to be updated, and then a tanh layer is used to produce a new $C_t^{'}$, which is added to the state, and then update the old cell state. The last thing is to determine the output value, starting with a sigmoid layer to identify which part of the state of the output cell is used. Then, the condition of the cell is manipulated with a tanh value between -1 and 1 and the output is multiplied with the output of the sigmoid gate to output the determined part.

3 Parallel Gannet Optimization Algorithm with Communication Strategy

In this section the optimal grouping arrangement of PGOA and its two different communication strategies are presented. By using parallel communication strategies, multiple populations with different strategies can more easily adapt to changes in the problem environment, improving the stability and robustness of the optimization process. It is equivalent to dividing the solution space into multiple subspaces and allowing individuals in each subspace to perform related activities. Each population searches different parts of the solution space independently and works together by sharing information. And it can focus on optimizing different objectives to improve the overall optimization performance and effectively enhance the ability of the algorithm to find the optimal solution.

3.1 Parallel Grouping Strategy

The parallel approach is to split the whole population in several small populations and let them explore and exploit together. Each small population makes its own progress, which in turn increases the level of development of the entire population [42-44]. With this approach, several individuals in multiple populations can search different regions of the solution space simultaneously, exploring the solution space more comprehensively and possibly finding better solutions. Multiple populations explored separately is equivalent to being able to perform multiple calculations separately, which allows faster convergence to the optimal solution, especially for complex optimization problems. Different grouping situations will give different results, and if the number of groups is too small or too large it will make exploration and exploitation less efficient, or even less efficient than before the grouping. In this paper, a new grouping strategy is proposed to select the appropriate number of groups. The number of population is defined as *Popsize*, and the number of divided groups is *Groupsize*, which can be obtained from the following equation:

$$Groupsize = \left\lceil \frac{Popsize * 0.8}{\sqrt{Popsize}} \right\rceil \quad \textbf{(20)}$$

3.2 Communication Strategies for Group Complementarity and Individual Learning Evolution

After using parallel grouping for GOA, convergence may still be too slow or the optimal solution cannot be found. Therefore, it is necessary to add communication policy to further optimize GOA. Communication strategy refers to the way in which individuals in a population communicate with each other, so that they can learn new information to improve their efficiency. Choosing an effective way can greatly reduce the probability of falling into the local optimum and effectively accelerate the probability of convergence.

In this section, two PGOAs with their own communication strategy are presented.

The group complementary communication strategy is to make each group have a mutual progress link, there is no extreme development situation between each group, so that each group can progress together. Each group will develop its own strengths to other groups, and through mutual learning, the strengths of the whole group can be gradually expanded. General implementation steps: Through a certain number of iterations, randomly select a set number of individuals to learn from the best individuals in each group (the number of groups divided is obtained by Equation (20)) or the best individuals in the whole group. If the number of Popsize is 20, the Groupsize calculated by Equation (20) is 4.

Among the population of the grouped, $X_i^g(t)$ is the i-th individual in group g, $X_{Best}^1(t)$ is the best individual in the 1st group, $X_{Best}^2(t)$ is the best individual in the 2nd group, and so on, $X_{Best}^{All}(t)$ is the best one among all groups. The mathematical form of the group complementary communication strategy is as follows:

$$X_i^g(t)=\begin{cases} X_i^g(t)+(X_{Best}^1(t)-X_i^g(t))\times r',r6\in[0.1/5] \\ X_i^g(t)+\left(\left(X_{Best}^1(t)+X_{Best}^2(t)\right)/2-X_i^g(t)\right)\times r',r6\in[1/5,2/5] \\ X_i^g(t)+\left(\left(X_{Best}^1(t)+X_{Best}^2(t)+X_{Best}^3(t)\right)/3-X_i^g(t)\right)\times r',r6\in[2/5,3/5] \\ X_i^g(t)+\left(\left(X_{Best}^1(t)+X_{Best}^2(t)+X_{Best}^3(t)+X_{Best}^4(t)\right)/4-X_i^g(t)\right)\times r',r6\in[3/5,4/5] \\ X_i^g(t)+\left(\left(X_{Best}^{All}(t)+X_i^g(t)\right)\right)\times r',r6\in[4/5,1] \end{cases} \quad \textbf{(21)}$$

where g is a positive integer in the interval [0,Groupsize], t is the current iteration, and r' and r6 are two random numbers in the range of (0,1).

The individual learning evolutionary communication strategy is for individuals in a group to learn from other good individuals to achieve self-improvement. Just as some people do not have a clear perception of their own level, they always think that their level is higher than others' and subconsciously ignore their shortcomings. One must learn how to learn the advantages of others so that one can continuously improve oneself.

The mathematical form of individual learning evolutionary communication strategy is as follows. r'' in Equation (22) is a random number in the range of (0,1).

$$X_i^g(t)=X_{Best}^{All}(t)\times(\ln(2.5)+\log_{10}1.5\times r'') \quad \textbf{(22)}$$

The above are the two PGOAs with different communication strategies, Figure 2 is a main process framework of PGOA. T is the count of current iterations, E is the maximum number of iterations, and R1 is the count of iterations set for the communication strategy to be performed. The pseudo-code for PGOA with two communication strategies is given in Algorithm 1. They will be referred to as PGOA1 and PGOA2 in the following.

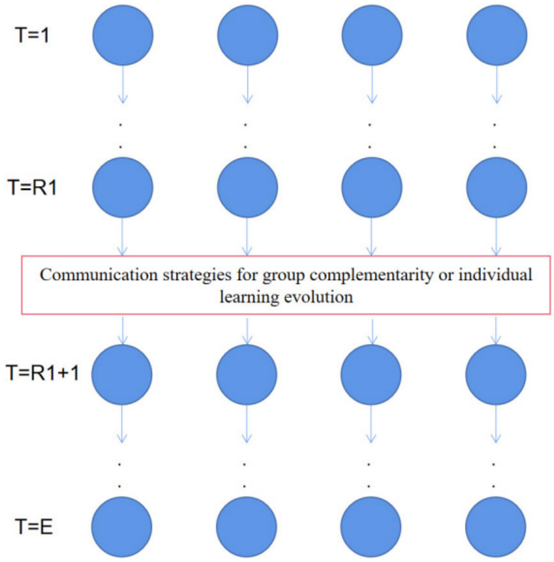


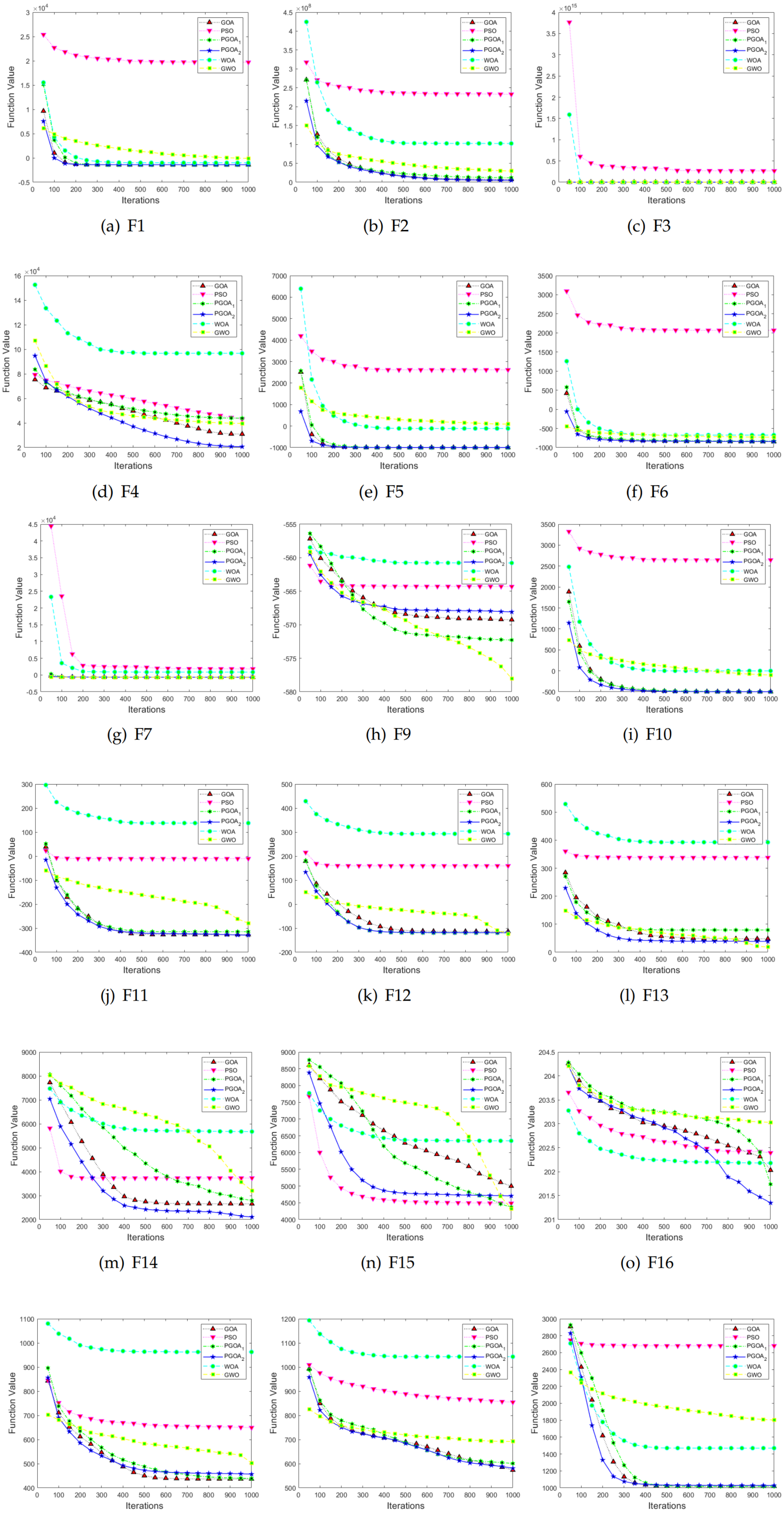
Figure 2. The main framework of PGOA

Algorithm 1. The pseudo code of PGOA

Input: The size of a population (N), number of iterations (max_$iter$), number of dimensions (Dim), number of iterations set for the communication strategy to be performed ($R1$).

Output: The location of Gannet and its fitness value.

Equation (20) is used to set the number of groups Groupsize, each group is $G(g)$, ($g \leq$ Groupsize);
Initialize the $G(g)$, X, $G(g).X_{Best}$, r and q are all random numbers from 0 to 1;
Generate memory matrix $G(g).MX$, calculate the fitness value of $G(g).X$;
for i = 1: max_iter **do**
  for g = 1: Groupsize **do**
    if $i = R1$ then
      The group complementary communication strategy: Use Equation (21) to update $G(g).X_i$;
      The individual learning evolutionary communication strategy: Use Equation (22) to update $G(g).X_i$;
    end if
    if $rand > 0.5$ **then**
      for $G(g).MX_i$ **do**
        Update the location Gannet with U-shaped or V-shaped movement using Equation (5);
      end for
    else
      for $G(g).MX_i$ **do**
        Update the location Gannet with sudden turning or Levy flight using Equation (15);
      end for
    end if
    for $G(g).MX_i$ **do**
      Calculate the fitness value of $G(g).MX_i$;
      If the value of $G(g).MX_i$ is better than the value of $G(g).X_i$, replace $G(g).X_i$ with $G(g).MX_i$;
    end for
  end for
end for



(a) F1 (b) F2 (c) F3

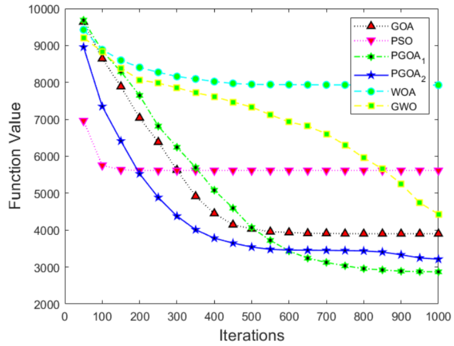
(d) F4 (e) F5 (f) F6

(g) F7 (h) F9 (i) F10

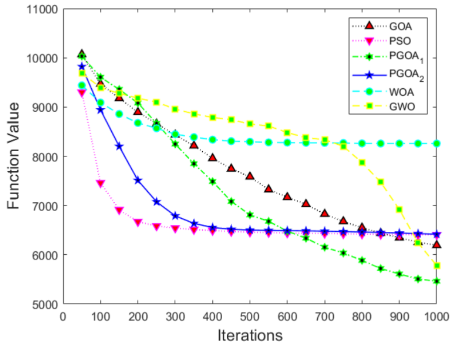
(j) F11 (k) F12 (l) F13

(m) F14 (n) F15 (o) F16

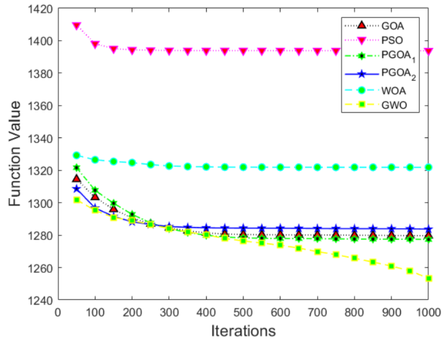
(p) F17 (q) F18 (r) F21



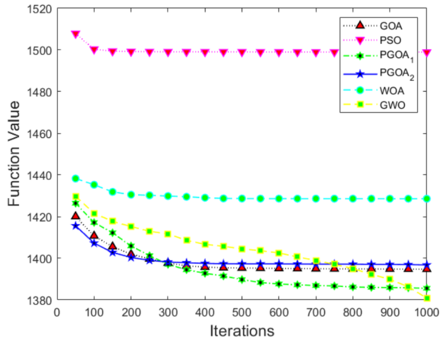
(s) F22



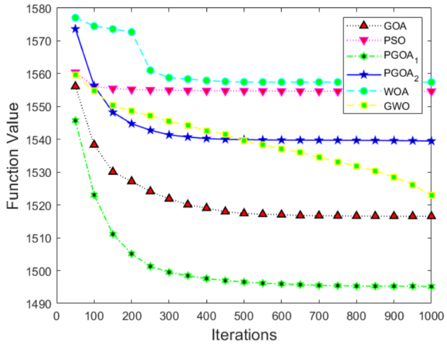
(t) F23



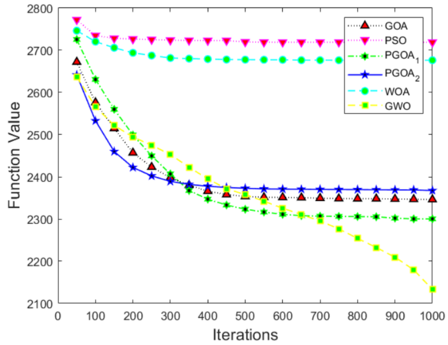
(u) F24



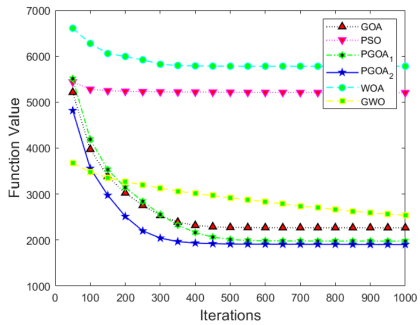
(v) F25



(w) F26



(x) F27



(y) F28

Figure 3. The convergence curve of the algorithm under the benchmark function test of CEC2013

4 Experimental Results and Analysis

In this section, we use functions from the CEC2013 testsets to test and compare the effectiveness of PGOA with other algorithms. Mainly, we compare their respective average fitness values and standard deviations, and further analyze them by iterative curves of fitness values.

4.1 Parameter Settings

All experiments in the paper were studied on a computer with Windows 10 Professional 64-bit, Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz and 16GB running memory. The algorithms that appear were run on MATLAB R2021a. Table 1 presents some parameter settings for the algorithms covered in this paper. To show the effectiveness of the PGOA algorithm, we keep the basic parameters of the GOA and the two modified PGOAs the same, which are the gannet's weight m, the speed vel in water vel, and the fixed parameter c obtained in the original algorithm GOA. In the parameter design of both PGOAs, we set the number of parallel groups Groupsize to 4, which is obtained using Equation (20) based on the number of individuals in the experiment. R is the number of iterations at which communication exchanges are performed, and its parameter setting is a value determined based on multiple experiments. Other algorithms use their own default parameter values.

Table 1. Parameter design of related algorithms

Algorithms	Parameter settings
GOA	vel=1.5, m=2.5, c=0.2
PGOA1	vel=1.5, m=2.5, c=0.2, Groupsize=4, R=40
PGOA2	vel=1.5, m=2.5, c=0.2, Groupsize=4, R=20
PSO	V_max=6, w=0.3, c1=c2=2
WOA	a: from 2 to 0 linearly decreasing, a2: from -1 to -2 linearly decreasing
GWO	a: Decreases linearly from 2 to 0

4.2 Experimental Analysis on CEC 2013

There are 28 different benchmark functions in CEC2013. Among them, F1-F5 are unimodal functions with a unique global optimization. F6-F20 are a multimodal function, which is more complex than the unimodal function and has multiple local optima. F21-F28 are combinatorial functions.

In the experiments the initial parameter values are the same for each algorithm. The number of individuals is 40, the dimension is 30, the maximum number of iterations is 1000, and the initial solution range was from -100 to 100 in order to put each algorithm under fair conditions for testing. We have tested these algorithms 30 times and Table 2 gives the execution time of each algorithm under the above conditions, where the GOA algorithm has the shortest execution time and the PGOA with two parallel communication strategies has a longer execution time compared to the other algorithms, but in the following it is concluded that their solutions are highly efficient. Table 3 shows the average results of these 30 tests. The last row of Table 3 shows the number of times PGOA1 and PGOA2 won compared to other algorithms such as GOA. Among them, they won 20 and 23 times each compared to GOA, 27 and 26 times each compared to PSO, 27 and 27 times each compared to WOA, and 19 and 20 times each compared to GWO. In general, it can be seen that these two PGOA algorithms with different parallel strategies perform better than GOA, PSO, WOA and GWO.

Table 2. Execution time of each algorithm

Algorithms	Execution time (unit: s)
GOA	767.481
PGOA1	1103.093
PGOA2	1021.518
PSO	785.611
WOA	813.728
GWO	801.619

In order that the effectiveness of the PGOA algorithm can be clearly observed, Figure 3 shows the convergence curves of each algorithm in the CEC 2013 test function. The X-axis is the count of iterations and the Y-axis indicates the corresponding adaptation degree. A marker is made on the curve every 50 generations, which makes it easier to study a convergence of the curve.

In the unimodal functions of F1-F5 in Figure 3, it can be seen that PGOA1 and PGOA2 have better convergence ability than other algorithms. Compared to the original algorithm GOA, they are better at finding the optimal solution at the beginning and have improved in convergence speed. Overall PGOA2 is also better than PGOA1. The effect is even more obvious in F4. Although the early stage of PGOA2 is not as good as that of GOA and PGOA2, the convergence speed of PGOA2 gradually accelerates as the iterative count grows, and the convergence ability in the later stage is stronger compared with other algorithms.

In the multimode functions of F6-F18 in Figure 3. The convergence ability of PGOA2 is still stronger than the other algorithms in most cases. The efficiency of the search on the F6, F10, F14 and F16 is better demonstrated, and there is no premature convergence. As the complexity of the function increases, the advantage of PGOA1 is gradually revealed, and the effect of finding the optimal solution and convergence speed is not much different from that of PGOA2. In F9, it is not as good as the convergence ability of GWO, but it is better than GOA, PGOA2 and other algorithms. In F14 and F15, although the convergence ability in the beginning is inferior to PSO, GOA and PGOA2, its convergence curve has not leveled off as the growing count of iterations, indicating that it has a strong ability to find the best in the later stage.

In the combinatorial functions of F21-F28 in Figure 3, PGOA1 highlights its advantages. In F22 and F23, although it converges more slowly in the early stage, it can be seen in the later stage that it does not end the search for the best solution, indicating that its convergence ability will continue to improve with further increase in the number of iterations. In F26, it finds a well-solved way in the initial stage without the problem of getting stuck in a local optimum.

According to the above analysis, the proposed PGOA1 algorithm has an advantage in its ability to explore at a later stage and converges better in some cases with complex function tests or in combinatorial functions. Compared with other algorithms, the PGOA2 algorithm converges faster in the case of unimodal functions or multimode functions, has better exploration ability in the early stage. Overall, the improved two PGOA algorithms are largely improved compared to the original algorithm.

Table 3. Results of each algorithm tested under CEC 2013 benchmark function

Function	GOA		PGOA1		PGOA2		PSO		WOA		GWO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F1	-1.40E+03	1.59E-04	-1.40E+03	3.53E-05	-1.40E+03	3.89E-03	2.03E+04	3.41E+03	-8.80E+02	3.31E+02	3.38E+02	1.49E+03
F2	7.27E+06	3.15E+06	1.16E+07	3.60E+06	5.21E+06	1.87E+06	2.32E+08	1.16E+08	9.92E+07	3.76E+07	3.22E+07	1.56E+07
F3	2.62E+09	2.29E+09	2.04E+09	1.52E+09	1.01E+09	9.74E+08	1.53E+14	5.86E+14	3.54E+10	1.52E+10	6.18E+09	4.56E+09
F4	2.97E+04	8.47E+03	3.99E+04	8.49E+03	1.97E+04	6.30E+03	4.30E+04	9.41E+03	9.11E+04	3.34E+04	4.17E+04	5.91E+03
F5	-1.00E+03	5.21E-03	-1.00E+03	3.99E-03	-1.00E+03	1.23E-02	2.99E+03	1.95E+03	-2.18E+02	2.23E+02	-7.40E+01	4.08E+02
F6	-8.37E+02	2.49E+01	-8.32E+02	2.94E+01	-8.52E+02	2.54E+01	2.18E+03	1.29E+03	-6.27E+02	8.87E+01	-7.32E+02	4.91E+01
F7	-6.66E+02	4.61E+01	-6.81E+02	3.05E+01	-6.72E+02	3.78E+01	5.36E+03	1.79E+04	4.60E+03	1.34E+04	-7.24E+02	1.85E+01
F8	-6.79E+02	5.03E-02	-6.79E+02	6.62E-02	-6.79E+02	6.11E-02	-6.79E+02	5.81E-02	-6.79E+02	5.09E-02	-6.79E+02	3.75E-02
F9	-5.68E+02	5.25E+00	-5.72E+02	3.33E+00	-5.69E+02	3.49E+00	-5.65E+02	3.61E+00	-5.61E+02	2.67E+00	-5.78E+02	3.15E+00
F10	-4.99E+02	4.94E-01	-4.97E+02	1.90E+00	-4.99E+02	3.09E-01	2.73E+03	8.18E+02	-6.15E+01	1.22E+02	-1.16E+02	2.10E+02
F11	-3.13E+02	2.33E+01	-3.15E+02	2.08E+01	-3.22E+02	2.50E+01	2.87E+00	5.42E+01	1.58E+02	1.30E+02	-2.80E+02	3.33E+01
F12	-1.18E+02	5.41E+01	-1.19E+02	5.06E+01	-1.20E+02	6.74E+01	1.62E+02	6.62E+01	2.57E+02	1.00E+02	-1.26E+02	6.90E+01
F13	5.15E+01	4.73E+01	7.82E+01	5.97E+01	4.69E+01	5.86E+01	3.59E+02	7.52E+01	3.70E+02	9.34E+01	2.47E+01	4.16E+01
F14	2.93E+03	6.31E+02	2.55E+03	9.08E+02	2.45E+03	4.94E+02	3.70E+03	5.42E+02	5.89E+03	7.56E+02	3.91E+03	1.67E+03
F15	5.02E+03	8.10E+02	4.36E+03	7.54E+02	4.67E+03	7.61E+02	4.57E+03	6.84E+02	6.27E+03	1.04E+03	4.55E+03	1.99E+03
F16	2.02E+02	3.95E-01	2.02E+02	6.29E-01	2.01E+02	4.49E-01	2.02E+02	6.37E-01	2.02E+02	4.05E-01	2.03E+02	5.36E-01
F17	4.43E+02	4.35E+01	4.46E+02	3.22E+01	4.56E+02	3.39E+01	6.39E+02	6.55E+01	9.50E+02	1.02E+02	5.11E+02	5.12E+01
F18	5.88E+02	3.70E+01	6.03E+02	4.69E+01	5.75E+02	3.84E+01	8.25E+02	6.74E+01	1.06E+03	9.77E+01	6.84E+02	2.88E+01
F19	5.10E+02	2.80E+00	5.09E+02	3.01E+00	5.08E+02	2.21E+00	3.39E+04	2.84E+04	6.34E+02	6.01E+01	7.19E+02	3.62E+02
F20	6.14E+02	1.04E+00	6.14E+02	1.24E+00	6.14E+02	1.03E+00	6.15E+02	2.00E-01	6.15E+02	2.20E-01	6.14E+02	1.32E+00
F21	1.03E+03	8.90E+01	1.02E+03	8.60E+01	1.01E+03	9.84E+01	2.69E+03	1.19E+02	1.44E+03	3.22E+02	1.83E+03	4.04E+02
F22	3.65E+03	5.82E+02	3.17E+03	7.14E+02	3.22E+03	6.79E+02	5.89E+03	1.08E+03	7.98E+03	9.70E+02	4.56E+03	9.82E+02
F23	6.36E+03	8.98E+02	5.38E+03	7.06E+02	6.09E+03	1.03E+03	6.54E+03	8.56E+02	7.95E+03	8.65E+02	6.21E+03	1.82E+03
F24	1.28E+03	9.74E+00	1.28E+03	6.83E+00	1.28E+03	6.26E+00	1.39E+03	3.44E+01	1.32E+03	9.60E+00	1.26E+03	1.12E+01
F25	1.40E+03	9.95E+00	1.39E+03	8.08E+00	1.40E+03	1.04E+01	1.50E+03	2.12E+01	1.43E+03	1.08E+01	1.38E+03	8.51E+00
F26	1.54E+03	7.02E+01	1.51E+03	7.56E+01	1.54E+03	7.10E+01	1.55E+03	8.03E+01	1.58E+03	6.08E+01	1.51E+03	6.81E+01
F27	2.35E+03	9.22E+01	2.34E+03	8.29E+01	2.38E+03	7.50E+01	2.69E+03	1.26E+02	2.68E+03	7.81E+01	2.18E+03	7.74E+01
F28	2.13E+03	9.65E+02	2.53E+03	1.19E+03	1.89E+03	6.10E+02	5.35E+03	6.26E+02	6.09E+03	9.56E+02	2.80E+03	5.28E+02
Win (PGOA1)	20		-		-		27		27		19	
Win (PGOA2)	23						26		27		20	

5 Multiple Regression Prediction based on PGOA-LSTM

Multiple regression prediction plays an important role in many fields. It can use multiple factors to predict the relevant data. The prediction model optimized by LSTM neural network can make the prediction results achieve better accuracy and reliability. However, with the increase of influencing factors, its prediction error will be large. How to improve its prediction accuracy is a problem that people have studied in recent years. In this section, we analyze the proposed new model with the results predicted by other methods, and the results show that it has a high accuracy in multiple regression prediction.

5.1 PGOA-LSTM Prediction Model

When using LSTM neural networks to deal with prediction problems, the effectiveness with default parameters is often poor [45-46]. Therefore, how to set the initial parameter values is most important, and a good setting scheme can effectively enhance the prediction capability of LSTM neural networks [47-48].

The PGOA algorithm proposed in this paper is used to find the three best hyperparameters in the LSTM neural network, which are the count of nodes in the hidden layer, the training epochs and the learning rate. The learning rate setting directly affects the learning speed of the neural network, if the learning rate is too low, it will lead to slow convergence, and if it is too high, it will lead to the local optimum problem and cannot converge to the optimal value. The count of nodes in the hidden layer and the setting of the training epochs will affect the efficiency of the prediction model. The optimized three hyperparameters are used to train the model.

The implementation steps of the PGOA-LSTM model are as follows:

Step 1: Divide the data set into a training set and a test set in the ratio of 4:1. Data normalization is performed on them.

Step 2: Initialize the amount of populations, dimensions and maximum iterations. Set a hyperparameter range matrix and group the population.

Step 3: Initialize the positions of the individuals in each group according to the range set by Step 2. The best individual position, the optimal individual fitness value is initialized.

Step 4: Follow the PGOA algorithm proposed above to find the optimal individual. Whenever the position is updated, the current $X_i(t)$ is substituted into the fitness function. In the fitness function, a basic LSTM model is defined, where the number of hidden nodes, the maximum number of iterations and the learning rate are the three parameters found by the PGOA algorithm in the optimal search process. Based on the values of these three optimized parameters, the mean square error (MSE) of the prediction results is calculated as the fitness value after inputting the normalized data set and making predictions. Based on the values returned by the fitness function, the fitness values of each group of individuals are compared and the best $X_{Best}^{All}(t)$ is found among all individuals until the end of the iteration.

Step 5: The three optimal parameter values obtained in Step 4 are taken to create a model with the best prediction results.

5.2 Information about the Datasets

Two datasets are used in this paper, one is Electricity Load Database and the other is Real Estate Market Database. In the electricity load database we selected 625 data of electricity consumption in June 2020. This includes 12 feature variables for temperature, humidity, liquid deposition and wind speed for three cities. In the real estate market database we selected 2000 data to be used for the prediction experiment. This includes 12 feature variables such as number of bedrooms, number of bathrooms, and number of squares. In order to better observe the changes in house prices, we turn the house prices in the dataset into a feature variable measured in thousands. They are both datasets that predict a target outcome based on multiple influencing factors.

5.3 Model Parameter Settings and Evaluation Metrics

In this paper, a single-layer LSTM neural network structure is used, and the solver is set to 'adam'. To avoid the problem of gradient explosion, the gradient threshold is set to 1 here, and the other parameters are default values except for the hyperparameters to be optimized.

Table 4. Basic parameter setting of the optimization algorithms

Parameter name	Parameter value
Pop_size	20
Dim	3
Max_iter	200
HiddenUnits	[10,200]
MaxEpochs (Electricity Load Database)	[20,50]
MaxEpochs (Real Estate Market Database)	[30,100]
InitialLearnRate	[0.001,0.05]

For the algorithms involved in the optimization performed by the LSTM neural network, the settings of their basic parameter values are shown in Table 4. To ensure fairness and better comparison of experimental results, we set the necessary parameters needed in each algorithm to be the same. Only the maximum number of iterations in the hyperparameters is divided into ranges according to the size of the dataset. In the experiments in this paper, the size of the dataset of Electricity Load Database is smaller than that of Real Estate Market Database, so the range of its maximum number of iterations is set one level smaller.

We use the proposed PGOA-LSTM model as well as other models to perform a multiple regression prediction on the above two datasets. The following are the model accuracy assessment metrics that we use.

MAE (Mean Absolute Error) is mostly used to test the average difference of predicted data results from the actual data results.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{y_i} - y_i\right| \tag{23}$$

RMSE (Root Mean Square Error) measures the deviation of the predicted data results from the true data results.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{y_i} - y_i\right)^2} \tag{24}$$

R^2 is the coefficient of determination and takes a value in the range of $(-\infty, 1]$, with larger values indicating more accurate predicted data results.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}{\sum_{i=1}^{n}(\overline{y_i} - y_i)^2} \tag{25}$$

5.4 Experimental Results and Analysis of Multiple Regression Prediction

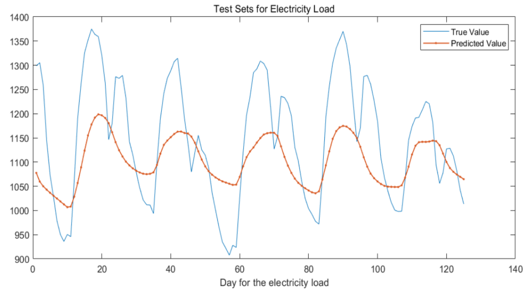
We used PGOA and the other mentioned optimization algorithms to find the most suitable hyperparameters for the LSTM neural network to make predictions. Then experiments were conducted on the LSTM neural networks optimized by the different algorithms using the two datasets separately.

Figure 4 and Figure 5 the prediction comparison graphs of the LSTM models optimized with each algorithm for the electricity load database and the real estate market database, respectively. After observation, it can be concluded that the two models, PGOA1-LSTM and PGOA2-LSTM, have the best fit with the real data results.

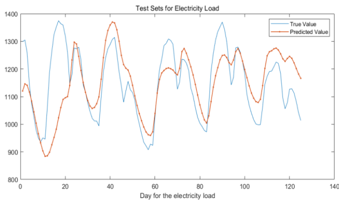
Table 5 shows that the mean errors of the prediction results of the two models PGOA1-LSTM and PGOA2-LSTM are small, and they are approximately the same as the mean absolute errors of the GOA model. Among all the prediction models, PGOA2-LSTM has the highest coefficient of determination, and its prediction accuracy is higher compared to the other models.

In Table 6, several evaluation metrics of the PGOA1-LSTM model are excellent. It also has the most excellent coefficient of determination compared with other models. Overall, the PGOA1-LSTM model has better predictive power.

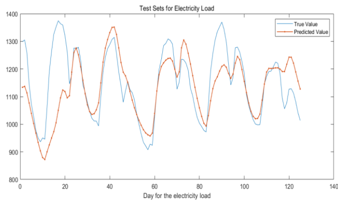
In summary, we used the PGOA2-LSTM model for multiple regression prediction of smaller datasets and the PGOA1-LSTM of larger datasets.



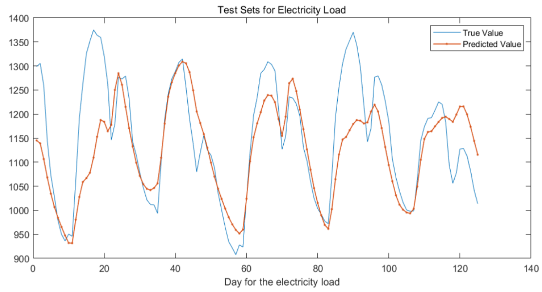
(a) LSTM prediction



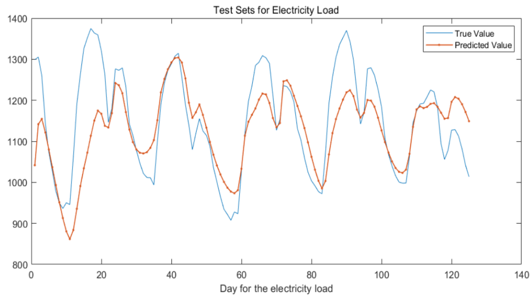
(b) PSO-LSTM prediction



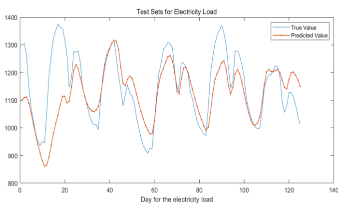
(c) GOA-LSTM prediction



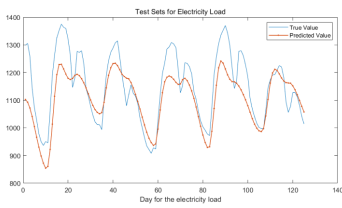
(d) PGOA$_1$-LSTM prediction



(e) PGOA$_2$-LSTM prediction



(f) WOA-LSTM prediction



(g) GWO-LSTM prediction

Figure 4. Comparison graph of true and predicted values from the Electricity Load Database

(a) LSTM prediction

(b) PSO-LSTM prediction

(c) GOA-LSTM prediction

(d) PGOA$_1$-LSTM prediction

(e) PGOA$_2$-LSTM prediction

(f) WOA-LSTM prediction

(g) GWO-LSTM prediction

Figure 5. Comparison graph of true and predicted values from the Real Estate Market Database

Table 5. Prediction errors and hyperparameters of different models (Electricity Load Database)

Models	MAE	RMSE	R^2	HiddenUnits	InitialLearnRate	MaxEpochs
LSTM	90.089	110.449	0.205	100	0.0010	30
PSO-LSTM	67.471	101.288	0.331	154	0.0063	35
GOA-LSTM	74.198	99.355	0.356	98	0.0023	26
PGOA1-LSTM	69.221	92.146	0.446	81	0.0078	32
PGOA2-LSTM	57.066	85.088	0.528	120	0.0210	43
WOA-LSTM	71.644	105.381	0.276	68	0.0097	41
GWO-LSTM	72.704	103.957	0.295	24	0.0038	24

Table 6. Prediction errors and hyperparameters of different models (Real Estate Market Database)

Models	MAE	RMSE	R^2	HiddenUnits	InitialLearnRate	MaxEpochs
LSTM	17.015	25.665	0.208	100	0.0010	50
PSO-LSTM	16.368	23.185	0.354	91	0.0021	52
GOA-LSTM	15.820	22.793	0.376	106	0.0276	47
PGOA1-LSTM	13.783	19.869	0.526	131	0.0041	87
PGOA2-LSTM	15.631	21.102	0.464	121	0.0049	67
WOA-LSTM	16.867	23.268	0.349	84	0.0018	36
GWO-LSTM	16.253	23.087	0.361	108	0.0176	42

6 Conclusion

In this paper, we add a parallel approach to the GOA and propose a grouping strategy and two PGOA communication strategies. Under the test of 28 benchmark functions, we compare the proposed two PGOAs with different communication strategies with other algorithms, and the experimental results show that the PGOA1 with the first communication strategy has an advantage in the late exploration ability and converges better in the case of some complex function tests or combined functions. The PGOA2 algorithm converges faster in the case of unimodal functions or multimodal functions, has better exploration ability in the early stage, and does not have the problem of falling into local optimum. Therefore, PGOA has better efficiency and stability. By combining with LSTM neural network, we obtain a PGOA-LSTM model for multiple regression prediction. The PGOA goes to optimize the three hyperparameters in the LSTM model, which are the count of nodes in the hidden layer, the training epochs and the learning rate. The accuracy of the PGOA-LSTM model was tested with two datasets. The results obtained were that the other models were not as accurate as the PGOA-LSTM model. Among them, the PGOA1-LSTM model predicts more accurately in larger datasets, while the PGOA2-LSTM model predicts more accurately in smaller datasets.

Although the proposed model has good results, further improvements are needed. For example, other parameters and architectures of the LSTM can affect the accuracy of prediction, and we only optimized three hyperparameters. In future work, we would research more valid communication strategies to enhance the efficiency of PGOA and optimize other parameters and architectures in the LSTM model.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95-international conference on neural networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[2] S. C. Chu, P. W. Tsai, J. S. Pan, Cat swarm optimization, *Pacific Rim international conference on artificial intelligence*, Guilin, China, 2006, pp. 854-858.

[3] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE computational intelligence magazine*, Vol. 1, No. 4, pp. 28-39, November, 2006.

[4] K. V. Price, Differential Evolution, in: I. Zelinka, V. Snášel, A. Abraham (Eds.), *Handbook of optimization*, Springer, 2013, pp. 187-214.

[5] X. S. Yang, X. He, Bat algorithm: literature review and applications, *International Journal of Bio-inspired computation*, Vol. 5, No. 3, pp. 141-149, July, 2013.

[6] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Advances in engineering software*, Vol. 95, pp. 51-67, May, 2016.

[7] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in engineering software*, Vol. 69, pp. 46-61, March, 2014.

[8] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, *Soft Computing*, Vol. 23, No. 3, pp. 715-734, February, 2019.

[9] J. S. Pan, P. Hu, S. C. Chu, Binary fish migration optimization for solving unit commitment, *Energy*, Vol. 226, No. 120329, July, 2021.

[10] B. Guo, Z. Zhuang, J. S. Pan, S. C. Chu, Optimal design and simulation for PID controller using fractional-order fish migration optimization algorithm, *IEEE Access*, Vol. 9, pp. 8808-8819, January, 2021.

[11] P. C. Song, S. C. Chu, J. S. Pan, H. Yang, Simplified Phasmatodea population evolution algorithm for optimization, *Complex & Intelligent Systems*, Vol. 8, No. 4, pp. 2749-2767, August, 2022.

[12] J. S. Pan, L. G. Zhang, R. B. Wang, V. Snášel, S. C. Chu, Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems, *Mathematics and Computers in Simulation*, Vol. 202, pp. 343-373, December, 2022.

[13] Z. Y. Shao, J. S. Pan, P. Hu, S. C. Chu, Equilibrium optimizer of interswarm interactive learning strategy, *Enterprise Information Systems*, Vol. 17, No. 1, pp. 1-25, 2023.

[14] J.-F. Chang, S. C. Chu, J. F. Roddick, J. S. Pan, A parallel particle swarm optimization algorithm with communication strategies, *Journal of information science and engineering*, Vol. 21, No. 4, pp. 809-818, July, 2005.

[15] S. C. Chu, X. W. Xu, S. Y. Yang, J. S. Pan, Parallel fish migration optimization with compact technology based on memory principle for wireless sensor networks, *Knowledge-Based Systems*, Vol. 241, Article No. 108124, April, 2022.

[16] M. H. Licht, Multiple regression and correlation, in: L. G. Grimm, P. R. Yarnold (Eds.), *Reading and understanding multivariate statistics*, American Psychological Association, 1995, pp. 19-64.

[17] G. T. Knofczynski, D. Mundfrom, Sample sizes when using multiple linear regression for prediction, *Educational and psychological measurement*, Vol. 68, No. 3, pp. 431-442, June, 2008.

[18] D. Zhang, D. Shen, Alzheimer's Disease Neuroimaging Initiative, Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease, *NeuroImage*, Vol. 59, No. 2, pp. 895-907, January, 2012.

[19] C. Heim, D. J. Newport, D. Wagner, M. M. Wilcox, A. H. Miller, C. B. Nemeroff, The role of early adverse experience and adulthood stress in the prediction of neuroendocrine stress reactivity in women: a multiple regression analysis, *Depression and anxiety*, Vol. 15, No. 3, pp. 117-125, 2002.

[20] C. W. Chu, G. P. Zhang, A comparative study of linear and nonlinear models for aggregate retail sales forecasting, *International Journal of production economics*, Vol. 86, No. 3, pp. 217-231, December, 2003.

[21] E. S. Gardner Jr., Exponential smoothing: The state of the art—Part II, *International journal of forecasting*, Vol. 22, No. 4, pp. 637-666, October-December, 2006.

[22] S. Makridakis, M. Hibon, ARMA models and the Box–Jenkins methodology, *Journal of forecasting*, Vol. 16, No. 3, pp. 147-163, May, 1997.

[23] F. Braghin, F. Lewis, R. Dwyer-Joyce, S. Bruni, A mathematical model to predict railway wheel profile evolution due to wear, *Wear*, Vol. 261, No. 11-12, pp. 1253-1264, December, 2006.

[24] D. Andre, C. Appel, T. Soczka-Guth, D. U. Sauer, Advanced mathematical methods of SOC and SOH estimation for lithium-ion batteries, *Journal of power sources*, Vol. 224, No. 15, pp. 20-27, February, 2013.

[25] C. K. Wells, A. R. Feinstein, S. D. Walter, A comparison of multivariable mathematical methods for predicting survival-III. Accuracy of predictions in generating and challenge sets, *Journal of Clinical Epidemiology*, Vol. 43, No. 4, pp. 361-372, 1990.

[26] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, *Neural computation*, Vol. 12, No. 10, pp. 2451-2471, October, 2000.

[27] M. R. Spitz, W. K. Hong, C. I. Amos, X. Wu, M. B. Schabath, Q. Dong, S. Shete, C. J. Etzel, A risk model for prediction of lung cancer, *Journal of the National Cancer Institute*, Vol. 99, No. 9, pp. 715-726, May, 2007.

[28] T. El Maaddawy, K. Soudki, A model for prediction of time from corrosion initiation to corrosion cracking, *Cement and concrete composites*, Vol. 29, No. 3, pp. 168-175, March, 2007.

[29] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tian, M. Adam, A. Gertych, R. S. Tan, A deep convolutional neural network model to classify heartbeats, *Computers in biology and medicine*, Vol. 89, pp. 389-396, October, 2017.

[30] Y. Xue, B. M. Onzo, R. F. Mansour, S. Su, Deep Convolutional Neural Network Approach for COVID-19 Detection, *Computer Systems Science and Engineering*, Vol. 42, No. 1, pp. 201-211, 2022.

[31] T. Y. Kim, S. B. Cho, Predicting residential energy consumption using CNN-LSTM neural networks, *Energy*, Vol. 182, pp. 72-81, September, 2019.

[32] T. Ergen, S. S. Kozat, Unsupervised anomaly detection with LSTM neural networks, *IEEE transactions on neural networks and learning systems*, Vol. 31, No. 8, pp. 3127-3141, August, 2020.

[33] Q. Q. He, C. Wu, Y. W. Si, LSTM with particle Swam optimization for sales forecasting, *Electronic Commerce Research and Applications*, Vol. 51, Article No. 101118, January-February, 2022.

[34] S. C. Chu, Z. G. Du, Y. J. Peng, J. S. Pan, Fuzzy hierarchical surrogate assists probabilistic particle swarm optimization for expensive high dimensional problem, *Knowledge-Based Systems*, Vol. 220, Article No. 106939, May, 2021.

[35] S. M. Zhang, X. Su, X. Jiang, M. Chen, T. Y. Wu, A Traffic Prediction Method of Bicycle-sharing based on Long and Short term Memory Network, *Journal of Network Intelligence*, Vol. 4, No. 2, pp. 17-29, May, 2019.

[36] P. Nagabushanam, S. T. George, S. Radha, EEG signal classification using LSTM and improved neural network algorithms, *Soft Computing*, Vol. 24, No. 13, pp. 9981-10003, July, 2020.

[37] S. Bouktif, A. Fiaz, A. Ouni, M. A. Serhani, Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches, *Energies*, Vol. 11, No. 7, Article No. 1636, July, 2018.

[38] L. Peng, S. Liu, R. Liu, L.Wang, Effective long short-term memory with differential evolution algorithm for

electricity price prediction, *Energy*, Vol. 162, pp. 1301–1314, November, 2018.
[39] Z. Chang, Y. Zhang, W. Chen, Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform, *Energy*, Vol. 187, Article No. 115804, November, 2019.
[40] R. Pal, A. A. Sekh, S. Kar, D. K. Prasad, Neural network based country wise risk prediction of COVID-19, *Applied Sciences*, Vol. 10, No. 18, Article No. 6448, September, 2020.
[41] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735-1780, November, 1997.
[42] A. D. Li, B. Xue, M. J. Zhang, Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies, *Applied Soft Computing*, Vol. 106, Article No. 107302, July, 2021.
[43] J. Yang, J. Yang, D. Zhang, J. Lu, Feature fusion: parallel strategy vs. serial strategy, *Pattern recognition*, Vol. 36, No. 6, pp. 1369-1381, June, 2003.
[44] J. J. Nassi, E. M. Callaway, Parallel processing strategies of the primate visual system, *Nature reviews neuroscience*, Vol. 10, No. 5, pp. 360-372, May, 2009.
[45] J. Ordieres, E. Vergara, R. Capuz, R. Salazar, Neural network prediction model for fine particulate matter (PM2. 5) on the US–Mexico border in El Paso (Texas) and Ciudad Juárez (Chihuahua), *Environmental Modelling & Software*, Vol. 20, No. 5, pp. 547-559, May, 2005.
[46] B. Sadeghi, A BP-neural network predictor model for plastic injection molding process, *Journal of materials processing technology*, Vol. 103, No. 3, pp. 411-416, July, 2000.
[47] A. E. Smith, A. K. Mason, Cost estimation predictive modeling: Regression versus neural network, *The Engineering Economist*, Vol. 42, No. 2, pp. 137-161, 1997.
[48] S. L. Lin, C. F. Huang, M. H. Liou, C. Y. Chen, Improving Histogram-based Reversible Information Hiding by an Optimal Weight-based Prediction Scheme, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 4, No. 1, pp. 19-33, January, 2013.

Biographies

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Professor of Shandong University of Science and Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section and Tainan Chapter Chair of IEEE Signal Processing Society. His current research interest includes the information hiding, artificial intelligence and wireless sensor networks.

Wei Wang received his B.S. degree from Yantai Nanshan College, China, in 2022. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests include swarm intelligence, watermarking, and QR codes.

Shu-Chuan Chu received a Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She has been a Research Fellow and Associate Professor in the College of Science and Engineering of Flinders University, Australia since December 2009. Currently, she is a Research Fellow with a Ph.D. advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. She also serves as an editorial board member for Engineering Applications of Artificial Intelligence (EAAI), Journal of Internet Technology (JIT) and Research Reports on Computer Science (RRCS). Her research interests are mainly in Swarm Intelligence, Intelligent Computing, and Wireless Sensor Networks.

Zhi-Yuan Shao received his B.S. degree from Taishan College of Science and Techonology, China, in 2020. He is currently pursuing the master degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests include swarm intelligence and knowledge graphs.

Hong-Mei Yang received the Ph.D. degree in Computer Software and Theory from Shandong University of Science and Technology in 2009. She is currently an associate professor at Shandong University of Science and Technology. Her current main research interests are image processing and multimedia communication security.