**Journal of Database Management**
Volume 34 • Issue 1

# Improved Equilibrium Optimizer for Short-Term Traffic Flow Prediction

Jeng-Shyang Pan, Shandong University of Science and Technology, China
https://orcid.org/0000-0002-3128-9025
Pei Hu, Shandong University of Science and Technology, China
Tien-Szu Pan, National Kaohsiung University of Science and Technology, Taiwan
Shu-Chuan Chu, Shandong University of Science and Technology, China*
https://orcid.org/0000-0003-2117-0618

## ABSTRACT

Meta-heuristic algorithms have been widely used in deep learning. A hybrid algorithm EO-GWO is proposed to train the parameters of long short-term memory (LSTM), which greatly balances the abilities of exploration and exploitation. It utilizes the grey wolf optimizer (GWO) to further search the optimal solutions acquired by equilibrium optimizer (EO) and does not add extra evaluation of objective function. The short-term prediction of traffic flow has the characteristics of high non-linearity and uncertainty and has a strong correlation with time. This paper adopts the structure of LSTM and EO-GWO to implement the prediction, and the hyper parameters of the LSTM are optimized by EO-GWO to transcend the problems of backpropagation. Experiments show that the algorithm has achieved wonderful results in the accuracy and computation time of the three prediction models in the highway intersection.

## KEYWORDS

Equilibrium Optimizer, Flow Prediction, Grey Wolf Optimizer, LSTM, Short-Term Traffic

## INTRODUCTION

With the development of social economy, the number of vehicles is increasing rapidly and the traffic congestion has seriously affected road safety and environmental pollution. The transportation departments formulate management strategy and improve service level through utilizing the existing highway facilities and transportation network resources (Chou et al., 2018). In recent years, intelligent transportation system (ITS) has become a hot research area (Wang, Chen, Cheng, Lin, and Lo, 2015; Zhuang, Luo, Pan, and Pan, 2020; Song, Pan, and Chu, 2020). In this paper, long short-term memory (LSTM) and meta-heuristics are studied here to predict the short-term traffic flow.

DOI: 10.4018/JDM.321758
*Corresponding Author

The prediction of traffic flow is an important part of ITS. It is a major reference in the vehicle path planning and the resources are allocated to the roads where congestion risks are most likely to occur (Chen, Lin, Chang, and Lo, 2012). The prediction is to estimate the traffic flow in the future period by the known historical data and the current traffic flow, which includes long-term and short-term predictions. The former is usually based on hours, days, months and even years to forecast the traffic flow, and it is helpful in planning the reasonable distribution of the road network. But it merely grasps the status of traffic flow from a macro perspective, it is difficult to meet people's requirement on traffic information. The short-term prediction effectively uses real-time data to forecast traffic conditions within 30 minutes in the future.

The neural network provides a simplified model that simulates the ability of biological neural assembly to solve multi-layer and nonlinear problems. Recently, it has received extensive research and attention from scholars. Lyu et al. used recurrent neural network (RNN) to obtain word representation, and then learned the importance of each word in text classification through convolutional neural network (CNN) (Lyu et al., 2021). The algorithm presented excellent performance on multiple text classification data sets. Due to the rich semantics and flexible representation of biomedical entities, the biomedical ontology matching problem remains an open challenge in the alignment quality. Xue et al. proposed an attention-based bidirectional LSTM network ontology matching technique to address this problem (Xue et al., 2021). Guo et al. suggested a method based on SIGNAN, which generated various high-fidelity images with only one input image (Guo et al., 2021). The method achieved the operations of shape change, illumination direction changes and super-resolution generation, and improved the sampling efficiency. Fantinato et al. investigated the integration of deep learning and service-oriented architecture (SOA) and discussed how SOA can be implemented by deep learning methods in different types of environments for various users (Fantinato et al., 2021).

Short-term traffic flow is a typical time series data. At present, the common prediction models of traffic flow contain autoregressive integrated moving average (ARIMA), Kalman filtering model and artificial neural network (ANN). Traditional prediction models are limited in finding the relationship between historical data and the future traffic flow, consequently, they don't understand deep correlation and implicit information. Deep neural network (DNN) implements complex non-linear relationship by distributed hierarchical feature representation (Jiao, Wu, Bie, Umek, and Kos, 2018; Shi, Guo, Niu, and Zhan, 2020; Xia, Wang, and Guo, 2020; Chen, Song, Hwang, and Wu, 2020). Many neural networks are proposed to assist traffic prediction, such as artificial neural network, RBF neural network, RNN and long short-term memory neural network (Jiang et al., 2018; Liu, Zhang, and Chen, 2019; Ke, Shi, Guo, and Chen, 2018).

Abdi and Moshiri (2015) introduced a new method for short-term traffic flow prediction based on artificial neural network. Mousavizadeh Kashi and Akbarzadeh (2019) combined wavelet transformation and artificial neural network to predict the short-term traffic flow. Wu, Tan, Qin, Ran, and Jiang (2018) proposed a model based on deep learning and attention to improve the prediction accuracy. Belhadi, Djenouri, Djenouri, and Lin (2020) used recurrent neural network to predict urban long-term traffic flow and the algorithm handled small and big data streams. Shen, Kong, and Chen (2011) adopted a hybrid model of Kalman filter, artificial neural network, and fuzzy combination to overcome the limitations of the existing single prediction models under different traffic conditions. Gu et al. (2019) claimed an improved Bayesian combination model with deep learning to transcend the error amplification of traditional methods. Zheng, Yang, Liu, Dai, and Zhang (2019) proposed a deep and embedded learning approach, which helped clear studying from route structure, fine-grained traffic information and weather conditions. Polson and Sokolov (2017) developed a deep learning model and successfully predicted the traffic flows in two special events. Liu, Chen, Zhuang, Guo, and Chen (2020) used LSTM to predict subway traffic flow by combining the optimized traditional input variables and meteorological variables. Zhao, Zeng, and Lu (2019) suggested a new short-term traffic flow prediction model based on the ensemble learning of LSTM, no negative constraint theory weight integration and population external optimization. Hodge, Krishnan, Austin, Polak, and Jackson (2014) introduced a binary neural network that incorporated spatiotemporal features into the forecasting process. Deng, Jia, and Chen

(2019) brought an improved random search based on unified design to optimize the hyper parameters of CNN. Liu and Chen (2017) proposed a hybrid model to predict the hourly passenger flow, which applied stacked autoencoders and a three-stage deep learning architecture. The backpropagation (BP) algorithm provides an efficient way to use the gradient descent on whole parameters. But it does not guarantee to find the global optimal solution and only acquires the local optimum. Improper settings even cause gradient explosion and gradient disappearance.

Meta-heuristic algorithm combines random search and local search, and quickly finds an optimal solution in a certain time (Lu, Hwang, and Huang, 2020; Xue and Wang, 2015; Meng and Pan, 2016; Chu, Huang, Roddick, and Pan, 2011). Scholars have utilized it in the neural network to train parameters and achieved great results (Hu et al., 2019; Pan, Hu, and Chu, 2019; Chu, Roddick, Su, and Pan, 2004). Traffic flow refers to the sum of the flow in a specific section of road within a unit time and the change of flow is a complex system with high time variability, uncertainty, and non-linearity. BP neural network has the abilities of self-learning and approaching any nonlinear system. Its performance is closely related to the selection of the parameters of neural network when it is used to model and predict traffic flow. Chan, Dillon, and Chang (2012) developed a short-term prediction model by combining the principles of particle swarm optimization (PSO), neural network and fuzzy inference system to adapt to the time-varying traffic flow. Li et al. (2019) applied a multi-objective PSO to optimize the parameters in the deep belief network (DBN) and enhanced the prediction ability of long-term. Raza and Zhong (2018) combined genetic algorithm (GA), neural network and local weighted regression to achieve the prediction under various input and traffic settings. Xu, Du, and Wang (2018) utilized the mind evolutionary algorithm (MEA) and wavelet neural network to predict short-term traffic flow. Although these algorithms have acquired great results, they do not fully consider the characteristics of meta-heuristics and the methods of jumping out of local traps, and even increase the time complexity.

As one of the popular deep learning models, LSTM deals with the dependence problems of short-term and long-term. This paper uses the advanced equilibrium optimizer algorithm to train the parameters of LSTM and realizes the short-term prediction of traffic flow. Equilibrium optimizer (EO) is a novel meta-heuristic algorithm, which adopts the conservation equation of mass in fluid mechanics (Faramarzi, Heidarinejad, Stephens, and Mirjalili, 2020). It appears high achievements in the engineering optimization applications of pressure vessel design, welded beam design and tension/compression spring design. Nonetheless, it has not yet applied in the neural network. The main contributions are summarized as follows.

1. It improves the solution quality of EO and does not increase extra evaluation of objective function.
2. The abilities of exploration and exploitation of new algorithm are proved by benchmark functions.
3. It applies the algorithm and LSTM to predict short-term traffic flow.

The remainder of this paper is as follows: Section 2 introduces the related works in this paper, including LSTM and EO. Sections 3 proposes the advanced EO algorithm (EO-GWO) to improve the optimal solutions and avoid falling into local traps. Section 4 verifies the solution abilities by 29 benchmark functions. Section 5 implements short-term traffic flow prediction with LSTM. Section 6 concludes the works and gives some advice for further studies.

## RELATED WORK

LSTM and EO are utilized to predict the short-term traffic flow and this section recalls the basic concepts of them.

### Long Short-Term Memory (LSTM)

Since the traditional RNN is equivalent to a multi-layer feedforward neural network, too many layers lead to the problems of gradient disappearance or explosion and lose historical information during training parameters. The historical information which it actually uses is very limited. Hochreiter and Schimidhuber (1997) proposed long short-term memory (LSTM) in 1997 to improve the traditional RNN model. The LSTM units are composed of memory cell, input gate and output gate. Cell is the core computing node and is used to record the current time state. Figure 1 is the unit of LSTM and its equations are as follows:

$$i_t = \delta\left(w_{xi}x_t + w_{xi}h_{t-1} + w_{ci}c_{t-1} + b_i\right) \tag{1}$$

$$f_t = \delta\left(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f\right) \tag{2}$$

$$c_t = f_t c_{t-1} + i_t tanh\left(w_{xc}x_t + w_{hc}h_{t-1} + b_c\right) \tag{3}$$

$$o_t = \delta\left(w_{xo}x_t + w_{ho}h_{t-1} + w_{co}c_{t-1} + b_o\right) \tag{4}$$

$$\mathrm{h}_t = o_t tanh\left(c_t\right) \tag{5}$$

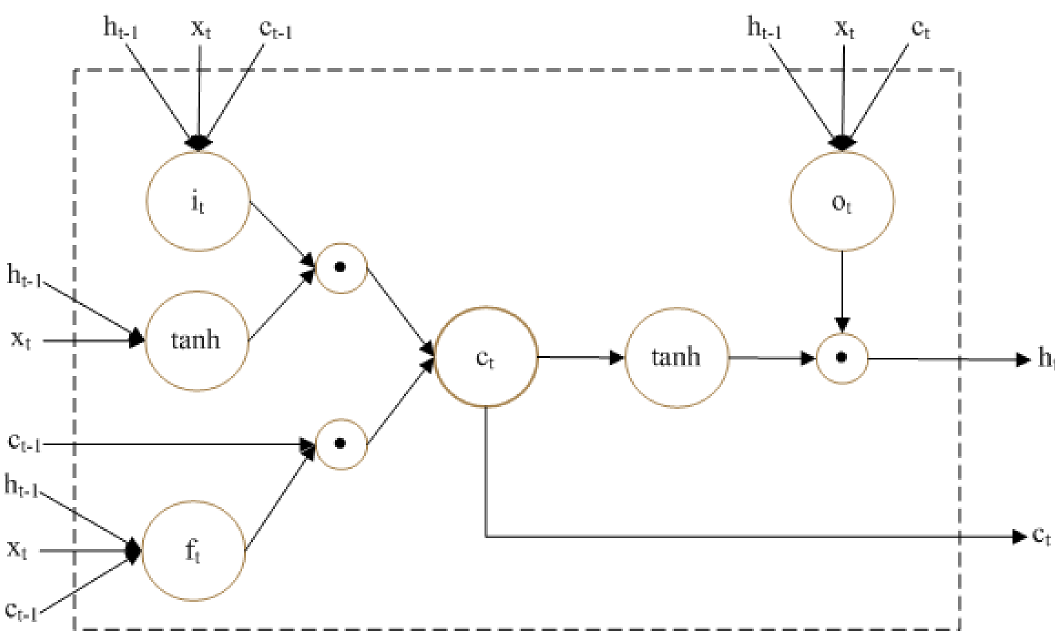
where $i$, $f$, $c$ and $o$ are input gate, forget gate, cell state and output gate respectively. $w$ and $b$ are weight coefficient and bias. $\delta$ is the sigmoid activation function. LSTM model includes the following steps:

(1) The output value of cell is calculated according to the Eqs. (1) - (5).
(2) The error term of cell is computed through the reverse direction.
(3) The weights and bias are updated with the gradient descent.

## Equilibrium Optimizer (EO)

In the EO, each particle represents a candidate solution, and the concentration is the position of a particle. The particles adopt Eq. (6) to update their concentrations by the global optimal solutions (equilibrium candidates) and they finally achieve the equilibrium (optimal solution).

**Figure 1.**
**The unit of LSTM**



$$\overrightarrow{X_i}\left(n+1\right)=\overrightarrow{X_{eq}}\left(n\right)+\left(\overrightarrow{X_i}\left(n\right)-\overrightarrow{X_{eq}}\left(n\right)\right)\vec{F}\left(n\right)+\frac{\vec{G}\left(n\right)}{\lambda V}\left(1-\vec{F}\left(n\right)\right) \tag{6}$$

where $\overrightarrow{X_i}\left(n\right)$ is the position of the $i^{th}$ particle at the $n^{th}$ iteration. Since $V$ is a unit volume, it has also been expressed as follows:

$$\overrightarrow{X_i}\left(n+1\right)=\overrightarrow{X_{eq}}\left(n\right)+\left(\overrightarrow{X_i}\left(n\right)-\overrightarrow{X_{eq}}\left(n\right)\right)\vec{F}\left(n\right)+\frac{\vec{G}\left(n\right)}{\lambda}\left(1-\vec{F}\left(n\right)\right) \tag{7}$$

$\overrightarrow{X_{eq}}$ is the equilibrium concentration and it is chosen from the equilibrium pool.

$$\overrightarrow{X_{eq}}\left(n\right)=\left\{\overrightarrow{X_{eq1}}\left(n\right),\overrightarrow{X_{eq2}}\left(n\right),\overrightarrow{X_{eq3}}\left(n\right),\overrightarrow{X_{eq4}}\left(n\right),\overrightarrow{X_{eqavg}}\left(n\right)\right\} \tag{8}$$

$\overrightarrow{X_{eq1}}$, $\overrightarrow{X_{eq2}}$, $\overrightarrow{X_{eq3}}$ and $\overrightarrow{X_{eq4}}$ are the positions of the first four optimal solutions in the control volume, respectively. $\overrightarrow{X_{eqavg}}$ is the average position of the four optimal solutions and $\overrightarrow{X_{eq}}$ randomly picks one from $\overrightarrow{X_{eq1}}$, $\overrightarrow{X_{eq2}}$, $\overrightarrow{X_{eq3}}$, $\overrightarrow{X_{eq4}}$ and $\overrightarrow{X_{eqavg}}$. $\vec{F}$ is an exponential term and controls the balance between exploration and exploitation of the algorithm. In the early phase, EO has the ability to search in more space and it gradually has a powerful exploitation with the evolution of the algorithm.

$$t\left(n\right)=\left(1-\frac{n}{Max\_iter}\right)^{a_2\frac{n}{Max\_iter}} \tag{9}$$

$$\vec{F}\left(n\right)=a_1 sign\left(\vec{r}-0.5\right)\left[e^{-\vec{\lambda}t\left(n\right)}-1\right] \tag{10}$$

where *Max_iter* means the maximum iteration. $\vec{r}$ and $\vec{\lambda}$ are two random vectors between [0,1]. $a_1$ controls the algorithmic exploration and $a_2$ determines its exploitation. Experiments show that $a_1 = 2$ and $a_2 = 1$, EO has the vigorous abilities of exploration and exploitation. *sign* is a signum function and controls the direction of exploration, where the value is computed by Eq. (11).

$$sign\left(x\right)\begin{cases}1 \ if(x>0)\\ 0 \ if\left(x=0\right)\\ -1 \ if(x<0)\end{cases} \tag{11}$$

$\vec{G}$ makes the algorithm powerfully find the optimal solution at the exploration stage and it is acquired by the following equations.

$$GCP=\begin{cases}0.5r_1 & if\left(r_2\le GP\right)\\ 0 & else\end{cases} \tag{12}$$

$$\overrightarrow{G_0}\left(n\right)=GCP*\left(\overrightarrow{X_{eq}}\left(n\right)-\overrightarrow{X_i}\left(n\right)\right) \tag{13}$$

$$\vec{G}\left(n\right) = \overrightarrow{G_0}\left(n\right) * \vec{F}\left(n\right) \tag{14}$$

where $r_1$ and $r_2$ are two random numbers between [0,1]. $GP$ dominates the participation probability of position updating through the generation rate. When $GP = 1$ implies that there are no particles joining in the optimization procedure and EO has high exploration ability. While $GP = 0$ represents that $X_{eq}$ participates in the optimization process. Empirical testing illustrates that EO has a well balance between exploitation and exploration when $GP = 0.5$. If $\vec{G}\left(1 - \vec{F}\right)$ and $\left(\overrightarrow{X_i} - \overrightarrow{X_{eq}}\right)\vec{F}$ have the same signs, $\vec{G}$ assists the algorithm searching in more place. When they have the opposite signs, $\vec{G}$ is helpful for local search.

## HYBRID OPTIMIZATION ALGORITHM EO-GWO

The four optimal solutions control the search direction of EO, and they affect the solution quality of the algorithm. To improve the convergence rate and avoid falling into local traps, this section proposes a hybrid algorithm EO-GWO, which includes a strategy to jump out of the local optimum.

The hybrid algorithms usually enhance the optimal solution of the population to improve the search speed, such as the *gbest* of PSO (Guo, Chen, Yu, Su, and Liu, 2015) and $\alpha$, $\beta$ and $\gamma$ of grey wolf optimizer (GWO) (Mirjalili, Mirjalili, and Lewis, 2014). They will be judged by the objective function when deeply optimizing them. If the objective function has less calculation amount, the time of the hybrid algorithms is not affected. However, the performance of the algorithms is seriously influenced when a lot of time is needed.

It is known from Eqs. (7) and (8) that the population is guided by the four optimal solutions of EO. If they are close to the optimal solutions, other particles quickly gather in this area and the convergence speed is improved. EO-GWO adopts the GWO algorithm to optimize them, so that they have better solutions and promote the exploitation ability of the algorithm. But the four optimal solutions are enhanced without calling the objective function to improve the execution efficiency. Suppose $\alpha$ $\beta$ and $\gamma$ are always $X_{eq1}$, $X_{eq2}$ and $X_{eq3}$. Their positions are updated by the following equations.

$$D_\alpha = \left|2r_2 \cdot X_{eq1} - X_i\right| \tag{15}$$
$$D_\beta = \left|2r_2 \cdot X_{eq2} - X_i\right| \tag{16}$$
$$D_\gamma = \left|2r_2 \cdot X_{eq3} - X_i\right| \tag{17}$$
$$X_1 = X_{eq1} - \left(2\mathrm{a} \cdot r_1 - \mathrm{a}\right) \cdot D_\alpha \tag{18}$$
$$X_2 = X_{eq2} - \left(2\mathrm{a} \cdot r_1 - \mathrm{a}\right) \cdot D_\beta \tag{19}$$
$$X_3 = X_{eq3} - \left(2\mathrm{a} \cdot r_1 - \mathrm{a}\right) \cdot D_\gamma \tag{20}$$
$$X_i\left(n+1\right) = \frac{X_1 + X_2 + X_3}{3} \tag{21}$$

where $r_1$ and $r_2$ are two random values between [0, 1]. With the iteration process, $a$ decreases linearly from 2 to 0.

After the method is executed, the optimized $X_{eq1}$, $X_{eq2}$, $X_{eq3}$ and $X_{eq4}$ estimate the objective function. EO-GWO judges whether they are superior to the original values, and it selects the four best solutions as the new positions of $X_{eq1}$, $X_{eq2}$, $X_{eq3}$ and $X_{eq4}$. The other particles are still updated by Eq. (7).

In EO, $X_{eq1}$, $X_{eq2}$, $X_{eq3}$ and $X_{eq4}$ have the same probability to direct the algorithm. Although the strategy makes the algorithm search more exploration space, it reduces the convergence rate. In order to improve the solution speed, the proportion that they are selected is 4: 2: 2: 1. The particles have a high probability of moving toward the $X_{eq1}$ and they also explore unknown space.

The above-mentioned methods accelerate the convergence. Once a objective function has multiple local optima or one solution is too excellent. EO-GWO is easy to fall into local traps and it is impossible to make further progress in searching for solutions. An escaping approach is utilized on the EO-GWO to improve the exploration ability.

In the meta-heuristic, exploration is the abilities of investigating different unidentified areas in the solution space and classifying the global optimum. If the fitness value of a particle is not updated for a long time, it means that the current search space of a particle is not in the region where the optimal solution is located. So, it is necessary to force it to search other places and the particle needs to learn from others. The equation is as follows:

$$X_i^j = X_{rj}^j \tag{22}$$

where *rj* is a random integer of [1, popSize] and $rj \neq i$. *popSize* is the number of population.

Each dimension of a particle randomly learns from the surrounding particles. If the area of the learned object has a better solution, the particle will move to it. If it is not, the particle in other dimensions influences its movement. This strategy improves the exploration and balances the exploitation. The pseudo code of the algorithm is shown in the Figure 2.

## EXPERIMENTAL RESULTS AND DISCUSSION

29 benchmark functions are used to validate the abilities of exploration and exploitation of EO-GWO. Table 1 shows the details of the functions (Hu, Pan, and Chu, 2020). *Limit* means the boundary of search space; *Dim* is the dimension and *fmin* indicates the theoretical optimum. *Type* represents the feature of benchmark function. They are composed of unimodal ($f_1$-$f_7$), multimodal ($f_8$-$f_{14}$), fixed-dimension ($f_{15}$-$f_{23}$) and composite ($f_{24}$-$f_{29}$) functions.

EO-GWO is compared with EO and two hybrid algorithms, WOA-SA (Mafarja and Mirjalili, 2017) and PSO-GWO (Mishra et al., 2020). Table 2 is the details of the algorithms. The number of population is 30 and the max iteration is 500. They run 30 times.

## EXPERIMENTAL RESULTS AND ANALYSIS

Table 3 shows the average values of optimal solutions. Wilcoxon's rank-sum test is implemented at a 5% significance level to judge whether the experimental results are statistically significant. Table 4 is the results of Wilcoxon's rank-sum test based on EO-GWO. "-" means that the algorithm is inferior to EO-GWO. "+" indicates that the compared algorithm is superior to EO-GWO. "=" represents that the algorithm and EO-GWO have the same performance.

As can be seen from Tables 3 and 4, EO-GWO has excellent performance. EO does well in 5 functions, while EO-GWO wins EO in 9 functions. Especially in the unimodal functions, the performance of EO-GWO has been greatly improved, which proves that the further optimization of the four optimal solutions and the position updating method accelerate the convergence of the algorithm and improve its exploitation. WOA-SA is better than EO-GWO in 7 functions and it has 16 functions worse than EO-GWO. The PSO-GWO algorithm does not optimize the optimal solution and it is found that EO-GWO is superior to PSO-GWO in both multimodal and unimodal functions. EO-GWO does not achieve as well as PSO-GWO in 4 functions and implements better than PSO-GWO in 16 functions.

In the unimodal functions, WOA-SA excels in $f_1$ and $f_2$, but it executes poorly in $f_3$ and $f_4$. EO-GWO has great achievements in $f_3$, $f_4$ and $f_5$. Especially in $f_5$, other algorithms are far away from the optimal

**Figure 2.**
**The pseudo code of EO-GWO**

```
Algorithm 1: EO-GWO
Initialize the positions and other parameters;
Calculate the objective function values;
Find Xeq1, Xeq2, Xeq3 and Xeq4;
for cyc = 1 : MAX_IT do
   for iter = 1 : MAX_IT2 do
      Execute Eqs. (15) - (21);
   endfor
   Calculate the objective function values of Xeq1, Xeq2, Xeq3 and Xeq4;
   Update Xeq1, Xeq2, Xeq3 and Xeq4;
   for i = 1 : popSize do
      if i != {Xeq1, Xeq2, Xeq3 and Xeq4} then
         prob = rand();
         if prob <= 0:4 then
            Xeq = Xeq1;
         else if prob <= 0:6 then
            Xeq = Xeq2;
         else if prob <= 0:8 then
            Xeq = Xeq3;
         else if prob <= 0:9 then
            Xeq = Xeq3;
         else
            Xeq = Xeqavg;
         Execute Eqs. (9) - (14);
         Calculate the objective function value;
   endfor
   Update Xeq1, Xeq2, Xeq3 and Xeq4;
   for i = 1 : popSize do
      if i falls into the trap then
         Execute Eq. (22);
   endfor
endfor
Output Xeq1.
```

value, while EO-GWO acquires the ideal result. WOA-SA and EO-GWO have the same performance in $f_6$, and PSO-GWO gets the worst data. EO and EO-GWO obtain the best result in $f_7$, but WOA-SA does not perform well. This illustrates that EO-GWO has a powerful exploitation ability and searches for the optimal solution in the known space. WOA-SA further enhances the optimal solution through simulated annealing (SA), but it does not complete successfully in the complex functions. The escaping strategy of EO-GWO avoids local traps and seeks for solutions in more unknown spaces. Although EO-GWO does not call the objective function for the optimized solutions to adjust the search direction, the algorithm still has excellent performance in the unimodal functions ($f_3$-$f_7$).

Multimodal function has many local optimal solutions, so it is suitable to judge whether the algorithm jumps local minima. EO-GWO implements perfectly in $f_9$, $f_{10}$, $f_{11}$ and $f_{12}$. EO achieves great results in $f_9$, $f_{11}$ and $f_{13}$, and WOA-SA has the best performance in $f_8$ and $f_{11}$. PSO-GWO does well in $f_{13}$. The behavior of EO-GWO is significantly better than other algorithms, which demonstrates that it is able to escape from the local traps and find the optimal solution in more space.

Fixed-dimension function merely has a few local optimum and its dimension is also small. The algorithms acquire the same data in $f_{16}$, $f_{17}$ and $f_{18}$. EO and EO-GWO perform well in $f_{14}$, $f_{15}$, $f_{20}$ and $f_{23}$. WOA-SA has outstanding accomplishments in $f_{21}$ and $f_{23}$. PSO-GWO implements wonderfully in

**Table 1.**
**The details of the benchmark functions**

| Name | Description | Limit | Dim | fmin | Type |
|---|---|---|---|---|---|
| $f_1$ | Sphere | [-100, 100] | 30 | 0 | Unimodal |
| $f_2$ | Schwefel's function 2.21 | [-10, 10] | 30 | 0 | Unimodal |
| $f_3$ | Schwefel's function 1.2 | [-100, 100] | 30 | 0 | Unimodal |
| $f_4$ | Schwefel's function 2.22 | [-100, 100] | 30 | 0 | Unimodal |
| $f_5$ | Rosenbrock | [-30, 30] | 30 | 0 | Unimodal |
| $f_6$ | Step | [-100, 100] | 30 | 0 | Unimodal |
| $f_7$ | Dejong's noisy | [-1.28, 1.28] | 30 | 0 | Unimodal |
| $f_8$ | Schwefel | [-500,500] | 30 | -12569 | Multimodal |
| $f_9$ | Rastringin | [-5.12,5.12] | 30 | 0 | Multimodal |
| $f_{10}$ | Ackley | [-32,32] | 30 | 0 | Multimodal |
| $f_{11}$ | Griewank | [-600,600] | 30 | 0 | Multimodal |
| $f_{12}$ | Generalized penalized 1 | [-50,50] | 30 | 0 | Multimodal |
| $f_{13}$ | Generalized penalized 2 | [-50,50] | 30 | 0 | Multimodal |
| $f_{14}$ | Fifth of Dejong | [-65, 65] | 2 | 1 | Fixed-dimension |
| $f_{15}$ | Kowalik | [-5, 5] | 4 | 0.00030 | Fixed-dimension |
| $f_{16}$ | Six-hump camel back | [-5, 5] | 2 | -1.0316 | Fixed-dimension |
| $f_{17}$ | Branins | [-5, 5] | 2 | 0.398 | Fixed-dimension |
| $f_{18}$ | Goldstein-Price | [-2, 2] | 2 | 3 | Fixed-dimension |
| $f_{19}$ | Hartman 1 | [1, 3] | 3 | -3.86 | Fixed-dimension |
| $f_{20}$ | Hartman 2 | [0, 1] | 6 | -3.32 | Fixed-dimension |
| $f_{21}$ | Shekel 1 | [0, 10] | 4 | -10.1532 | Fixed-dimension |
| $f_{22}$ | Shekel 2 | [0, 10] | 4 | -10.4028 | Fixed-dimension |
| $f_{23}$ | Shekel 3 | [0, 10] | 4 | -10.5363 | Fixed-dimension |
| $f_{24}$ | CF1 | [-5, 5] | 30 | 0 | Composite |
| $f_{25}$ | CF2 | [-5, 5] | 30 | 0 | Composite |
| $f_{26}$ | CF3 | [-5, 5] | 30 | 0 | Composite |
| $f_{27}$ | CF4 | [-5, 5] | 30 | 0 | Composite |
| $f_{28}$ | CF5 | [-5, 5] | 30 | 0 | Composite |
| $f_{29}$ | CF6 | [-5, 5] | 30 | 0 | Composite |

$f_{20}$ and $f_{22}$. Their results are not much different in the test functions, which verifies that the dimension of the objective function is an important indicator that influences the performance of the algorithm. EO-GWO does not execute well in the composite functions. Although it has a strategy to prevent jumping into local traps, the structures of functions are too complicated. The algorithms have not achieved ideal data in the functions.

From above discussion, the optimization ability is advanced by modifying the four optimal solutions and the new position updating strategy manages to avoid local traps. EO-GWO improves the solution quality.

**Table 2.**
**The details of the compared algorithms**

| Function | Parameters Setting |
|---|---|
| EO | a1=2; a2=1; GP=0.5; |
| EO-GWO | a1=2; a2=1; Max_iter2=5; |
| WOA-SA | T0=0.025; alpha=0.99; Max_iter2=5; |
| PSO-GWO | Vmax=6; wMax=0.9; wMin=0.2; c1=2; c2=2; Max_iter1=250; Max_iter2=250; |

**Table 3.**
**The statistical results of the compared algorithms**

| Function | EO | EO-GWO | WOA-SA | PSO-GWO |
|---|---|---|---|---|
| $f_1$ | 8.6683E-41 | 5.1649E-64 | **1.4918E-71** | 4.6528E-18 |
| $f_2$ | 8.7372E-23 | 1.0594E-38 | **3.2507E-48** | 2.5647E-10 |
| $f_3$ | 4.5256E-09 | **1.6450E-12** | 2.4600E+04 | 7.2919E-01 |
| $f_4$ | 3.6816E-10 | **3.6952E-12** | 4.8971E+01 | 1.2551E-02 |
| $f_5$ | 2.5346E+01 | **2.5057E+01** | 2.7481E+01 | 6.2451E+01 |
| $f_6$ | 1.0317E-05 | 1.5317E-06 | **1.4263E-06** | 2.8655E-04 |
| $f_7$ | **1.1645E-03** | 1.4513E-03 | 4.0469E-03 | 3.9574E-03 |
| $f_8$ | -8.9222E+03 | -8.9811E+03 | **-1.0677E+04** | -6.1824E+03 |
| $f_9$ | **0.0000E+00** | **0.0000E+00** | 2.7301E+01 | 6.9343E+01 |
| $f_{10}$ | 8.5857E-15 | **1.0066E-15** | 7.4015E-15 | 7.5613E-02 |
| $f_{11}$ | 4.9242E-04 | **2.4654E-04** | 1.2935E-02 | 1.1639E-02 |
| $f_{12}$ | 4.5348E-07 | **1.3828E-07** | 2.7403E-02 | 4.2836E-02 |
| $f_{13}$ | 2.9482E-02 | 8.6135E-02 | 1.2980E-01 | **1.0175E-02** |
| $f_{14}$ | **9.9800E-01** | **9.9800E-01** | 1.7588E+00 | 1.9519E+00 |
| $f_{15}$ | 1.6827E-03 | 3.7467E-03 | **6.4828E-04** | 5.9094E-03 |
| $f_{16}$ | **-1.0316E+00** | **-1.0316E+00** | **-1.0316E+00** | **-1.0316E+00** |
| $f_{17}$ | **3.9789E-01** | **3.9789E-01** | **3.9789E-01** | **3.9789E-01** |
| $f_{18}$ | **3.0000E+00** | **3.0000E+00** | **3.0000E+00** | **3.0000E+00** |
| $f_{19}$ | -3.8625E+00 | **-3.8628E+00** | -3.8568E+00 | **-3.8628E+00** |
| $f_{20}$ | -3.2659E+00 | -3.2600E+00 | -3.2341E+00 | **-3.2943E+00** |
| $f_{21}$ | **-8.5545E+00** | -7.6170E+00 | -7.9440E+00 | -5.1623E+00 |
| $f_{22}$ | -1.0049E+01 | -8.4310E+00 | -8.5250E+00 | **-9.1883E+00** |
| $f_{23}$ | **-9.8152E+00** | -9.3686E+00 | -6.9957E+00 | -9.2116E+00 |
| $f_{24}$ | 8.0000E+01 | **6.6668E+01** | 8.5548E+01 | 1.0000E+02 |
| $f_{25}$ | **1.1652E+02** | 2.3104E+02 | 2.0300E+02 | 1.3109E+02 |
| $f_{26}$ | **1.6095E+02** | 2.8067E+02 | 4.4225E+02 | 2.0338E+02 |
| $f_{27}$ | **3.6087E+02** | 7.3105E+02 | 5.6249E+02 | 3.6237E+02 |
| $f_{28}$ | **3.5437E+01** | 9.4831E+01 | 1.8251E+02 | 1.1459E+02 |
| $f_{29}$ | 8.0530E+02 | 9.0541E+02 | **7.5246E+02** | 7.6618E+02 |

**Table 4.**
**Wilcoxon's rank-sum test of the EO, WOA-SA and PSO-GWO on EO-GWO**

| Function | EO | WOA-SA | PSO-GWO | Function | EO | WOA-SA | PSO-GWO |
|---|---|---|---|---|---|---|---|
| $f_1$ | - | + | - | $f_{16}$ | = | = | = |
| $f_2$ | - | + | - | $f_{17}$ | = | = | = |
| $f_3$ | - | - | - | $f_{18}$ | = | = | = |
| $f_4$ | - | - | - | $f_{19}$ | - | - | = |
| $f_5$ | - | - | - | $f_{20}$ | = | - | = |
| $f_6$ | - | = | - | $f_{21}$ | = | + | - |
| $f_7$ | = | - | - | $f_{22}$ | = | + | = |
| $f_8$ | = | + | - | $f_{23}$ | = | - | - |
| $f_9$ | = | - | - | $f_{24}$ | = | - | = |
| $f_{10}$ | - | - | - | $f_{25}$ | + | = | = |
| $f_{11}$ | = | = | - | $f_{26}$ | + | - | + |
| $f_{12}$ | - | - | - | $f_{27}$ | + | + | + |
| $f_{13}$ | = | - | + | $f_{28}$ | + | - | = |
| $f_{14}$ | = | - | - | $f_{29}$ | + | + | + |
| $f_{15}$ | = | - | - | | | | |

### Convergence and Time Analysis

Table 5 shows the running time (seconds) of the algorithms. PSO-GWO has the shortest time, while EO-GWO and WOA-SA have large time complexity. This is because they increase the optimization of the optimal solutions in each iteration. EO-GWO does not add the evaluation of objective function, so the time is less than WOA-SA in most test functions. If the objective function is too simple, such as, $f_{15}$-$f_{23}$, the execution of objective function spends a small of time. The structure of EO-GWO is more complex than WOA-SA and the advantages of EO-GWO are not obvious.

Unimodal function only has a global optimal solution and no local trap. Consequently, it is useful to judge the convergence speed of the algorithm. From Figure 3 known, EO-GWO beats other algorithms. PSO-GWO has the slowest speed in $f_1$ and $f_2$. EO-GWO obtains the best performance before 400 iterations and WOA-SA acquires good convergence in the later iterations. In $f_3$ and $f_4$, WOA-SA has the worst speed and PSO-GWO does not perform well in $f_5$, $f_6$ and $f_7$. EO-GWO has the outstanding achievements in $f_3$, $f_4$ and $f_5$. The rate of EO-GWO is faster than EO and WOA-SA in $f_6$, and EO and EO-GWO almost have the same curve in $f_7$. Although WOA-SA improves the optimal solution, it can not search more space when the objective function is complicated. EO-GWO not only accelerates the convergence, but also searches for solutions in more space.

## APPLICATION FOR SHORT-TERM TRAFFIC FLOW PREDICTION

The neural network optimizes parameters through BP, hoewever this method has inherent defects such as slow learning speed, low accuracy, and easy to fall into local minima. Meta-heuristic algorithm is a global optimization process. Since it finds the global optimal solution in a multi-dimensional search space, it is widely used to train the parameters of neural networks. The parameters of neural network are optimized through meta-heuristic algorithm, and then the obtained parameters are further optimized accurately by the neural network.

**Table 5.**
**The running time of the compared algorithms**

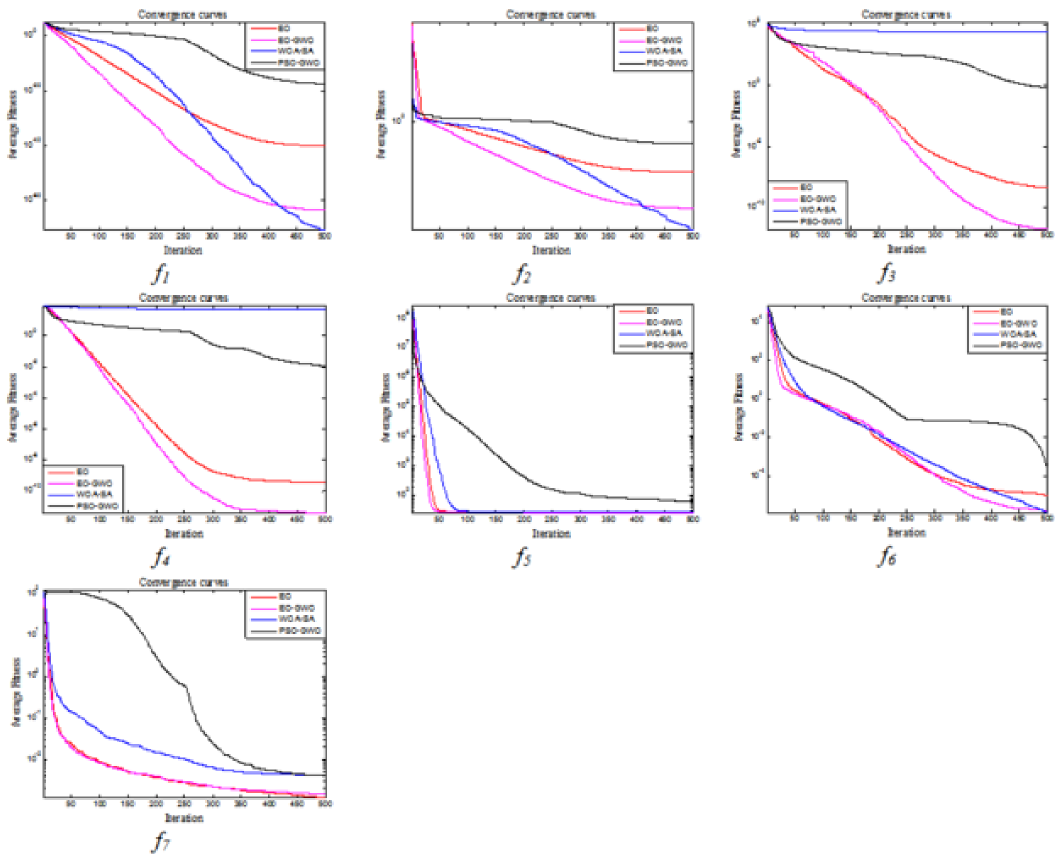
| Function | EO | EO-GWO | WOA-SA | PSO-GWO |
|---|---|---|---|---|
| $f_1$ | 2.1880E-01 | 4.6796E-01 | 6.1207E-01 | **1.1466E-01** |
| $f_2$ | 2.2590E-01 | 4.9172E-01 | 6.1968E-01 | **1.2138E-01** |
| $f_3$ | 6.8140E-01 | 8.1834E-01 | 1.1467E+00 | **5.7032E-01** |
| $f_4$ | 2.3910E-01 | 3.7684E-01 | 6.3226E-01 | **1.2945E-01** |
| $f_5$ | 2.7580E-01 | 4.3751E-01 | 6.2832E-01 | **1.7533E-01** |
| $f_6$ | 2.4510E-01 | 4.0175E-01 | 6.0227E-01 | **1.4646E-01** |
| $f_7$ | 3.0910E-01 | 4.6285E-01 | 6.8044E-01 | **2.0921E-01** |
| $f_8$ | 2.6360E-01 | 4.2518E-01 | 6.2151E-01 | **1.6381E-01** |
| $f_9$ | 2.6340E-01 | 5.6279E-01 | 6.2820E-01 | **1.6773E-01** |
| $f_{10}$ | 2.7250E-01 | 5.7558E-01 | 6.3197E-01 | **1.7631E-01** |
| $f_{11}$ | 2.8340E-01 | 5.7947E-01 | 6.5131E-01 | **1.9015E-01** |
| $f_{12}$ | 4.6280E-01 | 6.1837E-01 | 8.6146E-01 | **3.5414E-01** |
| $f_{13}$ | 4.5350E-01 | 6.1677E-01 | 8.5708E-01 | **3.5327E-01** |
| $f_{14}$ | 7.7730E-01 | 8.6787E-01 | 8.9496E-01 | **6.3633E-01** |
| $f_{15}$ | 2.4540E-01 | 3.4939E-01 | 2.8613E-01 | **1.1011E-01** |
| $f_{16}$ | 2.1788E-01 | 3.1589E-01 | 2.3929E-01 | **8.1823E-02** |
| $f_{17}$ | 2.1968E-01 | 3.2120E-01 | 2.4411E-01 | **8.6741E-02** |
| $f_{18}$ | 2.1051E-01 | 3.1614E-01 | 2.2002E-01 | **7.8254E-02** |
| $f_{19}$ | 2.7291E-01 | 3.7824E-01 | 3.2333E-01 | **1.5370E-01** |
| $f_{20}$ | 2.7780E-01 | 3.8719E-01 | 3.6277E-01 | **1.6027E-01** |
| $f_{21}$ | 3.0201E-01 | 4.0447E-01 | 3.6872E-01 | **1.8126E-01** |
| $f_{22}$ | 3.3245E-01 | 4.3163E-01 | 3.9059E-01 | **2.1050E-01** |
| $f_{23}$ | 3.7370E-01 | 4.7675E-01 | 4.4170E-01 | **2.5614E-01** |
| $f_{24}$ | 4.3470E+01 | 4.4240E+01 | 5.0697E+01 | **4.3188E+01** |
| $f_{25}$ | 4.5285E+01 | 4.6769E+01 | 5.2930E+01 | **4.5114E+01** |
| $f_{26}$ | 4.3877E+01 | 4.4466E+01 | 5.1681E+01 | **4.3873E+01** |
| $f_{27}$ | **5.0344E+01** | 5.8954E+01 | 5.9019E+01 | 5.0475E+01 |
| $f_{28}$ | **5.0275E+01** | 5.8610E+01 | 5.8932E+01 | 5.0459E+01 |
| $f_{29}$ | 5.0495E+01 | 5.8903E+01 | 5.9073E+01 | **5.0399E+01** |

The characteristics of traffic flow are mainly described by traffic flow, vehicle speed and density, among which traffic flow is particularly important. It intuitively reflects the operation of traffic; therefore traffic flow is generally selected as the prediction parameter. This section adopts the LSTM and meta-heuristics to predict short-term traffic flow.

## The Prediction Model of Traffic Flow Based on LSTM

The prediction is based on the historical data of traffic flow and uses a certain method to build a reasonably mathematical model to predict the future traffic flow. When forecasting the short-term

**Figure 3.**
**Convergence curves of the unimodal functions**




traffic flow at a highway intersection, the flow on the traffic section has an inevitable relationship with the previous several periods of the upstream. Consequently, they have the relationships of sequence and space. In this study, the data of the upstream is utilized to predict the traffic flow of the specific road and the following three models are applied to achieve short-term predictions, as shown in the Figure 4.where $f(x\text{-}1, t\text{-}1)$ represents the traffic flow of the last intersection of $x$ at time $t\text{-}1$.

The dimension of meta-heuristic is the number of parameters of neural network to be trained. The numbers of cell and output are 10 and 1. The input numbers are 1, 2 and 3 by Models 1, 2 and 3 respectively, so the dimensions are 360, 400 and 440. LSTM uses the sum of squared errors as the fitness function by default. The following equation is adopted to improve the prediction accuracy and the structure of the prediction model is shown in the Figure 5.

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

where $n$ is the number of training data, $y_i$ and $\overline{y_i}$ are the real and prediction values of $i^{th}$.

**Figure 4.**
**The prediction models of traffic flow**

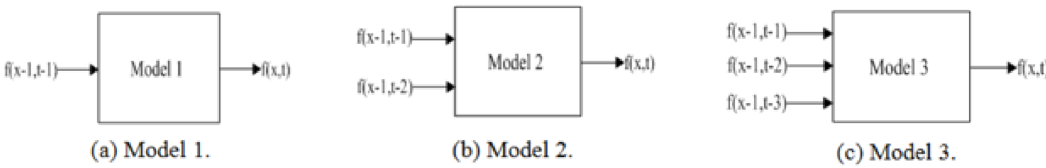
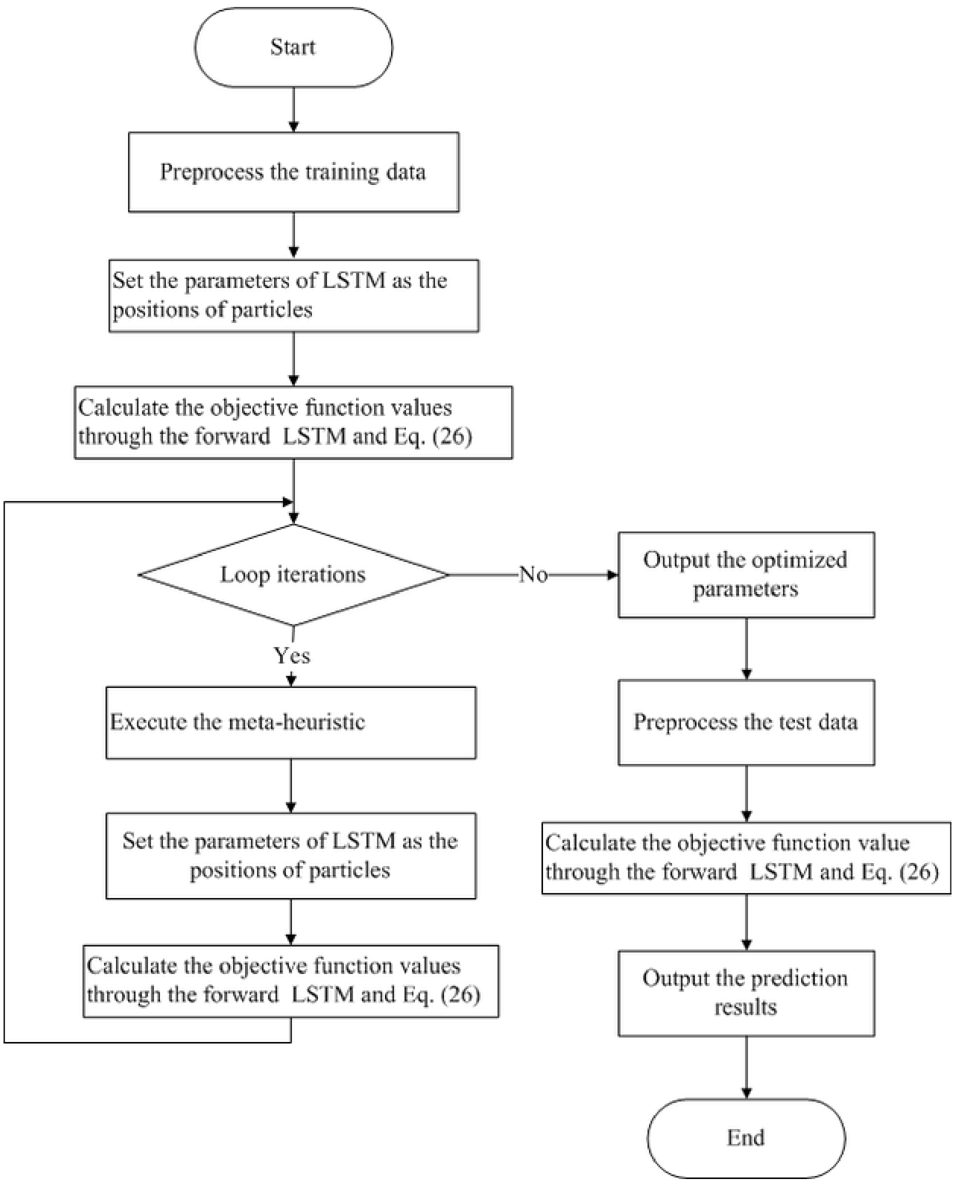



**Figure 5.**
**The flow chart of traffic flow prediction**



## SIMULATION RESULTS

The data comes from part of Wilson way OC, the PEMS data set of California highway network, and it contains the data between March 14, 2011 and May 29, 2011. The traffic volume is summarized at intervals of 5 minutes, so a traffic intersection has 288 data per day. The data from the first 53 days is used for training and the remaining 30% for testing. The compared algorithms contain EO, EO-GWO, WOA-SA, PSO-GWO, GA (Raza and Zhong, 2018) and PSO (Chan, Dillonm, and Chang, 2012). Tables 6 and 7 depicts the statistical results, including the values of prediction error (*Error*) and running time (*Time*). Tables 8 and 9 are the Wilcoxon's rank-sum test and Friedman test.

**Table 6.**
**The statistical results of prediction error**

| Model | EO | EO-GWO | WOA-SA | PSO-GWO | GA | PSO |
|---|---|---|---|---|---|---|
| *Model 1* | 4.0288E-01 | 4.0262E-01 | **4.0185E-01** | 4.0513E-01 | 4.5514E-01 | 4.0360E-01 |
| *Model 2* | 3.9585E-01 | 3.9459E-01 | **3.9444E-01** | 4.0094E-01 | 4.2553E-01 | 3.9740E-01 |
| *Model 3* | 3.8920E-01 | 3.8799E-01 | **3.8745E-01** | 3.9722E-01 | 4.0451E-01 | 3.9227E-01 |

**Table 7.**
**The statistical results of running time**

| Model | EO | EO-GWO | WOA-SA | PSO-GWO | GA | PSO |
|---|---|---|---|---|---|---|
| *Model 1* | **4.0955E+03** | 4.1344E+03 | 4.8274E+03 | 4.1109E+03 | 4.1714E+03 | 4.3566E+03 |
| *Model 2* | **4.1326E+03** | 4.1545E+03 | 4.8258E+03 | 4.1935E+03 | 4.2231E+03 | 4.4140E+03 |
| *Model 3* | **5.4105E+03** | 5.5314E+03 | 6.4495E+03 | 5.5119E+03 | 5.5418E+03 | 5.6884E+03 |

**Table 8.**
**Wilcoxon's rank-sum test of prediction error and time on EO-GWO**

| Model | EO | | WOA-SA | | PSO-GWO | | GA | | PSO | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Error | Time | Error | Time | Error | Time | Error | Time | Error | Time |
| *Model 1* | = | + | + | - | - | + | - | - | - | - |
| *Model 2* | - | + | = | - | - | + | - | - | - | - |
| *Model 3* | = | + | = | - | - | + | - | = | - | = |

**Table 9.**
**Friedman test of prediction error**

| Model | EO | EO-GWO | WOA-SA | PSO-GWO | GA | PSO |
|---|---|---|---|---|---|---|
| *Model 1* | 3 | 2 | 1 | 5 | 6 | 4 |
| *Model 2* | 3 | 2 | 1 | 5 | 6 | 4 |
| *Model 3* | 3 | 2 | 1 | 5 | 6 | 4 |
| *Average* | 3 | 2 | 1 | 5 | 6 | 4 |
| *P-value* | 2.4788E-03 | | | | | |

Table 6 shows the prediction errors of the three models, and WOA-SA are all smaller than those of EO, EO-GWO, PSO-GWO, GA and PSO. Compared to other algorithms, EO-GWO can improve the prediction accuracy by more than 13.26%, which is a good illustration of the ability of the proposed algorithm to capture the trend of data changes and thus make accurate predictions.

The selection of hyperparameters for LSTM prediction models has an important impact on prediction accuracy. Traditional hyperparameter selection methods are random; therefore, it can be reduced by forming hyperparameters into a multi-dimensional solution space and obtaining optimal parameter combination by traversing the solution space. Optimizing the LSTM using meta-heuristics yields smaller evaluation errors, and WOA-SA has the best performance, which presents that WOA-SA improves the overall performance of the LSTM. The proposed model has the prediction error with Model 3 of 38.75%, indicating that better parameters are obtained during the iterative process using the proposed algorithm. EO-GWO can effectively find the optimal parameter combination of LSTM network traffic prediction and reduce the prediction error.

Tables 6 and 7 reveals that WOA-SA has the best prediction errors and GA has the highest running time. EO wins the performance in the running time and PSO-GWO is superior to EO-GWO. Through the non-parametric statistical analysis, it is found that the prediction error of WOA-SA is better than EO-GWO in the Model 1 and they obtain the same statistical results in the Models 2 and 3. WOA-SA and EO-GWO beat others in the Model 2. EO is inferior to EO-GWO in the Model 2 and they acquire the same prediction in the Model 1. PSO-GWO, PSO and GA are not as good as EO-GWO in three models. Friedman test demonstrates that the rank value of EO-GWO is 2 and the P-value is 2.4788E-03. It is less than 0.05, so the hypothesis holds. In terms of running time, EO has the smallest time complexity in the three models. GA and PSO have the same statistical data as EO-GWO in the Model 3 and they are inferior to EO-GWO in the Models 1 and 2. EO-GWO is superior to WOA-SA and PSO-GWO in the Models 1, 2 and 3.

At the Model 1, according to the Eq. (23), EO, WOA-SA, EO-GWO, PSO-GWO, GA and PSO have the values of 4.0288E-01, 4.0185E-01, 4.0513E-01, 4.5514E-01 and 4.0360E-01, whereas EO-GWO has a value of 4.0262E-01. WOA-SA is superior to other algorithms, and EO and EO-GWO have the similar experimental data. PSO, PSO-GWO and GA are ranked 4, 5 and 6 respectively. At the Model 2, EO, WOA-SA, EO-GWO, PSO-GWO, GA and PSO have the values of 4.0288E-01, 3.9740E-01, 3.9459E-01, 4.0094E-01 and 4.2553E-01, while EO-GWO has a value of 3.9585E-01. EO-GWO and WOA-SA have the similar Wilcoxon's rank-sum. The ranks of EO, PSO, PSO-GWO and GA are 3, 4, 5 and 6 respectively. At the Model 3, EO, WOA-SA, EO-GWO, PSO-GWO, GA and PSO have the values of 3.8920E-01, 3.8745E-01, 3.9722E-01, 4.0451E-01 and 3.9227E-01, while EO-GWO has a value of 3.8799E-01. EO-GWO has the similar performance of WOA-SA in the Wilcoxon's rank-sum. The ranks of EO, PSO, PSO-GWO and GA are 3, 4, 5 and 6 respectively, the P-value is smaller than 0.05.

WOA-SA is suitable for identifying the optimal solution in the simple prediction model. EO-GWO illustrates its advantages when the model is complexity. It searches in more space and reduces the evaluation of the objective function. EO-GWO has made a well balance between prediction accuracy and running time. The algorithms have small prediction errors in the Model 3 and large errors in the Model 1, which presents that the prediction accuracy of the model is related to the used historical data.

The algorithms obtain poor prediction results on the *4567-th*, *1205-th*, *1198-th*, *1037-th*, *900-th*, *1336-th*, *633-th*, *578-th*, *1150-th* and *586-th* test data of the three models. This is because the data suddenly rises or falls, and the model cannot make a good judgment. Taking the *1198-th* as an example, the data in the Model 3 is as follows: 20, 24, 23 and the label data is 9, which has a wide deviation from the historical data. The predicted values of the model are between [17, 18] and the error rates exceed 80%.

## CONCLUSION

In order to advance the training accuracy and reduce running time of LSTM, this paper proposes the improved EO-GWO algorithm. Experiments show that EO-GWO has forceful solution ability and high efficiency in the benchmark functions. Traffic flow prediction is based on the historical traffic flow data. It utilizes a certain method to construct a reasonably mathematical model and then the model is adopted to predict the future traffic flow. LSTM is used to predict short-term traffic flow and EO-GWO algorithm is applied to train its parameters. EO-GWO achieves great results in the three prediction models of traffic flow. In future work, weather, date, emergencies and other factors can be added to the prediction model of traffic flow to achieve better results. Equilibrium optimizer and its variant are also applied to more deep learning structures.

## DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## ACKNOWLEDGMENT

This research was supported by the National Natural Science Foundation of China [grant number 61872085]; the Natural Science Foundation of Fujian Province [grant number 2018J01638]; the Fujian Provincial Department of Science and Technology [grant number 2018Y3001]; and the Key Science and Technology Research of Henan Province, China [grant number 212102210516]. Shu-Chuan Chu is a corresponding author on this paper.

## REFERENCES

Abdi, J., & Moshiri, B. (2015). Application of temporal difference learning rules in short-term traffic flow prediction. *Expert Systems: International Journal of Knowledge Engineering and Neural Networks*, *32*(1), 49–64. doi:10.1111/exsy.12055

Belhadi, A., Djenouri, Y., Djenouri, D., & Lin, J. C.-W. (2020). A recurrent neural network for urban long-term traffic flow forecasting. *Applied Intelligence*, *50*(10), 3252–3265. doi:10.1007/s10489-020-01716-1

Chan, K. Y., Dillon, T. S., & Chang, E. (2012). An intelligent particle swarm optimization for short-term traffic flow forecasting using on-road sensor systems. *IEEE Transactions on Industrial Electronics*, *60*(10), 4714–4725. doi:10.1109/TIE.2012.2213556

Chen, C.-H., Lin, B.-Y., Chang, H.-C., & Lo, C.-C. (2012). The novel positioning algorithm based on cloud computing-a case study of intelligent transportation systems. *Information-An International Interdisciplinary Journal*, *15*(11A), 4519–4524.

Chen, C.-H., Song, F., Hwang, F.-J., & Wu, L. (2020). A probability density function generator based on neural networks. *Physica A*, *541*(123344), 1–10. doi:10.1016/j.physa.2019.123344

Chou, C.-C., Shen, C.-W., Gao, D., Gao, Y., Wang, K., & Tsai, S.-B. (2018). Modelling the dynamic impacts of high speed rail operation on regional public transport—from the perspective of energy economy. *Energies*, *11*(5), 1–15.

Chu, S.-C., Huang, H.-C., Roddick, J. F., & Pan, J.-S. (2011). Overview of algorithms for swarm intelligence. In *Proceedings of the 3rd International Conference on Computational Collective Intelligence*. Springer Press.

Chu, S.-C., Roddick, J. F., Su, C.-J., & Pan, J.-S. (2004). Constrained ant colony optimization for data clustering. In *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*. Springer Press. doi:10.1007/978-3-540-28633-2_57

Deng, S., Jia, S., & Chen, J. (2019). Exploring spatial–temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data. *Applied Soft Computing*, *78*, 712–721. doi:10.1016/j.asoc.2018.09.040

Fantinato, M., Peres, S. M., Kafeza, E., Chiu, K. W., & Hung, C. K. (2021). A Review on the Integration of Deep Learning and Service-Oriented Architecture. [JDM]. *Journal of Database Management*, *32*(3), 95–119. doi:10.4018/JDM.2021070105

Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, *191*(105190), 1–21.

Gu, Y., Lu, W., Xu, X., Qin, L., Shao, Z., & Zhang, H. (2019). An improved Bayesian combination model for short-term traffic prediction with deep learning. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), 1332–1342. doi:10.1109/TITS.2019.2939290

Guo, W., Chen, G., Yu, C., Su, J., & Liu, Z. (2015). A two-stage clustering sleep scheduling algorithm with particle swarm optimization in wireless sensor networks. *Ad-Hoc & Sensor Wireless Networks*, *27*(1), 27–49.

Guo, Y., Pan, J. S., Qiu, C., Xie, F., Luo, H., Liu, Z., & Tan, J. (2021). SinGAN-Based Asteroid Surface Image Generation. [JDM]. *Journal of Database Management*, *32*(4), 28–47. doi:10.4018/JDM.2021100103

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735 PMID:9377276

Hodge, V. J., Krishnan, R., Austin, J., Polak, J., & Jackson, T. (2014). Short-term prediction of traffic flow using a binary neural network. *Neural Computing & Applications*, *25*(7), 1639–1655. doi:10.1007/s00521-014-1646-5

Hu, P., Pan, J.-S., & Chu, S.-C. (2020). Improved binary grey wolf optimizer and its application for feature selection. *Knowledge-Based Systems*, *195*(105746), 1–14. doi:10.1016/j.knosys.2020.105746

Hu, P., Pan, J.-S., Chu, S.-C., Chai, Q.-W., Liu, T., & Li, Z.-C. (2019). New hybrid algorithms for prediction of daily load of power network. *Applied Sciences*, *9*(21), 1–22.

Jiang, S., Lu, C., Zhang, S., Lu, X., Tsai, S.-B., Wang, C.-K., Gao, Y., Shi, Y., & Lee, C.-H. (2019). Prediction of ecological pressure on resource-based cities based on an RBF neural network optimized by an improved ABC algorithm. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 47423–47436. doi:10.1109/ACCESS.2019.2908662

Jiao, L., Wu, H., Bie, R., Umek, A., & Kos, A. (2018). Towards real-time multi-sensor golf swing classification using deep cnns. *Journal of Database Management*, *29*(3), 17–42. doi:10.4018/JDM.2018070102

Ke, X., Shi, L., Guo, W., & Chen, D. (2018). Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, *20*(6), 2157–2170. doi:10.1109/TITS.2018.2864612

Li, L., Qin, L., Qu, X., Zhang, J., Wang, Y., & Ran, B. (2019). Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. *Knowledge-Based Systems*, *172*, 1–14. doi:10.1016/j.knosys.2019.01.015

Liu, G., Chen, Z., Zhuang, Z., Guo, W., & Chen, G. (2020). A unified algorithm based on HTS and self-adapting PSO for the construction of octagonal and rectilinear SMT. *Soft Computing*, *24*(6), 3943–3961. doi:10.1007/s00500-019-04165-2

Liu, L., & Chen, R. C. (2017). A novel passenger flow prediction model using deep learning methods. *Transportation Research Part C, Emerging Technologies*, *84*, 74–91. doi:10.1016/j.trc.2017.08.001

Liu, L., Chen, R. C., & Zhu, S. (2020). Impacts of weather on short-term metro passenger flow forecasting using a deep lstm neural network. *Applied Sciences*, *10*(8), 1–17.

Liu, Y., Zhang, Y., & Chen, C.-H. (2019). Review on deep learning in intelligent transportation systems. In *Proceedings of the 28th International Conference on Computer Engineering and Networks*. Springer Press.

Lu, H.-C., Hwang, F., & Huang, Y.-H. (2020). Parallel and distributed architecture of genetic algorithm on apache hadoop and spark. *Applied Soft Computing*, *95*(106497), 1–15. doi:10.1016/j.asoc.2020.106497

Lyu, S., & Liu, J. (2021). Convolutional Recurrent Neural Networks for Text Classification. [JDM]. *Journal of Database Management*, *32*(4), 65–82. doi:10.4018/JDM.2021100105

Mafarja, M. M., & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, *260*, 302–312. doi:10.1016/j.neucom.2017.04.053

Meng, Z., & Pan, J.-S. (2016). Quasi-affine transformation evolutionary (quatre) algorithm: A parameter-reduced differential evolution algorithm for optimization problems. In *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE Press. doi:10.1109/CEC.2016.7744308

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Mishra, A. K., Das, S. R., Ray, P. K., Mallick, R. K., Mohanty, A., & Mishra, D. K. (2020). Pso-gwo optimized fractional order pid based hybrid shunt active power filter for power quality improvements. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 74497–74512. doi:10.1109/ACCESS.2020.2988611

Mousavizadeh Kashi, S. O., & Akbarzadeh, M. (2019). A framework for short-term traffic flow forecasting using the combination of wavelet transformation and artificial neural networks. *Journal of Intelligent Transport Systems*, *23*(1), 60–71. doi:10.1080/15472450.2018.1493929

Pan, J.-S., Hu, P., & Chu, S.-C. (2019). Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power. *Processes*, *7*(11), 1–24.

Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C, Emerging Technologies*, *79*, 1–17. doi:10.1016/j.trc.2017.02.024

Raza, A., & Zhong, M. (2018). Hybrid artificial neural network and locally weighted regression models for lane-based short-term urban traffic flow forecasting. *Transportation Planning and Technology*, *41*(8), 901–917. doi:10.1080/03081060.2018.1526988

Shen, G., Kong, X., & Chen, X. (2011). A short-term traffic flow intelligent hybrid forecasting model and its application. *Journal of Control Engineering and Applied Informatics*, *13*(3), 65–73.

Shi, Y., Guo, W., Niu, Y., & Zhan, J. (2020). No-reference stereoscopic image quality assessment using a multi-task CNN and registered distortion representation. *Pattern Recognition*, *100*(107168), 1–14. doi:10.1016/j.patcog.2019.107168

Song, P.-C., Pan, J.-S., & Chu, S.-C. (2020). A parallel compact cuckoo search algorithm for three-dimensional path planning. *Applied Soft Computing*, *94*, 106443–106458. doi:10.1016/j.asoc.2020.106443

Wang, H.-W., Chen, C.-H., Cheng, D.-Y., Lin, C.-H., & Lo, C.-C. (2015). A real-time pothole detection approach for intelligent transportation system. *Mathematical Problems in Engineering*, *2015*(869627), 1–7. doi:10.1155/2015/869627

Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C, Emerging Technologies*, *90*, 166–180. doi:10.1016/j.trc.2018.03.001

Xia, Y., Wang, J., & Guo, W. (2020). Two projection neural networks with reduced model complexity for nonlinear programming. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(6), 2020–2029. doi:10.1109/TNNLS.2019.2927639 PMID:31425123

Xu, L., Du, X., & Wang, B. (2018). Short-term traffic flow prediction model of wavelet neural network based on mind evolutionary algorithm. *International Journal of Pattern Recognition and Artificial Intelligence*, *32*(12), 1–14.

Xue, X., Jiang, C., Zhang, J., & Hu, C. (2021). Biomedical Ontology Matching Through Attention-Based Bidirectional Long Short-Term Memory Network. [JDM]. *Journal of Database Management*, *32*(4), 14–27. doi:10.4018/JDM.2021100102

Xue, X., & Wang, Y. (2015). Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio. *Artificial Intelligence*, *223*, 65–81. doi:10.1016/j.artint.2015.03.001

Zhao, F., Zeng, G.-Q., & Lu, K.-D. (2019). EnLSTM-WPEO short-term traffic flow prediction by ensemble LSTM, NNCT weight integration, and population extremal optimization. *IEEE Transactions on Vehicular Technology*, *69*(1), 101–113. doi:10.1109/TVT.2019.2952605

Zheng, Z., Yang, Y., Liu, J., Dai, H.-N., & Zhang, Y. (2019). Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3927–3939. doi:10.1109/TITS.2019.2909904

Zhuang, J., Luo, H., Pan, T.-S., & Pan, J.-S. (2020). Improved flower pollination algorithm for the capacitated vehicle routing problem. *Journal of Network Intelligence*, *5*(3), 141–156.

*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 in the College of Computer Science and Engineering, Shandong University of Science and Technology. He joined the Editorial Board of LNCS Transactions on Data Hiding and Multimedia Security, the Journal of Computers, and the Chinese Journal of Electronics. His current research interests include soft computing, information security, and signal processing.*

*Pei Hu received the M.Eng. degree in Computer Application Technology from Central South University, Changsha, China, in 2009. He is a lecturer and is currently pursuing the Ph.D. degree in the Shandong University of Science and Technology, Qingdao, China under the supervision of Dr. J.S.Pan. His research interests include evolutionary computation and neural network.*

*Tien-Szu Pan obtained his Doctorate Degree from the University of New Orleans, USA. Currently, he is an Associate Professor with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan. His current research interests include Computational Intelligence, Robot Design, Mechatronics, and Decentralized Control System.*

*Shu-Chuan Chu received the 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 is the Research Fellow in the College of Science and Engineering of Flinders University, Australia from December 2009. Currently, She is the Research Fellow with PhD advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.*