A Surrogate Model Based Genetic Algorithm for Complex Problem Solving

Ying Pei1, Hao Gao2 and Xiaosong Han2,*

1The College of Information Engineering, Changchun University of Finance and Economics, Changchun 130122, China
2 Key Laboratory for Symbol Computation and Knowledge Engineering of National Education Ministry, College of Computer Science and Technology, College of Software, Jilin University, Changchun 130012, China
Email: hanxiaosong@jlu.edu.cn

Abstract. It is well known that when the fitness function is relatively complex, the optimization time cost of the genetic algorithm will be extremely huge. To address this issue, the surrogate model was employed to predict the fitness value of the optimization problem, to reduce the number of actual calculated fitness values. In this paper, BP neural network, the least square method and support vector machine were fused in the genetic algorithm to evaluate partial individuals’ fitness. Sufficient benchmark numerical experiments were conducted, and the results proved that the strategy could reduce the calculating counts of fitness function on similar accuracy basis compared with simple genetic algorithm.

1. Introduction
Genetic algorithm (GA) is a stochastic optimization algorithm, which is evolved from the evolutionary phenomenon of survival of the fittest. After GA was put forward, because of its powerful global search ability, it has successfully been applied to various research fields. However, because the optimization results depend on the calculation of a large number of fitness functions, if the fitness calculation is very complex, the time cost of the algorithm optimization process will be highly expensive.

Many scholars have made some improvements to this problem. In 1995, Smith, Dike and Stegmann [1], proposed the fitness inheritance genetic algorithm. It was considered that the fitness of offspring chromosomes was inherited by the parent chromosome fitness. In 2001, Kim and Cho [2], proposed the cluster genetic hybrid algorithm, and achieved good optimization effect. In 2002, Emmerich et al. [3], proposed an algorithm for fitness estimation and confidence interval accumulation, which improved the searching ability of unknown regions. In 2008, Michael and Hod [4], proposed a prediction model of hybrid adaptive evolution, which uses the idea of swarm intelligence algorithm to evolve the fitness model parameters, and then used this model to predict the unknown individual fitness, got very good results. Yoel Tenne [5][6] of Kyoto University introduced a classifier into the optimization algorithm, which was used to predict whether the current parameter combination could lead to the failure of fitness calculation, thereby reducing the number of fitness calculations and improving the speed of optimization. In 2019, Lu Li [7] proposed some effective strategies to speed up PSO, and these can also be used in GA.

In this paper, we used surrogate model to train a simple regression machine to substitute fitness function. The regression algorithms were back propagation neural network, the least square method and support vector regression. In the following sections, we refer to back propagation neural network as BPNN, the least square method as LSM and support vector regression as SVR.
2. Surrogate Model Framework

Our framework is mainly described as follow. Surrogate models were employed to predict partial chromosome fitness. After that all individuals in the temporary population would have fitness values. Then, selection operation was performed on current population. In the selection operation, the winning chromosome would be called the next generation, when the population satisfied the requirement of the optimal solution, the algorithm would stop, or this loop evolved until it was satisfied. The flow of the genetic algorithm based on the surrogate model is given in Fig 1.

![Flow chart of the genetic algorithm based on the surrogate model](image)

Figure 1. Flow chart of the genetic algorithm based on the surrogate model

We implemented BPNN, LSM and SVR in the surrogate model section, respectively. BPNN [8], a feed-forward neural network for backward propagation learning, is one of the most widely used neural networks in machine learning field. According to the characteristics of BPNN, we proposed a new fitness prediction genetic algorithm, Back Propagation Neural Network Genetic Algorithm (BPGA). That is, using BP neural network regression analysis to predict partial individuals’ fitness values. LSM is a mathematical optimization technique [9], in this paper, the multiple linear regression analysis is carried out by the least square method, to find the functional relationship between the sample chromosome and fitness, obtain the regression coefficients, and use the fitted function is to predict other chromosomes’ fitness values. Similarly, LSM is also fused with GA, and the surrogate model is based on the multiple linear regression analysis. The new algorithm is named Least Square Genetic Algorithm (LSGA). Support vector machine (SVM) is a machine learning method based on the VC dimension theory and the minimum principle of structural risk in statistical learning theory [10]. The ε-SVR and GA was combined to a novel model, Support Vector Regression Genetic Algorithm (SVRGA). Support vector machines are used as surrogate models to predict fitness. In addition, the role of the kernel
function in SVR is very important. It transforms linear problems to nonlinear problems and maps training samples from lower dimensional space to higher dimensional space. In SVRGA, we chose the most commonly used Gauss radial basis function.

In the surrogate model framework, the chromosomes which fitness values were evaluated by fitness function were selected as input samples, namely training set. The training set maintaining strategy is simple. We used the First In First Out (FIFO) principles to complete this step. The surrogate model used a local training set instead of a global training set with all the chromosomes in history. Since the chromosomes in history are growing very fast, if all the chromosomes are added to the training set, the scale of the surrogate model will become larger and larger, and this will cause a huge training cost. Therefore, maintaining a fixed training set will take advantage of the model. What’s more, after evolving generation by generation, the population is quite different from the initial one. So, FIFO could ensure the training set is similar to latest generation populations, and the regression results will be quite reasonable.

3. Numerical Experiments

The 12 benchmark functions employed in this paper are derived from the Virtual Library of Simulation Experiments website [11]. Among them, function 1 to function 3 is a unimodal function, unimodal function is relatively simple, mainly used to test whether the optimized algorithm is effective. Function 5 to function 12 is a multimodal function, its local minimum points increase exponentially compared to the unimodal function, and the extremum of multimodal functions is multiple, which makes it difficult to solve the global extremum. We mainly use these multimodal functions to test the performance of the optimization algorithm. Function 4 is a discontinuous ladder function. All functions have a dimension of 30, and the global minimum is 0. Our goal is to find the global minimum of these functions.

| No. | Function name             | D                | Function name             | D                |
|-----|----------------------------|------------------|----------------------------|------------------|
| 1   | Sphere function            | [-100, 100]      | 7 | Ackley’s function          | [-32, 32]       |
| 2   | Quadric function           | [-100, 100]      | 8 | Generalized Griewank function | [-600, 600] |
| 3   | Rosenbrock function        | [-30, 30]        | 9 | Generalized penalized function 1 | [-50, 50] |
| 4   | Step function              | [-100, 100]      | 10 | Generalized penalized function 2 | [-50, 50] |
| 5   | Schwefel’s function        | [-500, 500]      | 11 | Noncontinuous Rastrigin’s function | [-0.5, 0.5] |
| 6   | Generalized Rastrigin function | [-5.12, 5.12] | 12 | Weierstrass function       | [-5.12, 5.12] |

The benchmark functions in Table 1 are the objective function of the GA to be measured. But these functions are looking for global minimum. We use the formula (3.1) to translate them into fitness functions for global maxima. At the same time, in order to facilitate the comparison and analysis of experimental results, the range of the function is scaled to the interval [0, 1000] by using the formula (3.2).

\[
\bar{f} = \text{obj}_{\text{max}} - \text{obj} \quad (1)
\]

\[
f = \frac{1000 \times (f - f_{\text{min}})}{f_{\text{max}} - f_{\text{min}}} \quad (2)
\]

In formula (3.1), \(\text{obj}_{\text{max}}\) is the maximum of the objective function, \(\text{obj}\) is the objective function value, \(\bar{f}\) is the fitness function value. In the formula (3.2), the \(f_{\text{min}}\) is the smallest fitness value, the \(f_{\text{max}}\) is the largest fitness value, \(f\) is a standardized fitness value.

We implemented the BPGA, LSGA and SVRGA on the basis of JGAP3.5 developed by Klaus [12]. JGAP is a public genetic algorithm project written by JAVA, and it provides a large number of application examples. Since JGAP is open source code and highly modular, it can be easily used to solve a variety of optimization problems, and the latest version is JGAP3.6.1. In this section, we will evaluate the advantages and disadvantages of the three algorithms from two aspects, the running speed and optimization results. The experimental results are as follows.
Table 2. Results comparison between BPGA, LSGA, SVRGA and SGA

| No. | BPGA Average fitness | BPGA Average run time | LSGA Average fitness | LSGA Average run time | SVRGA Average fitness | SVRGA Average run time | SGA Average fitness | SGA Average run time |
|-----|----------------------|-----------------------|----------------------|-----------------------|-----------------------|-----------------------|----------------------|---------------------|
| 1   | 961.21               | 1211                  | 965.45               | 4523                  | 928.29                | 981                   | 999.10               | 89                  |
| 2   | 969.75               | 880                   | 965.99               | 4913                  | 959.73                | 1431                  | 999.34               | 142                 |
| 3   | 996.39               | 1231                  | 992.39               | 6724                  | 976.39                | 1391                  | 999.99               | 123                 |
| 4   | 970.15               | 1181                  | 963.82               | 4693                  | 935.59                | 1371                  | 998.35               | 120                 |
| 5   | 840.22               | 850                   | 867.06               | 4833                  | 843.68                | 991                   | 980.74               | 84                  |
| 6   | 897.96               | 810                   | 894.33               | 5805                  | 813.52                | 991                   | 979.33               | 85                  |
| 7   | 458.21               | 920                   | 526.58               | 6205                  | 354.04                | 1010                  | 776.31               | 97                  |
| 8   | 964.17               | 860                   | 962.40               | 4853                  | 943.13                | 1001                  | 998.80               | 94                  |
| 9   | 958.35               | 1241                  | 939.49               | 5273                  | 917.68                | 1031                  | 1000.00              | 95                  |
| 10  | 929.67               | 870                   | 939.11               | 5203                  | 935.58                | 1034                  | 1000.00              | 89                  |
| 11  | 842.97               | 7005                  | 833.90               | 9805                  | 712.51                | 6775                  | 933.32               | 120                 |
| 12  | 975.49               | 820                   | 972.55               | 7556                  | 956.09                | 1041                  | 999.27               | 89                  |

From the point of view of running time, we can be drawn from the experimental results of the prediction table that the running time of the BPGA algorithm is the shortest, while the running time of SVRGA is the longest, and the running time of LSGA is in the middle. The reason is that the BP network performs network correction in the direction of the gradient of the error function, and its convergence depends on the initialization location of the learning model, so it is easy to fall into the local minimum. While SVRGA is slow, it is efficient and stable. Because that the training surrogate model is time-consuming, we run slower than SGA, but when the cost of fitness evaluation is great, the training model time is negligible. Experimental results show that when the fitness evaluation is over 8ms, our algorithm is faster than SGA. From the point of view of the test results, we compare the deviations between the average fitness obtained by each algorithm respectively, as a whole, SVRGA works best, BPGA and LSGA has a greater deviation between average fitness and optimal fitness in some multimodal test functions.

4. Conclusion

On the basis of simple genetic algorithm, we applied surrogate model to improve the calculation speed of chromosome fitness. The regression models used in this paper include BP neural network, multiple linear least squares regression and support vector regression. Finally, through benchmark experiments we have proved that the three regression models contribute to the computational efficiency of fitness in genetic algorithm. In terms of the speed of operation and the effect of the test, the three models show their respective advantages. SVRGA is the most stable one and achieves the best prediction effect, but the speed is slightly slower than the other two models. The BPGA’s the prediction effect is slightly lower than the SVRGA, better than LSGA, and the speed is faster. The surrogate model based on the least square method is slightly worse than the first two, but the operation speed is faster.

5. Acknowledgements

The work was supported by National Science Foundation of China under Grant 61876069 and Jilin Scientific and Technological Development Program under Grant 20180201045GX and 20180201067GX.

6. References

[1] Smith RE, Dike BA, Stegmann SA. Fitness inheritance in genetic algorithms [C]. In Proceedings of the ACMsymposium on App- lied computing, pp. 345-350. 1995.
[2] Kim H.S, Cho S.B. An efficient genetic algorithm with less fitness evaluation by clustering [C]. In Proceedings of IEEE Congress on Evolutionary Computation, pp. 887-894. 2001.

[3] Emmerich M, Giotis A, Ozdenir M, Back T and Giannakoglou K. Metamodel-assisted evolution strategies. In Parallel Problem Solving from Nature, pp. 371-380. 2002.

[4] Michael D. Shmidt and Hod Lipson. Coevolution of Fitness Predictor. In IEEE Transactions Evolutionary Computation, pp. 324-335. 2008.

[5] TenneYoel. A computational intelligence algorithm for expensive engineering optimization problems [J]. Engineering Applications of Artificial Intelligence, 25(5): 1009-1021. 2002.

[6] TenneYoel. An algorithm for computationally expensive engineering optimization problems [J]. International Journal of General Systems, 42(5): 458-488. 2013.

[7] Li Lu, Liang Yanchun, Li Tingting, et al. Boost particle swarm optimization with fitness estimation[J]. Natural Computing, 2019, 18(2):229-247.

[8] Rumelhart, David E, Hinton, Geoffrey E, Williams, Ronald J. Learning representations by back-propagating errors. Nature. 323 (6088): 533–536. 1986.

[9] Wang Guorong, Wei Yimin, Qiao SanZheng. Equation Solving Generalized Inverses. Generalized Inverses:Theory and Comput-ations. Beijing: Science Press: 6. 2004.

[10] Cortes C, Vapnik V. Support-vector networks. Machine Learning, 20 (3): 273–297. 1995.

[11] Surjanovic, S. & Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets [CP/OL]. Retrieved May 31, 2020, from http://www.sfu.ca/~ssurjano.

[12] Klaus M. Java Genetic Algorithms and Genetic Programming (JGAP). [CP/OL]. http://jgap.sourceforge.net/, 2012.