Method of Association Rules Mining and Its Application in Analysis of Seawater Samples

https://doi.org/10.3991/ijoe.v14i05.8648

Qiuhong Sun(✉)
Hebei University of Science and Technology, Shijiazhuang, China
Hebei Normal University, Shijiazhuang, China
sunqiuhong@hebust.edu.cn

Xinhang Xu, Yonghong Liu, Hongtao Zhang
State Grid Hebei Electric Power Research Institute, Shijiazhuang, China

Abstract—This paper aims to set up new rules for processing seawater quality monitoring data collected by photoelectric sensor network, and mine out the useful information contained in the data. For this purpose, the immune algorithm was introduced to the classical genetic algorithm, the fitness function was designed, and the crossover and mutation probabilities were adjusted, thus creating the adaptive immune genetic algorithm (IIGA). The new algorithm was described in details and applied in an actual case. Through the comparison between the IIGA, IGA and apriori algorithms, the author concluded that the IIGA not only shortened the mining time, but also ensured the operation accuracy. The research findings are of great importance to the association rules mining in various fields.

Keywords—Data mining, Association Rules, Immune Genetic Algorithm (IGA), Potential Data.

1 Introduction

In recent years, China has established a marine environment monitoring network in the seas under its jurisdiction. The monitoring agencies were assigned with clear duties on the monitoring and early warning of marine environment quality. The years of monitoring has accumulated a large amount of raw data, which far exceeds the scope of the single-index evaluation standards for marine water quality established in the past (1997). To solve the problem, this paper aims to set up new rules for processing seawater quality monitoring data collected by photoelectric sensor network, and mine out the useful information contained in the data.

In data mining, the most popular association rules algorithms are apriori algorithm and optimized apriori algorithms. The existing method accesses the database through various search algorithms, set support threshold to solve the frequent item sets, and generates association rules based on the frequent set through a certain algorithm. Since the database should be accessed repeated to determine the frequent item set, the

existing method may increase the burden of I/O and the workload, and reduce the mining efficiency. Therefore, the genetic algorithm has been introduced to the research on association rules.

The genetic algorithm is a research hotspot in foreign countries, as stated in [1-3]. combine immune algorithm and genetic algorithm into the immune genetic algorithm (IGA). Thanks to the unique concentration control function, the new algorithm outperforms immune algorithm and genetic algorithm in many aspects, such as the avoidance of local optimum trap, the maintenance of population diversity, and the accuracy of convergence results, as stated in [4-6]. So far, the IGA has been successfully applied in production system optimization, resource scheduling, allocation of water resources, etc.

In view of the fastness and accuracy of the IGA, this paper attempts to incorporate it into the research on association rules, and thus creates an improved immune genetic algorithm (IIGA) for the mining of association rules. Then, the IIGA was implemented in the mining of seawater quality monitoring database, aiming to dig out potentially useful information from the data.

2 Algorithm improvement

2.1 Algorithm analysis

Association rule mining is a procedure which is meant to find frequent patterns, correlations, associations, or causal structures from data sets found in various kinds of databases such as relational databases, transactional databases, and other forms of data repositories. Powerful search strategies are required to improve the performance of association rules mining.

Nowadays, the most popular association rules algorithms are apriori algorithm and optimized apriori algorithms, as stated in [7-9]. However, these algorithms face some shortcomings like high computing complexity. Therefore, the genetic algorithm has been gradually introduced to association rules mining.

Known for its high efficiency, the genetic algorithm is inspired by the biological competition strategy of survival of the fittest. It can encode and decode database information according to the specific strategy, and is suitable for encoding and searching. The problem is the genetic algorithm is sometimes troubled by low accuracy and redundancy.

In reference to the biological immune mechanism, the IGA is developed by integrating the memory function into the genetic algorithm. Compared with the genetic algorithm, the IGA carries the following prominent features, as stated in [3].

1. The fitness function and objective function are the antigen and the solution algorithm, respectively. The former is also the constraint of problem solving.
2. The antibody is the candidate solution of the problem, and the antibody set contains a group of antibodies. Similar to the genetic algorithm, the IGA also uses binary encoding and decimal encoding.

3. Repulsion between antibodies reflects the interaction between antibodies and the binding capacity of them.
4. The affinity of antigen antibody and antigen-induced antibody in different groups reveals the matching degree of antibody and antigen in the IGA.
5. The memory unit, as antibody group in the IGA, is a guarantee of the speed and quality of convergence and the diversity of the population.
6. Similar to organisms, the vaccine can prevent pathogenic organisms in advance through analysis of the pathogenic mechanism. The problem-solving process requires some prior knowledge of the evolution environment and the estimation of the best individual gene, as stated in [10-12].

2.2 Improvement design

In the genetic algorithm, some excellent genes are lost prematurely, due to the selection problem of crossover and mutation operators. In this case, the search range gets narrower, making it hard to find the global optimum. It also dampens the search efficiency in the late stage of evolution. In specific application, sampling error is almost inevitable if the number of data is very limited. The error will cause deviation from the expected results. The previous studies have shown that these defects can be resolved by self-adaptive genetic algorithm, as stated in [13-15]. Here, the self-adaptive genetic algorithm is incorporated into the IGA, forming the IIGA. In the old algorithm, the crossover and mutation probabilities are expressed as follows:

$$P_c = \begin{cases} P_{c1} - \dfrac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{\max} - f_{avg}} & f' \geq f_{avg} \\ P_{c1} & f' < f_{avg} \end{cases} \tag{1}$$

$$P_m = \begin{cases} P_{m1} - \dfrac{(P_{m1} - P_{m2})(f_{\max} - f')}{f_{\max} - f_{avg}} & f \geq f_{avg} \\ P_{m1} & f < f_{avg} \end{cases} \tag{2}$$

In the new algorithm, the crossover and mutation probabilities are expressed as follows:

$$P_c = \begin{cases} \dfrac{P_{c1}(f_{avg} - f') + P_{c2}(f' - f_{\min})}{f_{avg} - f_{\min}} & f' < f_{avg} \\ \dfrac{P_{c2}(f_{avg} - f') + P_{c3}(f' - f_{avg})}{f_{\max} - f_{avg}} & f' \geq f_{avg} \end{cases} \tag{3}$$

$$P_m = \begin{cases} \dfrac{P_{m1}(f_{avg} - f) + P_{m2}(f - f_{\min})}{f_{avg} - f_{\min}} & f < f_{avg} \\ \dfrac{P_{m2}(f_{avg} - f) + P_{m3}(f - f_{avg})}{f_{\max} - f_{avg}} & f \geq f_{avg} \end{cases} \tag{4}$$

where f_{max} is the maximum individual fitness of the population; f_{avg} is the average individual fitness of the population; f_{min} is the minimum individual fitness of the population; f' is the maximum fitness of the population; f is the fitness of mutated individuals.

In this research, the improved *Pc* and *Pm* are non-zero and automatically change with the individual fitness. Through the improvement, the genetic function of excellent individuals of the population increases, with no evolutionary stagnation or convergence to local optimum.

Comparing the average individual fitness in the current population, the excellent individuals should be retained in the genetic evolution of the population. In this way, the improved algorithm can jump out of the local optimum trap and avoid premature convergence. Besides, the IIGA consumes much less time in problem-solving than the original algorithms.

2.3 Algorithm design

Coding design. In terms of computer processing speed, binary encoding is the fastest, simplest to implement and most widely used on water quality monitoring. According to the characteristics of the water quality monitoring data in this paper, real number coding is selected. The real number encoding can be mixed with binary code, which can make a good mining effect.

Fitness function design. Fitness function is:

$$F(x) = w_s \times \frac{Supp(X)}{Supp_{\min}} + w_c \times \frac{Conf(X)}{Conf_{\min}} \tag{5}$$

Where $w_s + w_c = 1$, and w_s, w_c are constant; $Supp_{min}$ is pre-set support threshold. $Conf_{min}$ is pre-set confidence threshold. $Supp(x)$ is support value $Conf(x)$ is confidence value.

Design of immune memory function. Biological cells react much more slowly in the first invasion than in the second. But if the pathogen first invades, the body's memory cells would store the disease in the memory bank. So, when the pathogen strikes again, the body can respond quickly. The immune genetic algorithm simulates the immune system. Memory cells are implemented through a database. When the pathogen invades again, the saved data can be found in the database, thus speeding up the calculation.

Antibody promoting and blocking function design. The improved algorithm (IIGA) introduces the immune algorithm in the genetic algorithm, takes advantage of the immune algorithm. The fitness function is selected according to the actual problem. The formula of adaptive cross mutation probability is adjusted. The specific operations are as follows:

The antibody represents the fitness function $f(x_i)$，X is the immune system which is not empty.

The antibody to the set X：

$$\rho(x_i) = \sum_{j=1}^{N} \left| f(x_i) - f(x_j) \right| \tag{6}$$

Antibody concentration：

$$d(x_i) = \frac{1}{\rho(x_i)} = \frac{1}{\sum_{j=1}^{N} \left| f(x_i) - f(x_j) \right|} \tag{7}$$

Selection based on antibody concentration：

$$P_s(x_i) = \frac{\rho(x_i)}{\sum_{i=1}^{N} \rho(x_i)} = \frac{\sum_{j=1}^{N} \left| f(x_i) - f(x_j) \right|}{\sum_{i=1}^{N} \sum_{j=1}^{N} \left| f(x_i) - f(x_j) \right|} \tag{8}$$

The selection probability of the mixture concentration based on antibody and fitness is：

$$P(x_i) = \partial \times P_s(x_i) + (1-\partial) \times P_f = \partial \times \frac{\sum_{j=1}^{N} \left| f(x_i) - f(x_j) \right|}{\sum_{i=1}^{N} \sum_{j=1}^{N} \left| f(x_i) - f(x_j) \right|} + \frac{(1-\partial) f(x_i)}{\sum_{i=1}^{N} f(x_i)} \tag{9}$$

where P_f is The probability of choosing an individual based on fitness；∂ is concentration attenuation coefficient, $0<\partial<1$.

The adoption of this scheme can preserve the diversity of the population and increase the convergence rate. The greater the fitness function value is, the greater the probability of selection, and the promotion of the population. Corresponding to this, the higher the antibody concentration, the less likely it is to be selected.

Adaptive crossover and mutation operation design. As shown in 2.2. Adaptive crossover mutation is used to adapt the adaptive crossover mutation rate in the previous section, in order to improve the convergence of the algorithm. In the adaptive immune genetic algorithm, the crossover and mutation probability vary with the fitness function. The calculation formula of cross probability P_c is as shown as formula (3). The general value of P_{c1} is 0.9, and the general value of P_{c2} is 0.6.

In classical genetic algorithm, variation operation is a supplementary search operation. The variation operation is mainly used to maintain the individual diversity of the population.

The calculation formula of mutation probability P_m is as shown as formula(4). The general value of P_{m1} is 0.1，and the general value of P_{m2} is 0.001。

2.4 References

The adaptive immune genetic algorithm (IIGA) is introduced in this paper. Take advantage of the immune algorithm the fitness function is selected. According to the actual problem, the formula of adaptive cross mutation probability is adjusted.

The IIGA is implemented in the following steps:

1. Configure the parameters;
2. Generate the initial population with real coding;
3. Scan the entire database, and calculate *F(x)*, $Supp(x)$and $Conf(x)$ of each individual in database I;
4. Add the individuals surpassing the pre-set significance threshold to the rule table; otherwise, perform the following steps;
5. Carry out antibody promotion and blocking (selection) operations;
6. Perform adaptive crossover and mutation operations;
7. Terminate the implementation if the number of pre-set iterations is reached; otherwise, go to step (3);
8. Mine the output rule table and obtain the rule results.

The addition of antigen recognition greatly shortens the running time of the IIGA. Besides, the calculation formula of crossover and mutation probabilities are adjusted properly. The implementation process is shown in Figure 1.

2.5 Algorithm validation

The programming was carried out on Matlab2015 toolbox. The experimental data were extracted from the open source UCI Ecoli dataset. The specific parameters were configured as follows: *Pc*=0.95; *Pm*=0.01; $Supp(x)$=0.3; numbers=100; number of iterations=300; number of generations in the population=40.

The experimental results of the IIGA, the apriori algorithm and the IGA are shown in Table 1.

According to Table 1, the IIGA and IGA shared similar number of simplified properties, the number of breakpoints and the number of optimization rules; however, the IIGA outperformed the IGA and the apriori algorithm in operation accuracy and mining time. This is because the IIGA reduces the number of mining rules, improves the accuracy and shortens the execution time.

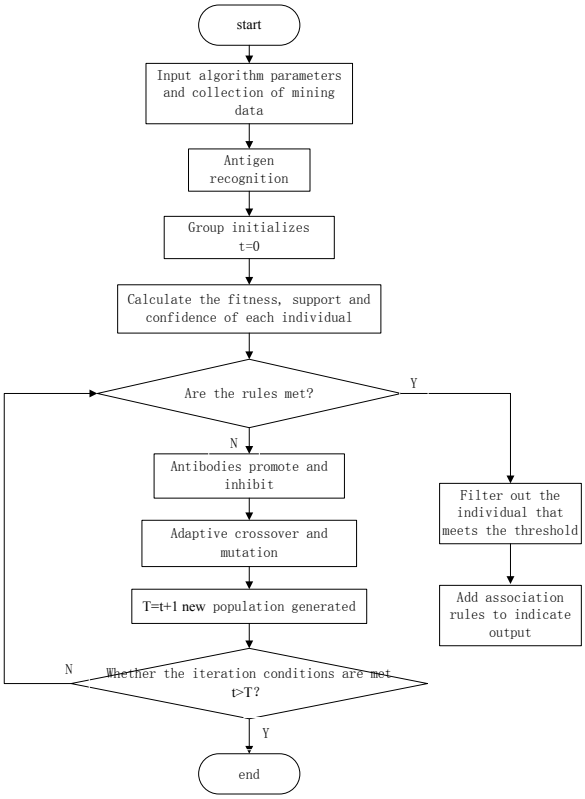

Fig. 1. IIGA flow chart

Table 1. Experimental results

	IIGA	**GA**	**Apriori**
Attribute number	5	5	7
breakpoints number	47	7	48
Rule number	26	6	36
Rule accuracy	98.6	8.1	91.3
Running time	18	9	46

The main reason for the short running time lies in the use of concentration selection plan based on vector moment. The high rule accuracy is attributed to the adjustment of crossover and mutation probabilities, which makes the mining rules more comprehensive, effective and concise.

Figures 2 and 3 compare the three algorithms at different support tresholds.

As can be seen from the two figures above, the IIGA maintained an edge over the IGA and the apriori algorithm in both running time and accuracy. Besides, the support threshold is negatively correlated with the running time and positively with the accuracy.

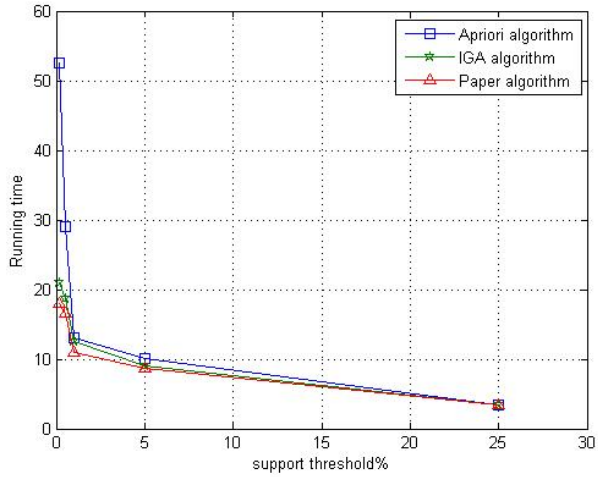

Fig. 2. The relationship between running time and support threshold

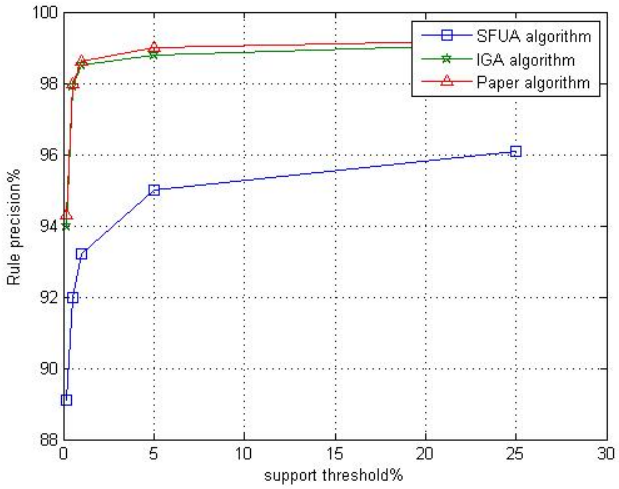

Fig. 3. The relationship between accuracy and the support threshold

3 Applications

3.1 Data preprocessing

The seawater quality data were obtained through the monitoring on a photoelectric sensor network. The data were analyzed by the IIGA for mining association rules based on data mining. Figure 4 shows the data captured from the photoelectric sensor network. The parameters include the time, latitude, temperature, salinity, turbidity, algae fluorescence, COD and buoy ID.

Because the mining parameters are numerical, real numbers were adopted in the following parts, and the monitoring value was divided into different intervals.

Table 2. Field description

Field name	Data type	instruction
SI	numeric	Wind wave intensity
TEMP	numeric	temperature
SAL	numeric	salinity
FA	numeric	Algae fluorescence
COD	numeric	Chemical oxygen demand
pH	numeric	pH value
TURB	numeric	turbidity

	A	B	C	D	E	F	G	H	I	J	K	L
1	Time	Latitude value	Longitude values	Salinity	Temperature	Turb	ST	Fluorescent algae	COD	The reserved inspection	Hi	Buoy ID
2	150909	3954.4474	11931.8872	0	0	0	0	0	10	7	3	8110004122
3	150915	3954.4474	11931.8872	0	4	1	1	3	12	10	25	8110004122
4	150921	3954.4474	11931.8872	0	3	0	3	2	13	11	7	8110004122
5	150927	3954.4474	11931.8872	0	2	0	1	2	14	11	7	8110004122
6	150933	3954.4474	11931.8872	0	0	0	0	2	10	7	6	8110004122
7	150939	3954.4474	11931.8872	0	3	1	2	3	13	0	0	8110004122
8	150945	3954.4474	11931.8872	1	1	7	4	2	13	11	6	8110004122
9	150951	3954.4474	11931.8872	0	0	6	3	0	10	10	6	8110004122
10	150957	3954.4474	11931.8872	0	3	2	2	0	11	11	6	8110004122
11	151003	3954.4474	11931.8872	0	1	0	7	1	11	10	4	8110004122
12	151009	3954.4474	11931.8872	0	0	0	0	0	4	2	2	8110004122
13	151015	3954.4474	11931.8872	0	0	4	0	0	7	4	1	8110004122
14	151021	3954.4474	11931.8872	0	2	8	1	0	11	10	6	8110004122
15	151027	3954.4474	11931.8872	0	0	0	0	0	9	7	3	8110004122

Fig. 4. Monitoring data

3.2 Coding

According to the actual needs of the IIGA algorithm, the monitoring values in each field were divided into different 1~n according to the intervals. In each field, 0 encoding can be added between one attribute and another. Under this constraint, the IIGA generated rules in a random manner. If the generated rule 02300 is covered by examples 22355 and 52366, then the rule will not be covered by example 35632. The parameters in Table 2 were mined and mapped to the results in Table 3.

3.3 Rule description

After mining the data in Table3, the generation rules were created (Table 4). Every rule and its encoding interval reflect a specific monitoring value. The potential information of monitoring data were mined according to different encoding rules.

Table 3. Mapping table

ID	SI	TEMP	SAL	FA	COD	pH	TURB
150908	1	1	1	1	1	2	1
150909	2	5	1	3	3	2	2
150910	2	4	1	2	4	1	1
150911	1	3	1	2	5	1	1
150912	1	1	1	2	1	2	1
150913	2	4	1	3	4	1	2
150914	1	2	2	2	4	3	8
150915	2	2	1	2	2	1	3
150916	1	3	1	3	5	2	3
150917	5	1	1	2	2	1	2
150918	2	2	2	3	4	3	3
150919	1	1	2	2	2	3	2
150920	1	1	2	3	5	3	3
150921	2	1	1	2	3	2	2
150922	4	1	1	1	2	1	1
150923	1	4	1	1	2	3	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 4. Rule description

Regular codes	parameter
0200010	75% support, 86 % confidence
0030003	62% support, 50% confidence
0100110	3 % support, 93 % confidence
0303000	81 % support, 97 % confidence
0020300	2 % support, 91 % confidence
0100010	68% support, 72 % confidence
3101010	80% support, 82 % confidence
0023020	29 % support, 88 % confidence
⋮	⋮

3.4 Equations

The monitoring data from the photoelectric sensor network were mined by the IIGA association rules. Taking rules 0200010, 0030003 and 0303000 for example:

(1) Rule 0200010: 56% means value of pH will be slightly higher if the water temperature rises; 86% means that it is very likely to happen in every season of the year.

(2) Rule 0030003: 62% means the turbidity will increase with water salinity; 50% means the situation is not very significant.

(3) Rule 0303000: 81% means the fluorescence seaweed will grow with the rise of seawater temperature; 97% means this situation is very likely to happen.

4 Conclusions

In this paper, the immune algorithm was introduced to the classical genetic algorithm, the fitness function was designed, and the crossover and mutation probabilities were adjusted, thus creating the adaptive immune genetic algorithm (IIGA). The new algorithm was described in details and applied in an actual case. Through the comparison between the IIGA, IGA and apriori algorithms, the author concluded that the IIGA not only shortened the mining time, but also ensured the operation accuracy. The research findings are of great importance to the association rules mining in various fields.

5 References

[1] Wang, J. Z., Zhang, F. Y, Liu, F., M, J. (2016). Hybrid Forecasting Model-based Data Mining and Genetic Algorithm-adaptive Particle Swarm Optimisation: A Case Study of Wind Speed Time Series. IET Renewable Power Generation, 10: pp. 287-298. https://doi.org/10.1049/iet-rpg.2015.0010

[2] Huynh, C. K., Lee, W. C. (2013). An Interference Avoidance Method using Two Dimensional Genetic Algorithm for Multicarrier Communication Systems. Journal of Communications and Networks, 15: pp. 486-495. https://doi.org/10.1109/JCN.2013.000088

[3] Ghorbaninejad, H., Heydarian, R. (2016). New Design of Waveguide Directional Coupler using Genetic Algorithm. IEEE Microwave and Wireless Components Letters, 26: pp. 86-88. https://doi.org/10.1109/LMWC.2016.2517165

[4] Abouelsaad, M. M., Abouelatta, M. A., Salama, A. R. (2013). Genetic Algorithm-optimised Charge Simulation Method for Electric Field Modelling of Plate-Type Electrostatic Separators. IET Science, Measurement & Technology, 7: pp. 16-22. https://doi.org/10.1049/iet-smt.2012.0058

[5] Mohammadi-Ivatloo, B., Rabiee, A., Soroudi, A. (2013). Nonconvex Dynamic Economic Power Dispatch Problems Solution using Hybrid Immune-Genetic Algorithm. IEEE Systems Journal, 7: pp. 777-785. https://doi.org/10.1109/JSYST.2013.2258747

[6] Ip, W. H., Wang, D., Cho, V. (2013). Aircraft Ground Service Scheduling Problems and Their Genetic Algorithm with Hybrid Assignment and Sequence Encoding Scheme. IEEE Systems Journal, 7: pp. 649-657. https://doi.org/10.1109/JSYST.2012.2196229

[7] Chung, S. H., Chan, H. K. (2011). A Two-Level Genetic Algorithm to Determine Production Frequencies for Economic Lot Scheduling Problem. IEEE Transactions on Industrial Electronics, 59: pp. 611-619. https://doi.org/10.1109/TIE.2011.2130498

[8] Verly, W., Araujo, L. R., Penido, D. R. R. (2016). A Method for Sizing of Industrial Electrical Systems using Genetic Algorithm. IEEE Latin America Transactions, 14: pp. 681-686. https://doi.org/10.1109/TLA.2016.7437210

[9] Tominaga, Y., Okamoto, Y., Wakao, S., Sato, S. (2013). Binary-based Topology Optimization of Magnetostatic Shielding by a Hybrid Evolutionary Algorithm Combining Genetic Algorithm and Extended Compact Genetic Algorithm. IEEE Transactions on Magnetics, 49: pp. 2093-2096. https://doi.org/10.1109/TMAG.2013.2240282

[10] Lu, T., Zhu, J. (2012). Genetic Algorithm for Energy-Efficient QoS Multicast Routing. IEEE Communications Letters, 17: pp. 31-34. https://doi.org/10.1109/LCOMM.2012.112012.121467

[11] Shi, L., Deng, Y. K., Sun, H. F., Wang, R., Ai, J. Q., Yan, H. (2012). An Improved Real-Coded Genetic Algorithm for the Beam Forming of Spaceborne SAR. IEEE Transactions on Antennas and Propagation, 60: pp. 3034-3040. https://doi.org/10.1109/TAP.2012.2194642

[12] Boudjelaba, K., Ros, F., Chikouche, D. (2014). Adaptive Genetic Algorithm-based Approach to Improve the Synthesis of Two-Dimensional Finite Impulse Response Filters. IET Signal Processing, 8: pp. 429-446. https://doi.org/10.1049/iet-spr.2013.0005

[13] Mota, T. A., Leal, J. F., Lima, A. C. (2015). Neural Equalizer Performance Evaluation using Genetic Algorithm. IEEE Latin America Transactions, 13: pp. 3439-3446. https://doi.org/10.1109/TLA.2015.7387252

[14] Thirugnanam, K., Singh, M., Kumar, P. (2014). Mathematical Modeling of Li-Ion Battery using Genetic Algorithm Approach for V2G Applications. IEEE Transactions on Energy Conversion, 29: pp. 332-343. https://doi.org/10.1109/TEC.2014.2298460

[15] Wei, X. K., Shao, W., Zhang, C., Li, J. L., Wang, B. Z. (2014). Improved Self-Adaptive Genetic Algorithm with Quantum Scheme for Electromagnetic Optimisation. IET Microwaves, Antennas and Propagation, 8: pp. 965-972. https://doi.org/10.1049/iet-map.2014.0034

6 Authors

Qiuhong Sun is a teacher of Hebei University of Science and Technology, Shijiazhuang 050000, China. And also works for Hebei Normal University, Shijiazhuang 050024, China. She works in the data analysis and evaluation for a long time.

Xinhang Xu works in State Grid Hebei Electric Power Research Institute, Shijiazhuang 050021, China. He often works in big data analysis, neural networks and algorithmic design.

Yonghong Liu works in State Grid Hebei Electric Power Research Institute, Shijiazhuang 050021, China. He works in data cleaning, data classification and programming.

Hongtao Zhang works in State Grid Hebei Electric Power Research Institute, Shijiazhuang 050021, China. He often works in decision tree, algorithm research and logic design.

Article submitted 30 March 2018. Final acceptance 04 May 2018. Final version published as submitted by the authors.