ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **1**, January **2016** pp. 145–151

BAYESIAN NETWORK PARAMETERS LEARNING METHOD BASED ON HYBRID SWARM INTELLIGENCE OPTIMIZATION ALGORITHM

Xiaoxin Du[1], Jianfei Zhang[1], Yuan Guo[1] and Bo Wang[2,*]

[1]College of Computer and Control Engineering
[2]Modern Education Technology Center
Qiqihar University
No. 42, Wenhua Street, Qiqihar 161006, P. R. China
*Corresponding author: wangbo_theone@163.com

Received July 2015; accepted October 2015

Abstract. *Bayesian network (BN) is an important tool for uncertainty knowledge representation and inference. BN parameters learning is one of the main steps of BN construction. It is a research hotspot in the field of data mining in recent years. When BN is complex, especially the network has hidden variables or small sample, the parameters learning is very difficult. In this paper we integrate cuckoo search algorithm (CSA) and cell membrane optimization algorithm (CMOA) to put forward a hybrid swarm intelligence optimization algorithm – cuckoo search algorithm optimized by cell membrane optimization algorithm (CSA-CMOA), and then bring CSA-CMOA into BN parameters learning. In our BN parameters learning method, we improve the convergence performance and speed of the algorithm by adjusting the random movement speed. BN parameters could be obtained by only giving the status of the evidence nodes and goal nodes, and do not need to know the middle nodes status, so it can realize the parameters learning with hidden variables. The experimental results show that the parameters learning method is effective and feasible.*
Keywords: Bayesian network (BN), Parameters learning, Cuckoo search algorithm (CSA), Cell membrane optimization algorithm (CMOA), Hidden variable

1. **Introduction.** Bayesian network (BN) [1] is an important tool for uncertainty knowledge representation and inference. At present it has been widely used for gene expression analysis [2], information retrieval [3], decision support systems [4], sports betting [5], and risk analysis [6]. BN parameters learning is one of the main steps of BN construction. For specific areas, the structure of BN can generally be obtained by domain knowledge and expert experience, but the parameters of the BN can only be got by learning. When BN is complex, especially the network has hidden variables or small sample, the parameters learning is very difficult. At present, expectation maximization (EM) algorithm and Gibbs sampling algorithm are two main parameters learning methods based on incomplete data. These two algorithms both have hypothesis of random lack, and they estimate missing variables by analyzing instances with complete data. However, for the presence of hidden variables there are no better BN parameters learning methods.

Cuckoo search algorithm (CSA) is a new type of swarm intelligence optimization algorithm which simulates natural cuckoo spawning behavior which was proposed by Cambridge University scholars Yang and Suash in 2009 [7]. The algorithm has advantages of few parameters and easy to implement. At present, it has been successfully applied in the practical problems of engineering optimization [8,9], and it has been developed into a new bright spot in the field of intelligent bionic optimization algorithm. However, CSA has weak search activity and slow convergence speed. Cell membrane optimization algorithm (CMOA) is an intelligent optimization algorithm that simulates cell membrane transport way. It has excellent global optimization capability, rapid convergence and the ability to obtain accurate solution. Especially in solving high-dimensional optimization problems it has better convergence performance [10]. In recent years it has been used in combinatorial optimization problems such as geometric constraint solving [11]. However, CMOA is easily trapped into local extremum and has slow convergence speed.

The process of BN parameters learning is a kind of optimization process. It is to find the best fitting parameters with the sample data. Especially for hidden variables or small sample, the selection of optimal parameters has great influence on BN parameters learning. Based on this, in this paper we integrate CSA and CMOA to put forward a hybrid swarm intelligence optimization algorithm – cuckoo search algorithm optimized by cell membrane optimization algorithm (CSA-CMOA), and CSA-CMOA can eliminate the defects of CSA and CMOA. And then we bring CSA-CMOA into BN parameters learning, through CSA-CMOA automatic optimization to achieve the purpose of BN parameters learning. This method is more suitable for BN parameters learning with hidden variables. This paper is organized as follows. The CSA-CMOA is put forward in Section 2. Section 3 discusses BN parameters learning method based on CSA-CMOA and gives the experimental result. The paper is concluded by Section 4.

2. Description of CSA-CMOA.

2.1. Cuckoo search algorithm (CSA). CSA uses the following representations [7].

Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. In the simplest form, each nest has one egg. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions.

CSA is based on three idealized rules:

1. Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generation;
3. The number of available hosts nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $\psi \in (0, 1)$. Discovering operate on some set of worst nests, and discovered solutions dumped from farther calculations.

2.2. Cell membrane optimization algorithm (CMOA). The substance is divided into three types according to the process of cell membrane transport substance: liposoluble substance, high concentration non-fat-soluble substances (HS) and low concentration non-fat-soluble substances (LS). Common terms in CMOA are stated as follows. Substance: solution. Concentration: the percentage of the substance number within a neighborhood and the total substance number. Liposoluble substance: elegant solution of the current iteration. Non-fat-soluble substances: inferior solution of the current iteration.

CSA has the advantages of simple optimization model, less control parameters, but search activity of CSA is not strong, and the convergence speed is slow. CMOA has strong search activity, but optimization model is too complex. From the above we can see CSA and CMOA have formed a significant complementary. So in this paper we put forward CSA-CMOA in order to improve the algorithm efficiency.

2.3. Related definitions of CSA-CMOA. In this paper, we study the unconstrained function optimization problems, as shown in Formula (1).

$$\begin{cases} \min f(x) \\ \text{s.t. } x \in [l \quad u] \end{cases} \tag{1}$$

In Formula(1), $[l \quad u] := \{x \in R^n \,|l_k \leq x_k \leq u_k,\, k = 1, \ldots, n\}$. $f(x)$ is optimization function, and $\min f(x)$ is minimum of optimization function. Assuming that Formula (1) constant has solution, that is the global optimal value exists.

Definition 2.1. ***(The search space frequency of Lévy flight)*** *For a nest, its search space frequency is defined as the ratio of the nest number within the scope of search space and the total number of the nests, see Formula (2).*

$$con = \frac{n^*}{n} \tag{2}$$

In Formula (2), n^* indicates the cuckoo number within the scope of search space, and n indicates the total number of the cuckoos.

Definition 2.2. ***(constrained search rule of Lévy flight)*** *In a certain search space, make a certain nest as the center and the Radius as the search radius, and randomly generate a new nest. Shrink the search radius according to* $Radius = Radius \times Pb$ *until* $\max\{Radius_k, \forall k\} > Pa$*. Among them, Pa is critical value of search stop, Pb is search rate of search radius, u is the upper bound of the solution space, and* n_1 *is the total number of nests which perform constrained search rule. The formula of search radius Radius is shown in Formula (3).*

$$Radius = \frac{u-1}{2 \times n_1} \tag{3}$$

Definition 2.3. ***(neighborhood updating rule of Lévy flight)*** *Randomly generate a number* ϕ *in the range of* $[0, 1]$*. Assuming that the probability of nest with carrier is Pc, if* $\phi \leq Pc$*, the nest moves from the search space of low frequency to high frequency, and consider high frequency location coordinates in the new search space as local search center; else, local search center sets at the original high frequency location coordinates. The cuckoo moves around local search center for local search. The formula of search radius* $Radius'$ *is shown in Formula (4).*

$$Radius' = \frac{u-1}{2 \times n_2} \tag{4}$$

In Formula (4), u is the upper bound of the resolution space, and n_2 is total number of nests which perform neighborhood updating rule. Make the new local search center as the center and the $Radius'$ as the search radius, the cuckoos randomly move l times and modify their scopes, and then record the optimal nest position in l nests.

When the logical value of Formula (5) is true, start constrained search rule and neighborhood updating rule of Lévy flight, otherwise perform no action.

$$\frac{\left|Fit\left(X_i^{(t)}\right) - Fit\left(X_i^{(t-1)}\right)\right|}{Fit\left(X_i^{(t)}\right)} \leq \Omega \text{ and } \frac{\left|Fit\left(X_i^{(t-1)}\right) - Fit\left(X_i^{(t-2)}\right)\right|}{Fit\left(X_i^{(t-1)}\right)} \leq \Omega \tag{5}$$

In Formula (5), $Fit\left(X_i^{(t)}\right)$ is the fitness value of optimal nest position in the tth iteration, $Fit\left(X_i^{(t-1)}\right)$ is the fitness value of optimal nest position in the $(t-1)$th iteration, $Fit\left(X_i^{(t-2)}\right)$ is the fitness value of optimal nest position in the $(t-2)$th iteration, and Ω is threshold (the value is experimental experience value, and in this paper the value is 0.0075).

2.4. **Description of CSA-CMOA.**

Step 1 Initialize the nests, randomly generate the position of nests $X_i^{(i)} = \left(X_1^{(i)}, X_2^{(i)}, \ldots, X_n^{(i)}\right)$, and n is the number of nests.

Step 2 Calculate initial fitness value of each nest according to the fitness function $Fit\left(X_i^{(i)}\right)$.

Step 3 The owner of each nest improves their own nest through Lévy flight mode, calculates the fitness value of each improved nest, compares it to the former fitness value, and retains better quality nest according to greedy heuristics. The owner of nest improves the nest according to Formula (6).

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus s \tag{6}$$

In Formula (6), t indicates the current number of iterations, α is control parameter of step length, the value obeys the standard normal distribution, $\oplus$ indicates the point to point multiplication, s is search path of Lévy flight, that is the step length of flight, see Formula (7).

$$s = 0.01 \times \frac{\mu}{|\upsilon|^{1/\beta}} \times \left(g_{best} - X_i^{(t)}\right) \tag{7}$$

In Formula (7), coefficient 0.01 is typical flight scale in the Lévy fight, μ and υ obey uniform distribution, that is $\mu \sim N\left(0, \delta_\mu^2\right)$, $\upsilon \sim N\left(0, \delta_\upsilon^2\right)$. The definition of δ_μ is shown in Formula (8), and the value of δ_υ is 1.

$$\delta_\mu = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right] \times \beta \times 2^{(\beta-1)/2}} \right\}^{1/\beta} \tag{8}$$

In Formula (8), Γ expresses standard gamma function, in this paper we set the value of β is 2/3, and g_{best} indicates the nest with the current optimal fitness value.

Step 4 Improve the nest with poor fitness value according to Formula (9).

$$X_i^{(t+1)} = X_i^{(t)} + rand \times \left(X_j^{(t)} - X_i^{(t)}\right) \tag{9}$$

In Formula (9), $rand$ is a random number between $[0, 1]$, and $X_j^{(t)}$ is a nest near the $X_i^{(t)}$.

Step 5 Compare the improved nest with the nest which possesses the current optimal fitness value.

Step 6 Determine whether meeting constrained search rule and neighborhood updating rule of Lévy flight as shown in Formula (5), if meeting turn to Step 7, otherwise, turn to Step 8.

Step 7 Perform constrained search rule and neighborhood updating rule according to Formula (3) and Formula (4).

Step 8 Perform Step 2 to Step 7 until reaching the maximum number of iterations, and output the nest of optimum fitness value.

3. BN Parameters Learning Method Based on CSA-CMOA.

3.1. **Methods described.** Each BN can be represented as a triple $< X, A, \Theta >$. $< X, A >$ defines a directed acyclic graph G, X is a node set, each node $X_i \in X$ represents a random variable within the modeling arena, and A is the collection of directed arcs between nodes. $\Theta = \{\theta_i\}$ is conditional probability parameter set, and $\theta_i = p\left(X_i \left| \prod (X_i) \right.\right)$ describes the conditional probability distribution of the node X_i on the parent node set $\prod (X_i)$. The graphic structure G qualitatively depicts the independent relationship between the random variable and conditional probability distribution quantifies the degree of dependence on its parent node, so BN uniquely determines joint probability distribution $P(X_1, \cdots, X_n) = \prod_{i=1}^n P(X_i \left| \prod (X_i) \right.)$ of random variable set $X = \{X_1, X_2, \cdots, X_n\}$ using graph structure and network parameters. If $X_i = k$ represents the ith node taking the kth value in network, $\prod (X_i) = j$ represents the jth value combination of the parent node set instantiation, N_{ijk} represents the number of $X_i = k$ and $\prod (X_i) = j$ occurrence at the same time, then conditional probability is $\theta_{ijk} = p\left(X_i = k \left| \prod (X_i) = j \right.\right) = N_{ijk} \Big/ \sum_k N_{ijk}$.

In this paper, we adopt the network composed of reason independence model as the research object. Assume that X_α is the αth parent node of X_β, $\lambda_{\alpha\beta}$ is the conditional probability from node X_α to node X_β. Using CSA-CMOA to learn BN parameters, the key is the construction of cuckoo individual model. The essence of BN parameters learning process is conditional probability $\lambda_{\alpha\beta}$ optimization process, so the cuckoo state can be expressed by all the conditional probability $\lambda_{\alpha\beta}$ of BN.

3.2. **Performance test.** Experiments are performed on the computer with Intel Core i3, 4 GB of memory, Windows 7 operating system, and implemented by JAVA programming. The training sample set is shown in Table 1, $(D_1, D_2, \ldots, D_{15})$ is input value of BN parameters learning, and D_{out} is output value of BN parameters learning.

The data in Table 2 is the average of MSE (mean square error) that CMOA, CSA and CSA-CMOA perform 50 times under the various iterations 5, 60, 160, 280, 400, 600. From Table 2 we can see, using CMOA, CSA and CSA-CMOA to learn the parameters of the BN is feasible. With the increasing of iterations, three methods are capable of achieving

Table 1. Sample set

D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{out}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.027684679544692
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.029687677559692
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0.135654684949433
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0.139835957677874
0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0.169939934349394
0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0.284734790473695
0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0.289393923430485
1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0.300237666574785
1	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0.312321334545221
0	0	1	1	1	0	0	0	0	0	0	1	1	0	0	0.405803729304305
0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0.422111236504083
1	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0.433223985989485
1	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0.444323409549923
1	1	0	1	0	0	0	0	0	1	1	0	1	0	1	0.582403042739475
1	1	0	0	1	0	0	0	0	0	1	1	0	1	0	0.593923042395894
0	0	1	1	0	1	0	0	1	1	0	0	0	1	1	0.588857303792023
1	1	1	0	1	1	0	1	1	1	1	1	0	0	1	0.734834758349334
1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	0.753375949457944
1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	0.838384884343966
1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	0.858584523662395
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.858594579495036
1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	0.866848586828345
1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0.904534349676069
1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	0.934565463454643
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0.945663469495869
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.984545459609485
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0.983455960003758
1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0.984586694560346
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.998998967997699

Table 2. The comparisons of MSE

iterations	CMOA	CSA	CSA-CMOA
5	0.8685439	0.3732498	0.3620021
60	0.7896889e-1	0.6554793e-2	0.5677322e-2
160	0.3580549e-2	0.6843898e-3	0.7838455e-4
280	0.5763456e-3	0.8934933e-4	0.9834345e-5
400	0.5657345e-3	0.3454506e-4	0.8364754e-5
600	0.4533267e-4	0.8744565e-5	0.9834685e-6

Table 3. The comparisons of three kinds of parameters learning methods MSE achieving e-3 magnitude

method	minimum number	maximum number	average number
CMOA	89	156	90.26
CSA	60	186	80.76
CSA-CMOA	20	106	50.99

the smaller MSE, but CSA-CMOA has minimum MSE and better convergence properties than the CMOA and CSA.

Three kinds of parameters learning methods respectively run 200 times, iterative minimum number, maximum number and average number of MSE achieving e-3 magnitude are shown in Table 3. From Table 3 we can see, the gap between minimum number and maximum number is big when CMOA and CSA achieve e-3 magnitude. This illustrates that the random motion results in the results shocking at optimal value and it is difficult to reach higher accuracy. CSA-CMOA is obviously better than the CMOA and CSA at the minimum number, maximum number and average number. This shows that CSA-CMOA has faster convergence speed and better ability of optimization and proves the advantages of using CSA-CMOA to learn BN parameters.

From the above analysis we can see that using our method to learn BN parameters can get smaller error, higher accuracy and fast speed.

4. Conclusions.

In this paper, we bring CSA-CMOA into BN parameters learning composed of reason independence model. We improve the convergence performance and speed of the algorithm by adjusting the random movement speed. The algorithm is simple, BN parameters could be obtained by only giving the status of the evidence node and goal node, and do not need to know the middle node status, so it can realize the parameters learning with hidden variables.

The next step of research work is to further improve the versatility of BN parameters learning method. In the process of BN parameters optimization we should consider the historical information influence on now, add the history information factors to the optimization rule, and then further improve the quality of optimization.

Acknowledgements. This work is supported by the Natural Science Foundation of Heilongjiang Province of China (F201333), the Ministry of Education Research of Humanities & Social Science Youth Fund in China (14YJC630188). The authors would like to thank the editors and reviewers of this manuscript for their very helpful suggestions.

REFERENCES

[1] M. Xiao and Y. Zhang, Parameters learning of Bayesian networks for multistate system with small sample, *Computer Science*, vol.42, no.4, pp.253-257, 2015.

[2] Y. Du, F. Shi and N. Yang, Construction method for Bayesian network based on Dempster-hafer/analytic hierarchy process, *Journal of Computer Applications*, vol.35, no.1, pp.140-151, 2015.

[3] M. Xiao and Y. Zhang, Parameters learning of Bayesian networds based on independence of causal influence model, *Control and Decision*, vol.30, no.6, pp.1008-1013, 2015.

[4] R. Di, X. Gao and Z. Guo, Parameter learning of discrete Bayesian network based on additive synergistic constraints, *Computer Simulation*, vol.31, no.10, pp.61-66, 2014.

[5] A. C. Constantinou, N. E. Fenton and M. A. Neil, Profiting from an inefficient association football gambling market: Prediction, risk and uncertainty using bayesian networks, *Knowledge-Based Systems*, vol.50, pp.60-86, 2013.

[6] I. C. Cárdenas, S. H. S. Al-Jibouri, J. I. M. Halman and F. A. van Tol, Modeling risk-related knowledge in tunneling projects, *Risk Analysis*, vol.34, no.2, pp.323-339, 2014.

[7] X. Yang and D. Suash, Cuckoo search via Lévy flights, *Proc. of World Congress on Nature and Biologically Inspired Computing*, India, pp.210-214, 2009.

[8] X. Yang and D. Suash, Engineering optimisation by cuckoo search, *Int'l Journal of Mathematical Modeling and Numerical Optimisation*, vol.1, no.4, pp.330-343, 2010.
[9] F. Wang, X. He and Y. Wang, The cuckoo search algorithm based on Gaussian disturbance, *Journal of Xi'an Polytechnic University*, vol.25, no.4, pp.566-569, 2011.
[10] S. Tan and W. Yu, New algorithm for global optimization: Cell membrane optimization, *Application Research of Computers*, vol.28, no.2, pp.455-457, 2011.
[11] C. Cao, F. Liu, D. Zhao and B. Zhang, Geometric constraint solving based on cell membrane optimization, *Journal of System Simulation*, vol.24, no.1, pp.120-122, 2012.