 JIOT, vol.1, no.2, pp.41-53, 2019

A New Method Based on Evolutionary Algorithm for Symbolic Network Weak Unbalance

Yirong Jiang[1], Weijin Jiang[2, 3, 4 *], Jiahui Chen[2, *], Yang Wang[2], Yuhui Xu[2], Lina Tan[2] and Liang Guo[5]

Abstract: The symbolic network adds the emotional information of the relationship, that is, the "+" and "-" information of the edge, which greatly enhances the modeling ability and has wide application in many fields. Weak unbalance is an important indicator to measure the network tension. This paper starts from the weak structural equilibrium theorem, and integrates the work of predecessors, and proposes the weak unbalanced algorithm EAWSB based on evolutionary algorithm. Experiments on the large symbolic networks Epinions, Slashdot and WikiElections show the effectiveness and efficiency of the proposed method. In EAWSB, this paper proposes a compression-based indirect representation method, which effectively reduces the size of the genotype space, thus making the algorithm search more complete and easier to get better solutions.

Keywords: Weak structural balance, signed networks, evolutionary algorithms, incremental computation, compressed representation.

Network is a general model of many complex systems. It represents the things in the system with nodes and the relations between things with edges. Starting from the emotional attributes of the side, the network can be divided into symbolic networks [Easley and Kleinberg (2019)] and unsigned networks. It is widely used in politics [Ghosn, Palmer and Bremier (2004)], society [Wasserman and Faust (1994)], biology [Parisien, Anderson and Eliasmith (2008)], e-commerce [Zolfaghar and Aghaie (2010)], cyberspace [Burke and Kraut (2008)], etc. applications.

Structural balance theory is the basic theory in symbolic networks. It was first proposed by Heider [Fritz (1946)] from the perspective of social psychology in the 1940s. Cartwright

[1] Tonghua Normal University, Tonghua, 134002, China.

[2] Key Laboratory of Hunan Province for New Retail Virtual Reality Technology, Hunan University of Commerce, Changsha, 410205, China.

[3] Institute of Big Data and Internet Innovation, Hunan University of Commerce, Changsha, 410205, China.

[4] School of Computer Science and Technology, Wuhan University of Technology, Wuhan, 430073, China.

[5] School of Bioinformatics, University of Minnesota, Twin Cities, USA.

[*] Corresponding Authors: Weijin Jiang. Email: jwj3666@163.com;
Jiahui Chen. Email: cjh810663304@163.com.

et al. [Cartwright and Harary (1956)] then redefined and expanded the theory in graph theory in the 1950s.

Arahona [Barahona (1999)] pointed out that solving the structural imbalance problem is an NP-hard problem. Terzi et al. [Terzi and Winkler (2011)] proposed a spectral method for solving the imbalance. Facchetti et al. [Facchetti, Iacono and Altafini (2011)] used the canonical transformation to give an efficient greedy algorithm for solving the imbalance. Chiang et al. [Chiang, Hsieh and Natarajan (2013)] used the Katz metric to find the number of negative loops and used this to measure the imbalance of the symbol network. Sun Yixiang et al. [Sun, Du and Gong (2014)] proposed a dense-mother algorithm for solving structural imbalances by using the characteristics of evolutionary algorithm global optimization.

In the 1960s, Davis [Davis (1977)] improved the structural balance theory. He believed that "the enemy of the enemy is a friend" is not necessarily correct in many occasions. His theory is called weak structural balance theory. Leskovec et al. [Leskovec, Huttenlocher and Kleinberg (2010)] have shown through experiments that weak structural equilibrium is more common than structural equilibrium in a large number of actual symbolic networks. However, it is usually not feasible to simply extend the method of solving structural imbalances such as Sun et al. [Sun, Du and Gong (2014)] to solve the weak structural imbalance. Earlier research on this problem was Doreian and Mrvar [Doreian and Mrvar (1996)]. In view of the good performance of evolutionary algorithms [Jong (2016)] in solving many NP-hard problems, and also inspired by the literature [Sun, Du and Gong (2014)], this paper proposes an evolutionary algorithm EAWSB for solving the weak imbalance of symbol networks. Experiments on large symbolic networks Epinions, Slashdot and WikiElections show that this method is effective and efficient.

1 Problem definition

1.1 Structural balance and weak structural balance

A symbolic network can be defined as a graph G(V, E, σ), where V and E are node sets and edge sets, respectively. The mapping σ: E→{+, −} defines the symbol properties of each edge. Fig. 1 is the four basic paradigms of the symbolic network. In the case of structural equilibrium, (a)(b) is balanced, (c)(d) is unbalanced. In the case of weak structural equilibrium, (a)(b)(d) is balanced and (c) is unbalanced.

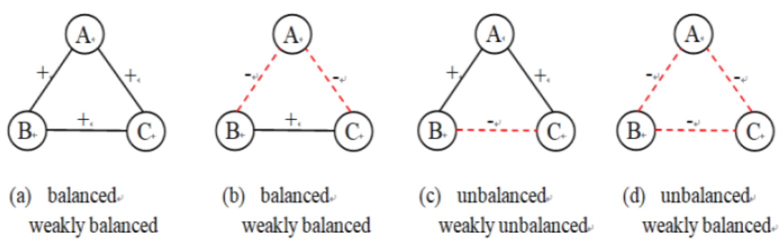

Figure 1: Four basic paradigms of signed networks

However, for the general symbolic network, the statistical method is no longer valid. At this time, its balance and weak balance are given by the following Theorem 1 [Cartwright and Harary (1956)] and Theorem 2 [Davis (1977)]. Theorem 1: A symbolic network is structurally balanced if and only if its node set can be divided into two classes and satisfy the following conditions: The edges in the same class are all positive, and the edges between different classes are all negative. Theorem 2: A symbolic network is weakly structurally balanced if and only if its node set can be divided into multiple classes and satisfy the following conditions: The edges in the same class are all positive, and the edges between different classes are all negative.

1.2 Calculation of structural balance and weak structural balance

By using Theorem 1 and Theorem 2, we can give another definition of (weak) unbalance. The nodes of a symbolic network are divided into several classes.

At present, the methods of seeking (weak) imbalance are mainly spectral methods [Terzi and Winkler (2011)], canonical transformation [Facchetti, Iacono and Altafini (2011)], Katz metric [Chiang, Hsieh and Natarajan (2013)], evolutionary algorithm [Sun, Du and Gong (2014)] and block model [Doreian and Mrvar (1996)]. The literature [Sun, Du and Gong (2014)] is also based on evolutionary algorithms, but it is only solved and discussed in a relatively small scale symbolic network and structural equilibrium case.

2 The Algorithm EAWSB

Considering the complexity of large-scale symbolic networks and the global optimization of evolutionary algorithms, combined with Theorem 2, this paper proposes an EAWSB (Evolutionary Algorithms for Weak Structural Balance) algorithm for solving the weak imbalance of symbol networks. The details are as follows.

2.1 Energy function and fitness function

According to Theorem 2, the energy function reflecting the weak imbalance can be defined as follows.

$$E(s) = \frac{1}{2} \sum_{(V_i, V_j) \in E} (1 - a_{ij} \delta(s_i, s_j)) \tag{1}$$

where $\delta(s_i, s_j)$ =1, if $s_i = s_j$, otherwise take -1. E(S) is the sum of the number of all negative edges in the same class and the number of all positive edges between different classes. The minimum value is the weak imbalance of the symbol network G.

The algorithm for defining the EAWSB is:

$$F(s) = \sum_{(V_i, V_j) \in E} a_{ij} \delta(s_i, s_j) \tag{2}$$

Since E(S)=(m-F(S))/2, minimizing E(S) is equivalent to maximizing F(S). In this case, if the maximum number of categories k is specified in advance, the optimization problem to be solved by the algorithm EAWSB is transformed into

$$\max F(s) = \max_{\substack{s_i \in \{0,1,\ldots,k-1\}, \\ i=1,2,\ldots,n}} \sum_{(V_i, V_j) \in E} a_{ij} \delta(s_i, s_j) \tag{3}$$

2.2 The natural representation and compression representation of the individual

The general individual representation is divided into direct representation and indirect representation [Jong (2016)].

For large symbolic networks, the value of n is too large, often tens of thousands, which seriously affects the performance of genetic operations and the overall algorithm [Liu, Meng, Ding et al. (2019)].

Theorem 3: Given the symbolic network G (V, E, _), suppose that A is the optimal solution of the optimization problem represented by (3), then for any I {1,... N}, there are all

$$s_i^* = \arg\max_{s_i \in \{0,\dots,k-1\}} \sum_{v_j \in N(v_i)} a_{ij}\delta(s_i, s_j) \tag{4}$$

The set $N(v_i) = \{v_k \mid (v_i, v_k) \in E\}$ is the neighborhood of v_i. It is proved that if suppose the existence of $h \in (1,\dots,n)$ does not satisfy the condition in the theorem, that is, $s_h^* \neq \arg\max_{s_h \in \{0,\dots,k-1\}} \sum_{v_j \in N(v_h)} a_{hj}\delta(s_h, s_j)$.

Because

$$\sum_{(v_i,v_j)\in E} a_{ij}\delta(s_i, s_j) = \sum_{(v_h,v_j)\in E} a_{hj}\delta(s_h, s_j) + \sum_{(v_i,v_j)\in E \wedge i\neq h} a_{ij}\delta(s_i, s_j)$$

$$= \sum_{v_j \in N(v_h)} a_{hj}\delta(s_h, s_j) + \sum_{(v_i,v_j)\in E \wedge i\neq h} a_{ij}\delta(s_i, s_j)$$

Note that h only appears in the first summation in the above formula, and does not appear in the second summation, so we can define $s_h^\# = \arg\max_{s_h \in \{0,\dots,k-1\}} \sum_{v_j \in N(v_h)} a_{hj}\delta(s_h, s_j)$, $s_j^\# = s_j^*$ for $j \neq h$, $j \in \{1,\dots,n\}$. This gives us a better solution than S*. Because

$$\sum_{(v_i,v_j)\in E} a_{ij}\delta(s_i^\#, s_j^\#) = \sum_{v_j \in N(v_h)} a_{hj}\delta(s_h^\#, s_j^\#) + \sum_{(v_i,v_j)\in E \wedge i\neq h} a_{ij}\delta(s_i^\#, s_j^\#)$$

$$> \sum_{v_j \in N(v_h)} a_{hj}\delta(s_h^*, s_j^*) + \sum_{(v_i,v_j)\in E \wedge i\neq h} a_{ij}\delta(s_i^\#, s_j^\#)$$

$$= \sum_{v_j \in N(v_h)} a_{hj}\delta(s_h^*, s_j^*) + \sum_{(v_i,v_j)\in E \wedge i\neq h} a_{ij}\delta(s_i^*, s_j^*)$$

$$= \sum_{(v_i,v_j)\in E} a_{ij}\delta(s_i^*, s_j^*)$$

Theorem 3 tells us the state of a node, that is, the optimal class value to which it belongs can be found by the state of its neighbor node using Eq. (4). So how do you find the dominating set U? Algorithm 1 gives a solution. Algorithm 1 generates a compressed representation:

1.Input: Adjacency matrix of symbol network G: $A=(a_{ij})_{n\times n}$
2. Calculate node degree array deg[0..n-1]
3. ori_deg[0..n-1]= deg[0.. n-1]
4. The value of the initialization array selNode[0..n-1] is 0.
5: for each i with ori_deg[i]=1 and deg[i]>0 do
6:     j=the neighbor of i
7:     selNode[j]=1, deg[j]=0
8:     for each j's neighbor p do
9:         if(deg[p]>0)deg[p]= deg[p]-1
10:     endfor
11: endfor
12: repeat
13. select a node with degree > 0 in roulette mode Randomly
14:     selNode[j]=1, deg[j]=0
15:     for each j's neighbor p do
16:         if(deg[p]>0)deg[p]= deg[p]-1
17:     endfor
18: until deg[i]=0 for alli∈{0.. n-1}
19: Output: All nodes i satisfying selNode[i]=1

Algorithm 1 consists of 3 parts. Part 1 (lines 2-4) defines three arrays, ori_deg and deg, which hold the degree information of nodes exactly the same at the beginning. Part 2 (lines 5-11) handles leaf nodes (i.e., nodes with degree 1). The leaf node has a unique neighbor node. Part 3 (lines 12-18) uses a degree ratio selection strategy to select a node, i.e., the probability that a node is selected is the sum of the degrees of a node divided by the degrees of all nodes.

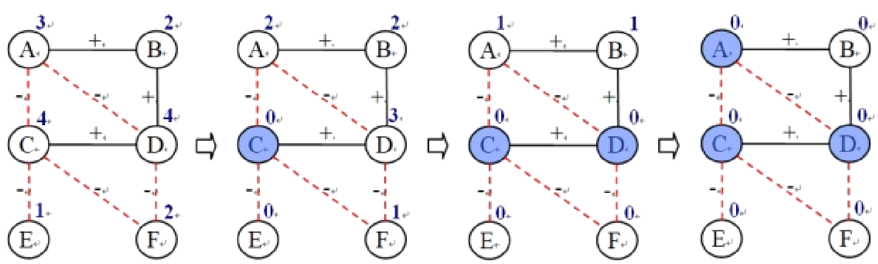

Figure 2: Illustration of Algorithm 1 (The number in the upper right corner of vertex i indicatesdeg [i])

Fig. 2 is an example of algorithm 1 generating compression coding. E is a leaf node that is generally not selected, but its neighbor nodes must be elected to the dominating set. The last three nodes A, C, and D are selected to dominate the set U. The individual compression code is ind_c=sAsCsD, the natural code is ind=sAsBsCsDsEsF, and the compression ratio is 50%.

2.3 Population initialization

The theory of homogeneity [Easley and Kleinberg (2019)] tells us that We will become more similar to our friends. The above selection-assignment process is repeated iniK times. Where iniK is a positive integer representing the initialization strength. The time complexity of population initialization is O(iniK*davg).

2.3.1 Genetic operator

1) cross

This paper uses the one-way crossover operator proposed by Tasgin et al. [Tasgin, Herdagdelen and Bingol (2006)]. The main idea is as follows. Find all the nodes in ind1 whose category value is s, change the category values of these nodes to s in ind2, and return the modified ind2.

2) variation

In this paper, a single point mutation is used to randomly select a node on the individual to be mutated and assign it a new category value. The time complexity of the mutation is O(1).

3) Choice

This paper adopts the league selection of league size 2 [Li, Kou and Lin (2002)], and adopts the elite retention strategy [Li, Kou and Lin (2002)]. The time complexity of the selection is O(1).

4) Rotation

The value of each gene of each individual is $\{0,...,k-1\}$. In the evolutionary process, each individual rotates with a small probability (generally 0.05 in this paper), i.e., the class value is $0 \rightarrow, 1 \rightarrow 2, ..., k-1 \rightarrow 0$.

2.3.2 Local search

Starting from Theorem 3, the local search can be designed as follows: For a given individual ind, a node vi on it is randomly selected, and the state of the node is modified.

$$s_i = \arg\max_{s_i \in \{0,...,k-1\}} \sum_{v_j \in N(v_i)} a_{ij} \delta(s_i, s_j)$$

2.3.3 Incremental calculation of fitness values

Eq. (2) can be used to directly calculate the individual's fitness value, but for large networks, the amount of calculation is large because the length of the individual is the number of network nodes.

Algorithm 2 Incremental calculation of fitness values after individual variation

1: Input: Current individual: ind, mutation position: h, the value of the hth gene sh before mutation: clsOld, the value of sh after mutation: clsNew

2: delta=0 //The fitness value increment is initialized to 0

3: For each neighbor j of vertex at h

4: if(J is in the dominating set)
5: clsNbr=class label at position j in ind
6: if (clsNbr=clsOld) delta=delta-2*a_{hj}
7: else if(clsNbr=clsNew) delta=delta+2*a_{hj}
8: }
9: else if (j in the degradation concentration){
10: maxEnergyNew=Before mutation maxEnergy(v_j)
11: maxEnergyOld=After mutation maxEnergy(v_j)
12: delta = maxEnergyNew-maxEnergyOld
13: }
14: Endfor
15: Output: Adaptation value of ind before mutation+delta

$$\max Energy(v_h) = \max_{s_h \in \{0,\ldots k-1\}} \sum_{v_j \in \mathrm{N}(v_h)} a_{hj} \delta(s_h, s_j)$$

2.3.4 EAWSB algorithm framework

Algorithm 3 is the overall framework of the EAWSB algorithm.

Algorithm 3 EAWSB algorithm framework

1: Input: G(V, E) adjacency matrix: A=(aij)n×n, population size: popSize, initialization strength: iniK, local search strength: locK, maximum evolution algebra: maxGen, league size: tourSize, Crossover probability: pc, probability of variation: pm, rotation probability: pr
2: Use natural representation or generate a compressed representation (A)
3: P←population initialization (popSize, iniK)
4: repeat
5: Pparent←select (P, tourSize)
6: Pchild←cross (Pparent,p_c
7: Pchild←variation (Pchild, p_m)
8: Pchild←rotation (Pchild, p_r)
9: P←local search (Pchild, locK)
10: Until the evolution termination condition (maxGen) is met
11: Output: the best individual in P

3 Expeeriments analysis

3.1 EAWSB algorithm composition

To be exact, EAWSB is a cluster of algorithms, which consists of EAWSB_N, EAWSB_I, EAWSB_C and EAWSB_IC. They have the same function, but have different performance in different occasions. The difference between them is shown in Tab. 1.

Table 1: Four constituents of EAWSB

	EAWSB_N	EAWSB_I	EAWSB_C	EAWSB_IC
Incremental calculation	×	√	×	√
Compressed representation	×	×	√	√

3.2 Experimental environment

Tab. 2 is the experimental environment of Algorithm EAWSB in this paper.

Table 2: Experimental environment of EAWSB

Hardware environment	Lenovo laptop savior e520, Quad-core processor, logical eight core, 16 G memory.
Operating system	Microsoft Windows [version 10.0.15063]
Development environment	java version "1.7.0_15" Java(TM) SE Runtime Environment (build 1.7.0_15-b03)

3.3 Data set

This article was conducted on three large symbolic network datasets, Epinions, Slashdot, and WikiElections. Epinions (epinions.com) is a product review website [Guha, Kumar and Raghavan (2004)]. Slashdot (slashdot.com) is a technology news site [Jérôme, Lommatzsch and Bauckhage (2009)] that allows users to mark authors as "friends" or "enemies" for other users'articles, forming a network of friends/enemies. WikiElections [Leskovec, Huttenlocher and Kleinberg (2010)] is a dataset for Wikipedia users voting for elections. It is a support or objection network. Tab. 3 is the original case of the three data sets. The experiment is mainly carried out on the large undirected symbolic network shown in Tab. 4.

Table 3: Original datasets

Raw data set	Number of nodes	Number of sides	Description
soc-sign-epini ons	131,828	841,372	Epinions Symbolic network
soc-sign-Slash dot090221	82,144	549,202	Slashdot Zoo Symbolic network February 21, 2009，Snapshot
wiki-Elec	8,297	103,591	Wikipedia Administrator election symbol network

Table 4: Preprocessed datasets

Experimental data set	Number of nodes	Number of sides
Epinions	131,513	708,507
Slashdot	82,062	498,532
WikiElections	7,114	99892

3.4 Operation results and running time

3.4.1 Parameter setting

For all algorithms, all data sets, EAWSB parameter settings. The results are as follows: population size popSize=500, initialization strength iniK=500, local search strength locK=500, maxGen=500, tourSize=2, crossover probability PC=0.8, mutation probability PM=0.1, rotation probability PR=0.05.

3.4.2 Operation results

Figs. 3-5 show the results of the four algorithms EAWSB_N, EAWSB_I, EAWSB_C, and EAWSB_IC on Epinions, Slashdot, and WikiElections. The operation is performed in five cases according to the number of categories k=2, 3, 4, 5, and 6, where k=2 is a structural equilibrium situation, which can be regarded as a special case of weak structural balance.

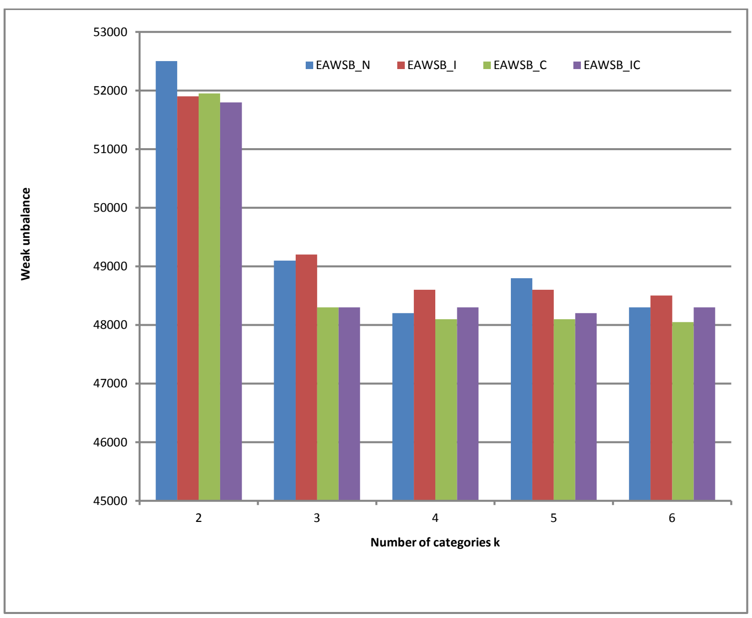

Figure 3: Results of the four algorithms on Epinions

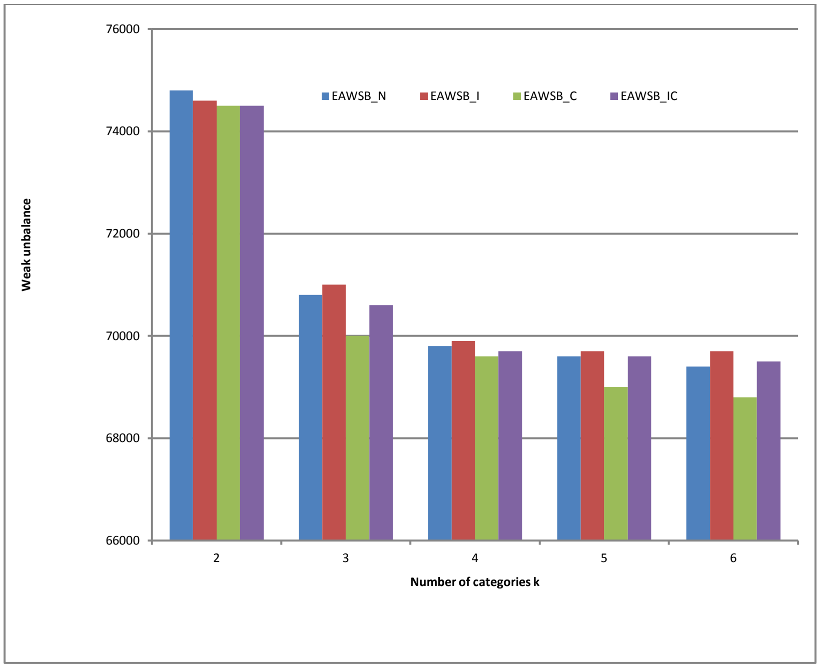

Figure 4: Results of the four algorithms on Slashdot

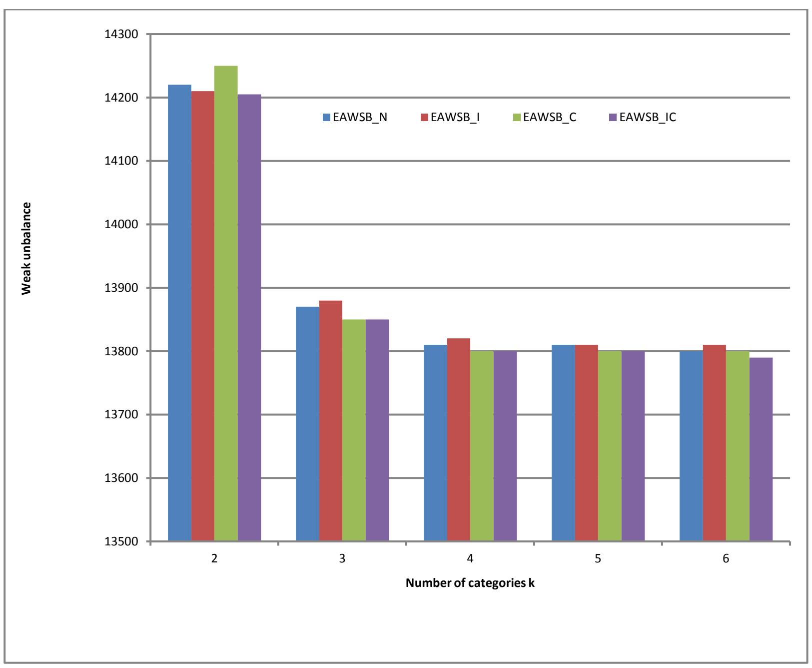

Figure 5: Results of the four algorithms on WikiElections

3.5 Performance comparison with similar algorithms

Meme-sb [Sun, Du and Gong (2014)] is a structural unbalanced algorithm based on the timid algorithm. Tab. 5 and Tab. 6 show the experimental results and running time of EAWSB_I and meme-sb u shows nder three large-scale symbolic network datasets in the number of categories k=2~6. Experiments show that EAWSB_I is significantly better than meme-sb on the two large datasets of Epinions and Slashdot. In addition, meme-sb is slightly better than EAWSB_I on WikiElections. Because meme-sb has a large "tearing" negative impact, the one-way crossover used by EAWSB_I is easier to maintain the integrity of the building block than the 2-point crossover used by meme-sb.

Table 5: Comparisons between experimental results of EAWSB_I and meme-sb

	Epinions		Slashdot		WikiElections	
k	EAWSB_I	meme-sb	EAWSB_I	meme-sb	EAWSB_I	meme-sb
2	51867	56544.5	74634.4	76334.33	14220.6	14204.67
3	49288.8	60851	70661.6	75021.67	13870.8	13858
4	48637.8	58854.33	69699.2	75299.67	13824.6	13809
5	48625	58628	69382.4	75465.67	13814.2	13793
6	48535.6	56799	69260.2	77196.67	13815	13805.33

Table 6: Comparisons between running time of EAWSB_I and meme_sb (s)

	Epinions		Slashdot		WikiElections	
k	EAWSB_I	meme-sb	EAWSB_I	meme-sb	EAWSB_I	meme-sb
2	1282.295	4191.39	754.2902	2549.973	131.527	306.6153
3	996.9442	4211.654	598.2798	2574.373	108.358	302.8893
4	811.7326	3807.2	487.3583	2580.679	97.3256	305.7383
5	706.6864	3814.085	433.6314	22537.29	93.458	344.1983
6	681.7204	3839.202	491.775	763.142	101.6392	342.976

4 Conclusion

Weak unbalance is an important indicator to measure the tension of the network [Hou, Wei, Wang et al. (2018)]. This paper starts from the weak structural equilibrium theorem, and integrates the work of predecessors, and proposes the weak unbalanced algorithm EAWSB based on evolutionary algorithm. Experiments on large symbolic networks Epinions, Slashdot, and Wiki Elections demonstrate the effectiveness and efficiency of this approach. In EAWSB, this paper proposes a compression-based individual indirect representation method, which effectively reduces the size of the genotype space, thus making the algorithm search more complete and easier to get a better solution. In this paper, an incremental fitness calculation method is proposed, which reduces the time complexity of fitness calculation from O (n) to O (d_{avg}), and greatly improves the efficiency of the algorithm.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (61772196; 61472136), the Hunan Provincial Focus Social Science Fund (2016ZDB006), Hunan Provincial Social Science Achievement Review Committee results appraisal identification project (Xiang Social Assessment 2016JD05), Key Project of Hunan Provincial Social Science Achievement Review Committee (XSP 19ZD1005). The authors gratefully acknowledge the financial support provided by the Key Laboratory of Hunan Province for New Retail Virtual Reality Technology (2017TP1026).

References

Burke, M.; Kraut, R. (2008): Mopping up: modeling wikipedia promotion decisions. *ACM Conference on Computer Supported Cooperative Work.*

Barahona, F. (1999): On the computational complexity of Ising spin glass models. *Journal of Physics A General Physics*, vol. 15, no. 10, pp. 3241-3251.

Cartwright, D.; Harary, F. (1956): Structural balance: a generalization of Heider's theory. *Social Networks*, vol. 63, no. 5, pp. 277-293.

Chiang, K. Y.; Hsieh, C. J.; Natarajan, N. (2013): Prediction and clustering in signed networks: a local to global perspective. *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1177-1213.

Davis, J. A. (1977): Clustering and structural balance in graphs. *Social Networks*, vol. 20, no. 2, pp. 27-33.

Doreian, P.; Mrvar, A. (1996): A partitioning approach to structural balance. *Social Networks*, vol. 18, no. 2, pp. 149-168.

Doreian, P.; Mrvar, A. (2009): Partitioning signed social networks. *Social Networks*, vol. 31, no. 1, pp. 1-11.

Easley, D.; Kleinberg, J. (2019): *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. New York: Cambridge University Press.

Facchetti, G.; Iacono, G.; Altafini, C. (2011): Computing global structural balance in large-scale signed social networks. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 52, pp. 20953-20958.

Fritz, H. (1946): Attitudes and Cognitive Organization. *Journal of Psychology*, vol. 21, no. 1, pp. 107-112.

Ghosn, F.; Palmer, G.; Bremier, S. A. (2004): The MID3 data set 1993-2001: procedures, coding rules, and description. *Conflict Management and Peace Science*, vol. 21, no. 2, 133-154.

Guha, R.; Kumar, R.; Raghavan, P. (2004): Propagation of trust and distrust. *International Conference on World Wide Web.*

Harary, F. A. (1961): Structural analysis of the situation in the Middle Eastin 1956. *Journal of Conflict Resolution*, vol. 10, no. 5, pp. 167-178.

Hou, M. W.; Wei, R.; Wang, T. G.; Cheng, Y.; Qian, B. Y. (2018): Reliable medical recommendation based on privacy-preserving collaborative filtering. *Computers, Materials & Continua*, vol. 56, no. 1, pp. 137-149.

Jong, K. A. D. (2016): *Evolutionary Computation: A Unified Approach*. MIT Press.

Jérôme, K.; Lommatzsch A.; Bauckhage, C. (2009): The slashdot zoo: mining a social network with negative edge. *Proceedings of the 18th International Conference on World Wide Web*, pp. 741-750.

Kunegis, J.; Preusse, J.; Schwagereit, F. (2013): What is the added value of negative links in online social networks. *Proceedings of the 22nd international conference on World Wide Web*, vol. 10, no. 9, pp. 727-736.

Liu, G. S.; Meng, K.; Ding, J. C.; Nees, J. P.; Guo, H. G. et al. (2019): An entity-association-based matrix factorization recommendation algorithm. *Computers, Materials & Continua*, vol. 58, no. 1, pp. 101-120.

Liu, Y.; Yang, Z.; Yan, X. Y.; Liu, G. C.; Hu, B. (2019): A novel multi-hop algorithm for wireless network with unevenly distributed nodes. *Computers, Materials & Continua*, vol. 58, no. 1, pp. 79-100.

Parisien, C.; Anderson, C. H.; Eliasmith, C. (2008): Solving the problem of negative synaptic weights in cortical models. *Neural Computation*, vol. 20, no. 6, pp. 1473-1494.

Sun, Y.; Du, H.; Gong, M. (2014): Fast computing global structural balance in signed networks based on memetic algorithm. *Physica A Statistical Mechanics & Its Applications*, vol. 415, no. 415, pp. 261-272.

Terzi, E.; Winkler, M. (2011): A spectral algorithm for computing social balance. *International Conference on Algorithms and MODELS for the Web Graph.*

Wasserman, S.; Faust, K. (1994): *Social Networks Analysis: Methods and Applications*.

Cambridge University Press.

Zheng, X. L.; Wang, F. Y. (2015): Social balance in signed networks. *Information Systems Frontiers*, vol. 17, no. 5, pp. 1077-1095.

Zolfaghar, K.; Aghaie, A. (2010): Mining trust and distrust relationships in social Web applications. *IEEE International Conference on Intelligent Computer Communication and Processing.*