Knowledge-Based Systems 85 (2015) 196–209

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A memetic algorithm for computing and transforming structural balance in signed networks

Lijia Ma [a], Maoguo Gong [a,*], Haifeng Du [b], Bo Shen [a], Licheng Jiao [a]

[a] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province 710071, China
[b] Center for Administration and Complexity Science, Xi'an Jiaotong University, Xi'an, Shaanxi Province 710049, China

ARTICLE INFO

Article history:
Received 29 October 2014
Received in revised form 10 April 2015
Accepted 5 May 2015
Available online 12 May 2015

Keywords:
Structural balance
Balance computation
Balance transformation
Memetic algorithm
Signed networks

ABSTRACT

Structural balance enables a comprehensive understanding of the potential tensions and conflicts of signed networks, and its computation and transformation have attracted increasing attention in recent years. The balance computation aims at evaluating the distance from an unbalanced network to a balanced one, and the balance transformation is to convert an unbalanced network into a balanced one. In this paper, firstly, we model the balance computation of signed networks as the optimization of an energy function. Secondly, we model the balance transformation as the optimization of a more general energy function incorporated with transformation cost. Finally, a multilevel learning based memetic algorithm, which incorporates network-specific knowledge such as the neighborhoods of node, cluster and partition, is proposed to solve the modeled optimization problems. Systematical experiments in real-world social networks demonstrate the superior performance of the proposed algorithm compared with the state-of-the-art algorithms on the computation and transformation of structural balance. The results also show that our method can resolve the potential conflicts of signed networks with the minimum cost.

1. Introduction

Social interaction involves friendly and hostile relationships in many complex systems, including multi-player online games, Wikipedia, Web 2.0, online communities and information recommendation systems [1–3]. In these systems, the entities with friendly relationships are friends, cooperators, alliances or memberships in a group, while those with hostile relationships are enemies, competitors, opponents or memberships in different groups. These systems can be represented as signed networks in which nodes represent social entities and positive/negative edges correspond to friendly/hostile relationships.

Structural balance, one of the most popular properties in ensembles of signed networks, reflects the origin of tensions and conflicts, and its computation and transformation have received much attention from physicist, sociologist, economist, ecologist, ecologist and mathematician [4–10]. The structural balance theory introduced by Heider states that the relations "the friend of my friend is my enemy" and "the enemy of my enemy is my enemy" are unbalanced in the strong definition, which is based on the statistical analysis of the balance of signed triads from the perspective of social psychology [11]. There are broad applications of the computation of structural balance, due to the ubiquity of the multi-relational organization of modern systems in a variety of disciplines. Furthermore, the pursuit of balance is desirable in many real-world signed systems. For instance, the pursuit of balance in international relationship networks can reduce military, economic and culture conflicts. The pursuit of balance in information systems can improve the authenticity of collected information and accelerate opinion diffusion [12–14].

The computation of structural balance aims at calculating the least imbalances of signed networks [7]. There are mainly two issues in the computation of structural balance: (i) how to verify imbalances and (ii) how to compute the least imbalances. There have been recent efforts in addressing these issues. The frustration index [5] evaluates the imbalances of signed networks as the number of negative cycles which have an odd number of negative links. The energy function proposed by Facchetti et al. [7,15] measures the imbalances as the number of unbalanced links in signed networks, and both the gauge transformation [7,15] and memetic algorithms [16] are utilized to minimize the energy function. Besides, some studies evaluate the imbalances as the sum of the positive links across two different communities and the negative

* Corresponding author.
E-mail address: gong@ieee.org (M. Gong).

http://dx.doi.org/10.1016/j.knosys.2015.05.006
0950-7051/

links within the same community [15,17], and thereafter they adopt the classical community detection algorithms, such as Infomap [18], FEC [19] and MODPSO [20], to identify the communities in signed networks. Note that, in many cases these studies are constraint to the strong definition of structural balance, and they are worth to be applied to the weak definition of structural balance. In the weak definition, "the friend of my friend is my enemy" is unbalanced while "the enemy of my enemy is my enemy" is balanced.

The transformation of structural balance focuses on how to convert an unbalanced network into a balanced one. Classical transformation models are divided into two categories: discrete-time dynamic models and continuous-time dynamic models [21]. Local triad dynamic and constrained triad dynamic are two representative discrete-time models [22,23]. With these models, an unbalanced network is finally evolved towards a balanced or a jammed state by changing signs of edges [5]. Another classical discrete-time model is based on transforming the least number of unbalanced links in signed networks [24,25]. Differential dynamic presented by Kulakowski et al. [26] is a classical continuous-time model, and it exhibits the evolutionary process from an unbalanced network to a balanced one. In this model, the final evolutionary state (i.e., conflict or harmony) of social networks is determined by the total amount of positive links [8]. These models can effectively transform an unbalanced network into a balanced one, and they are worth improving if they take into account the cost of balance transformation.

Recent studies have demonstrated that the computation and transformation of structural balance in signed networks are non-deterministic polynomial-time hard (NP-hard) problems [7,27]. This is of particular concern since the solution space increases with the size of a signed network exponentially. Memetic algorithms (MAs) as hybrid global–local search methodologies are widely adopted to solve NP-hard problems [28]. In general, a global search benefits explorations while a local search facilitates exploitations [28]. MAs synthesize the complementary advantages of global and local search methodologies [28–31], which makes it possible to solve large-scale NP-hard optimization problems. MAs have been applied into some large-scale networks optimizations, including balance computation [16], community detection [32,33], network resource allocation [34], neural network design [35], network robustness improvement [36] and network prediction [37–39].

In this study, we propose a fast memetic algorithm, referred to as MLMSB, to compute and transform structural balance of signed networks. First, we extend the energy function criterion [7] to the weak definition of structural balance, and model the computation of structural balance as the optimization of the extended energy function H. Second, by introducing a cost coefficient parameter, we present a more general energy function H_w to evaluate the balance transformation cost, and model the transformation of structural balance as the optimization of H_w. Finally, a multilevel learning based local search is integrated into a population-based genetic algorithm (GA) to solve the modeled optimization problems. For the proposed MLMSB algorithm to converge fast, we make full use of the network-specific knowledge such as the interactions of nodes, clusters and partitions. Experimental results on five real-world networks demonstrate that MLMSB outperforms the classical algorithms that compute and transform the structural imbalance of signed network. Furthermore, the results also illustrate that MLMSB can identify and resolve potential conflicts with the minimum transformation cost.

The rest of this paper is organized as follows. Section 2 briefly introduces ground knowledge about structural balance of signed networks. The optimization models and our method MLMSB are detailed in Section 3, followed by systematical empirical results in Section 4. The conclusions are summarized in Section 5.

2. Related backgrounds

A signed network with N nodes can be represented as an adjacency matrix J, with each item J_{ij} defined as follows.

$$J_{ij} = \begin{cases} +1 & \text{if nodes } v_i \text{ and } v_j \text{ are positively linked,} \\ -1 & \text{if nodes } v_i \text{ and } v_j \text{ are negatively linked,} \\ 0 & \text{if nodes } v_i \text{ and } v_j \text{ have no link.} \end{cases} \quad (1)$$

2.1. Structural balance

Structure balance theory investigates the balance of three interconnected individuals. Assuming that an edge labeled $+$ $(-)$ indicates a friendly (hostile) relationship between the corresponding two individuals, there are the following four types of social relations [11]:

- $+++$: "the friend of my friend is my friend";
- $++-$: "the friend of my friend is my enemy";
- $--+$: "the enemy of my enemy is my friend";
- $---$: "the enemy of my enemy is my enemy".

Structural balance theory in its strong definition claims that the social relations $+++$ and $--+$ are balanced while the relations $++-$ and $---$ are unbalanced. Structural balance theory in its weak definition illustrates that the social relations $+++$, $--+$ and $---$ are balanced while the relation $++-$ is unbalanced. Fig. 1 illustrates the differences between the strong and the weak definitions of structural balance.

Structural balance theory states that a complete signed network is balanced if and only if all its signed triads are balanced. For a non-complete signed network, it is balanced if it can be filled into a complete network by adding edges in such a way that the resulting complete network is balanced [1]. Wasserman gives an equivalent theorem from the perspective of clustering as follows [40].

The strong (weak) definition of structural balance: "A signed network G is completely balanced, if and only if its nodes can be divided into $k = 2$ $(k \geqslant 2)$ clusters such that the nodes are positively linked within each cluster whereas negatively linked between the clusters" [40].

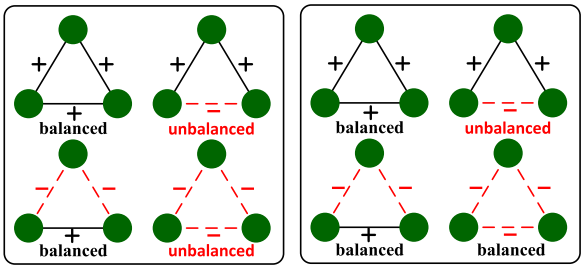
According to the equivalent balance theorem, the social relation $---$ is unbalanced (balanced) when its individuals are divided into $k = 2$ $(k = 3)$ clusters.

2.2. Energy function

The energy function [7] measures the imbalances of signed networks as the number of unbalanced links, and it is computed as follows.

$$h = \sum_{(i,j)} \frac{(1 - J_{ij}x_ix_j)}{2}, \quad (2)$$

where $J_{ij} \in \{+1, -1\}$ represents the sign of the edge between nodes v_i and v_j: $+1$ denotes a positive link, whereas -1 indicates a negative link; $x_i \in \{+1, -1\}$ is the identifier of cluster s_i. It is noteworthy that if two friendly nodes are assigned to different clusters, $(1 - J_{ij}x_ix_j)/2 = 1$, and $(1 - J_{ij}x_ix_j)/2 = 0$ otherwise. If two hostile nodes are assigned to different clusters, $(1 - J_{ij}x_ix_j)/2 = 0$, and $(1 - J_{ij}x_ix_j)/2 = 1$ otherwise. In other words, the link between nodes v_i and v_j is unbalanced when its corresponding $(1 - J_{ij}x_ix_j)/2$ value



(a) (b)

Fig. 1. Illustration of the structural balance theorem in (a) the strong definition and (b) the weak definition. A solid line represents a positive relationship (+) and a dash line indicates a negative relationship (−).

is 1. The minimization of h is to find the minimum number of unbalanced links by dividing the nodes of networks into two clusters, namely "+1" or "−1" [7]. Fig. 2 illustrates the identified unbalanced links of a toy network G_1 by optimizing h.

3. Models and methods

3.1. Optimization models

3.1.1. Optimization model for the computation of structural balance in signed networks

The computation of structural balance of signed networks in the weak definition can be modeled as follows.

$$\min\ H = \sum_{(i,j)} \left[\frac{(1 - J_{ij}\Theta(x_i, x_j))}{2}\right], \tag{3}$$

where H is the extended energy function and $\Theta(x_i, x_j)$ is a sign function whose value is 1 if $x_i = x_j$ and −1 otherwise. Here, the x_i value is an integer value in the range of $[1, N]$, where N is the number of nodes. The minimization of H is to find the minimum number of unbalanced links by dividing the nodes of a signed network into $k \geqslant 2$ clusters.

It is worthwhile to note that the energy function in Eq. (2) is a special case of H with k fixed at 2, and the function H can cater both the strong and weak definitions of the structural balance of signed networks. For instance, the social relation − − − can be classified as an unbalanced (balanced) structure with $H = 1$ ($H = 0$) by dividing the nodes into two (three) clusters. It is necessary to set $k = 2$ in advance when we use H to compute the structural balance of signed networks in the strong definition. An illustration of the computation of structural balance on a toy network G_2 is shown in Fig. 4.

3.1.2. Optimization model for structural balance transformation in signed networks

The unbalanced entities and connections of complex systems are the potential threats to their functionality. For complex systems to improve the functional security, it is necessary to alleviate their potential imbalances.

The extended energy function H can be further expressed as follows.

$$\begin{aligned} H &= -\sum_{(i,j),x_j \neq x_i} \left[\frac{J_{ij}^{+}\Theta(x_i, x_j)}{2}\right] - \sum_{(i,p),x_p = x_i} \left[\frac{J_{ip}^{-}\Theta(x_i, x_p)}{2}\right] \\ &= \sum_{(i,j),x_j \neq x_i} |J_{ij}^{+}|/2 + \sum_{(i,p),x_p = x_i} |J_{ip}^{-}|/2, \end{aligned} \tag{4}$$

where J_{ij}^{-} (J_{ij}^{+}) represents the negative (positive) links between nodes v_i and v_j. $\sum_{(i,p),x_i = x_p} |J_{ip}^{-}|/2$ and $\sum_{(i,j),x_j \neq x_i} |J_{ij}^{+}|/2$ indicate the number of the negative links within the same cluster and the positive links across different clusters, respectively.

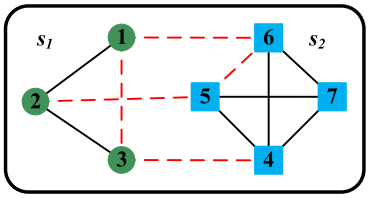


Fig. 2. Illustration of an unbalanced network G_1 with $h = 2$. G_1 is divided into two clusters by optimizing h. The negative links within the same cluster are unbalanced.

Algorithm 1. Framework of the proposed MLMSB

1: **Input**: The cost coefficient w; the population size: N_P; the size of mating pool: N_m; the tournament size: N_o; the crossover probability: p_c; the mutation probability: p_m; the maximum number of generations: g_{max}.
2: **Step 1.** *Initialization*: Adopt the initialization described in Section 3.2.3 to generate a population of initial solutions $\mathbf{X}(0) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_P}\}$; Set $t = 0$;
3: **Step 2.** *Selection*: Use the tournament selection described in Section 3.2.4 to choose N_m solutions from $\mathbf{X}(t)$ as parent solutions $\mathbf{Y}(t)$;
4: **Step 3.** *Genetic operation*: Employ the network-specific genetic operation described in Section 3.2.5 on $\mathbf{Y}(t)$ to evolve towards offspring solutions $\mathbf{Z}(t)$;
5: **Step 4.** *Local search*: Employ the node, cluster, and partition learning techniques described in Section 3.2.6 on the best solution $\mathbf{x}_l(t)$ of $\mathbf{Z}(t)$ to find an optimal solution $\mathbf{x}_l'(t)$ around the region generated by $\mathbf{x}_l(t)$.
6: **Step 5.** *Update*: Get the next population $\mathbf{X}(t+1)$ by choosing N_P solutions with low H_w from $\mathbf{X}(t), \mathbf{Z}(t)$ and $\mathbf{x}_l'(t)$.
7: **Step 6.** *Stop criteria*: If $t \geqslant g_{max}$, stop the algorithm and output $\mathbf{X}(t+1)$; otherwise $t = t + 1$ and go to **Step2**.
8: **Output**: Covert $\mathbf{X}(t+1)$ into the clustering divisions of networks and output the negative edges within clusters and positive edges across clusters.

A balanced network will be achieved if the signs of the unbalanced links found by the optimization of H are changed.

In reality, the cost of changing a positive link may be different from that of a negative link. Thus, a more general energy function H_w is introduced with cost coefficient parameter w evaluating the different cost of changing positive and negative links for balance transformation. The corresponding optimization model for the balance transformation with minimum cost is as follows:

$$\min\ H_w = w \cdot \sum_{(i,j),x_j \neq x_i} |J_{ij}^{+}|/2 + (1 - w) \cdot \sum_{(i,p),x_p = x_i} |J_{ip}^{-}|/2, \tag{5}$$

where $w \in \{0, 1\}$ evaluates the emphasis on either positive links or negative ones. If $0 \leqslant w < 0.5$, changing a negative link is more costly compared with changing a positive one; whereas if $0.5 < w \leqslant 1$, positive links affect more. An illustration of balance transformation on the toy network G_2 is shown in Fig. 3.

3.2. Memetic algorithm for the computation and transformation of structural balance in signed networks

Our algorithm MLMSB is first initialized with a population of solutions, each of which represents a possible balance

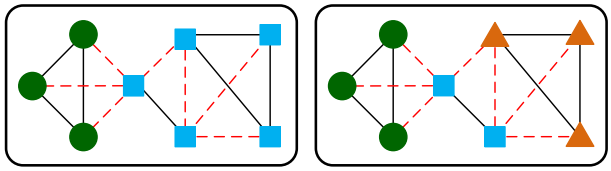


Fig. 3. Two balance transformations on the toy network G_2 under different values of w. The unbalanced toy network is transformed into a balanced one at minimum cost under (a) $w = 0.9$ and (b) $w = 0.1$. For the unbalanced toy network to be balanced, three negative edges in (a) and one positive edge in (b) are necessary to be changed of signs.

transformation. The population of solutions is then evolved via global search conducted by Genetic Algorithm (GA) and a network-specific local search iteratively. The network-specific knowledge, such as the neighborhoods of node, cluster and partition, is incorporated into the GA and local search. The framework of MLMSB is listed in Algorithm 1. The rest of this section will detail the definitions of neighborhoods and the representation, initialization, tournament selection, genetic operation and network-specific local search.

3.2.1. Definitions of neighborhoods

A signed network can be viewed as a microscopic graph $G = \{V, E\}$, with a set of nodes $V = \{v_1, v_2, \ldots, v_N\}$ and positive/negative links E. And the links between nodes can be represented as an adjacency matrix J as shown in Eq. (1).

In the meantime, from a macroscopic viewpoint, a signed network can also be divided into a set of clusters $S = \{s_1, s_2, \ldots, s_k\}$ with positive and negative connections E' between clusters, in summary $G' = \{S, E'\}$. Here, weighted adjacency matrices A^+ and A^- are used to measure the effect of the positive and negative links, respectively, and their items are defined as follows: $A^+_{ij} = \sum_{u \in s_i} \sum_{q \in s_j} J^+_{uq}$ and $A^-_{ij} = \sum_{u \in s_i} \sum_{q \in s_j} J^-_{uq}$. Especially, A^+_{ii}/A^-_{ii} indicates the number of internal positive/negative links within the cluster s_i; and A^+_{ij}/A^-_{ij} $(j \neq i)$ is the number of external positive/negative links between clusters s_i and s_j.

Fig. 5 demonstrates both the microscopic and the macroscopic viewpoints of a toy network.

Based on the knowledge that negative links have little influences on the formation of communities [17] and that the final balance state of signed networks is determined by the number of positive links [8], we define the neighborhoods of node, cluster and partition as follows.

Definition of the neighborhood of node. The neighborhood of a node v_i is a set of nodes that have a positive link with v_i.

Definition of the neighborhood of cluster. The neighborhood of a cluster s_i is a set of clusters that have external positive links with s_i.

Definition of the neighborhood of partition. The neighborhood of a partition S_i is a set of partitions that can be transformed into S_i by merging any of their two neighboring nodes or clusters.

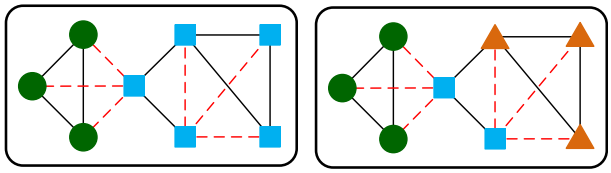


Fig. 4. Illustration of the computation of structural balance on a toy network G_2. G_2 is divided into (a) two clusters with $H = 4$ and (b) three clusters with $H = 0$.

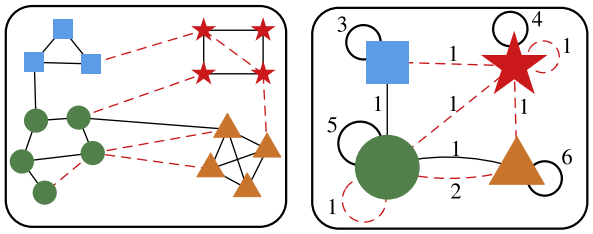


Fig. 5. Illustration of the representations of the toy network G_3 under the microscopic and the macroscopic viewpoints. (a) In a microscopic viewpoint, G_3 can be represented by a microscopic graph with 16 nodes and 20 positive and 7 negative links. (b) In a macroscopic viewpoint, G_3 is composed of 4 clusters, 18 internal positive and 2 internal negative links and 2 external positive and 5 external negative links. The self-loop represents the internal links between nodes in the same community. Nodes are in different clusters if they are plotted by different colors and shapes.

3.2.2. Representation

In this study, each chromosome (or solution) $\mathbf{x}_i$ is expressed as an integer vector [41]

$$\mathbf{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{iN}\},$$

where $x_{ij} \in \{1, 2, \ldots, N\}$ is the cluster identifier of node v_j [41]. For instance, $x_{ij} = 10$ indicates that the node v_j of ith chromosome is classified into the 10th cluster. With this representation, each solution represents a possible clustering division of this network. Thereafter, the nodes are divided into several clusters and the ones with the same cluster identifier are grouped in the same cluster.

In the transformation of structural balance, it is necessary to change signs of positive edges across different clusters and negative edges within the same cluster. Fig. 6 illustrates the integer vector representation on a toy network G_4 with 10 nodes. It can be clearly seen that the number of clusters is automatically determined by the integer vector $\mathbf{x}_i$.

3.2.3. Initialization

Initialization corresponds to **Step 1** of MLMSB. In general, the convergence of MAs can be accelerated by a population of initialized solutions with high quality and good diversity [28]. For the initialization to generate a population of high-quality solutions, it is necessary to incorporate network-specific knowledge. Here, we utilize the neighborhood of node, and generate a population of solutions by assigning the cluster identifiers of nodes to those of their neighbors. In order to maintain the diversity of the initial solutions, we adopt a random sequence strategy to update the cluster identifiers of nodes. Algorithm 2 lists the pseudocode of the network-specific initialization.

Algorithm 2. Network-specific initialization

1: **Input**: The size of population N_P.
2: Generate N_P solutions $\mathbf{X}(0) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_P}\}$, where $x_{ij} \leftarrow j, 1 \leqslant i \leqslant N_P, 1 \leqslant j \leqslant N$.
3: **for** each solution $\mathbf{x}_i$ of $\mathbf{X}(0)$ **do**
4: Generate a random sequence (e.g., $\{r_1, r_2, \ldots, r_N\}$).
5: **for** each chosen node v_{r_j} of G **do**
6: $x_{iu} \leftarrow x_{ir_j}, \forall u \in \{u | J_{ur_j} = 1\}$.
7: **end for**
8: **end for**
9: **Output**: $\mathbf{X}(0) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_P}\}$.

In the end, the network-specific initialization will generate a population of unsupervised solutions $\mathbf{X}(0)$.

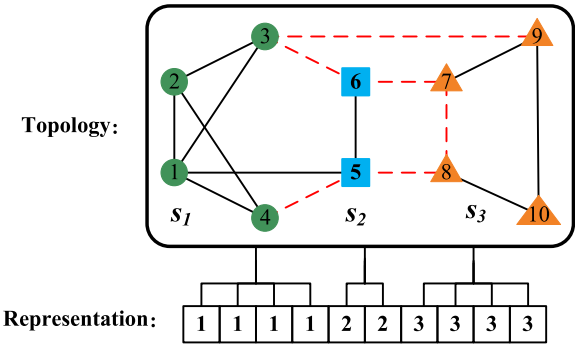


Fig. 6. Illustration of the integer vector representation on a toy network G_4. The toy network is divided into three clusters, $s_1 = \{1,2,3,4\}, s_2 = \{5,6\}$ and $s_3 = \{7,8,9,10\}$. The integer vector $\mathbf{x}_i$ can be represented as $\mathbf{x}_i = \{1,1,1,1,2,2,3,3,3,3\}$.

3.2.4. Tournament selection

In **Step 2** of MLMSB, we choose N_m solutions from the population as the parent chromosomes (solutions). Here, the classical tournament selection is used to choose the parent chromosomes. The tournament selection firstly generates N_m tournaments. Each tournament consists of N_o solutions chosen from the population randomly, where N_o is the size of the tournament. Then, the solution of each tournament with minimum H_w is chosen as the parent chromosomes $\mathbf{Y}(t)$, where t is the running number of generations.

3.2.5. Genetic operation

Genetic operation is the primary impetus of GA to generate promising regions in the solution space. Genetic operation mainly consists of crossover and mutation. The crossover aims to generate good offspring chromosomes by inheriting excellent properties from their parent chromosomes and the mutation is to generate disturbances [28].

In **Step 3** of MLMSB, we employ a two-way crossover and a neighborhood based mutation. The two-way crossover exchanges the clustering information of parent chromosomes $\mathbf{Y}(t)$. More specifically, for each pair of chromosomes $\mathbf{x}_a = \{x_{a1}, x_{a2}, \ldots, x_{aN}\}$ and $\mathbf{x}_b = \{x_{b1}, x_{b2}, \ldots, x_{bN}\}$ in $\mathbf{Y}(t)$, we choose a node v_q randomly and generate a random value within [0 1]. If the generated random value is smaller than the crossover probability p_c, then

$$\begin{cases} x_{a\mu} \leftarrow x_{bq}, & \forall \mu \in \{\mu | x_{b\mu} = x_{bq}\} \\ x_{bv} \leftarrow x_{aq}, & \forall v \in \{v | x_{av} = x_{aq}\} \end{cases}.$$

The two-way crossover can inherit useful clustering divisions from their parents [32]. Fig. 7 gives an illustration of the two-way crossover on two parent chromosomes $\mathbf{x}_a$ and $\mathbf{x}_b$. It shows that the offspring $\mathbf{x}_c$ inherits the cluster {1, 2, 3} and {4, 5, 6, 7} from $\mathbf{x}_b$ and $\mathbf{x}_a$, respectively.

In order to reduce useless explorations and generate dependable offsprings, we adopt a neighborhood based mutation which mutates the cluster identifiers of nodes with those of their neighbors. In the mutation processes, when a node has more than one neighbor and these neighbors are in different clusters, we randomly replace its cluster identifier with one of its neighbors. The neighborhood based mutation performs on each solution $\mathbf{x}_i$ generated by the crossover, and its processes can be expressed as follows. For each node v_j in $\mathbf{x}_i, 1 \leqslant j \leqslant N$, we generate a random value within $[0, 1]$ and choose at random a node v_u from the neighbors of v_j. If the generated random value is smaller than the mutation probability p_m, then $x_{ij} \leftarrow x_{iu}$.

Fig. 8 gives an illustration of the neighborhood based mutation. The solutions generated by the mutation form the offspring population $\mathbf{Z}(t)$.

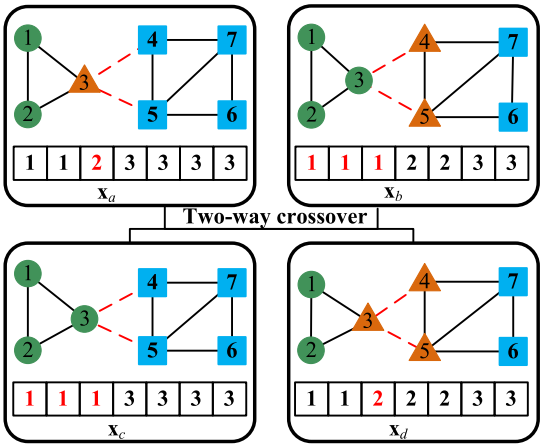


Fig. 7. Illustration of the two-way crossover. $\mathbf{x}_a = \{1,1,2,3,3,3,3\}$ and $\mathbf{x}_b = \{1,1,1,2,2,3,3\}$ are two parent chromosomes. Assumed that v_3 is chosen, nodes v_1, v_2 and v_3 in $\mathbf{x}_a$ are assigned with the cluster identifier of v_3 of $\mathbf{x}_b$ and node v_3 in $\mathbf{x}_b$ with the cluster identifier of v_3 of $\mathbf{x}_a$ after crossover.

3.2.6. Multi-level learning based local search

In MLMSB, GA can explore and exploit a few promising regions in the solution space. However, it cannot converge to the optimal solutions of promising regions within a few generations. A local search technique is designed to accelerate the convergence of GA to the optimal solutions [30]. Generally, start from an initial solution, and then a local search technique is adopted to repeatedly search for an improved solution by local changes [30]. A local search can effectively reduce useless explorations by incorporating problem-specific knowledge.

In this study, the neighborhoods of node, cluster and partition are incorporated into our multilevel learning technique which consists of a microscopic-level node learning, a macroscopic-level clustering learning and a partition learning. In **Step 4** of MLMSB, the multilevel learning is employed on the best solution $\mathbf{x}_l$ in $\mathbf{Z}(t)$. The visualization of the multilevel learning is shown in Fig. 9. In the following, detailed descriptions of the node learning, cluster learning and partition learning are given.

Node learning. In the node learning, each node v_{r_i} in $\mathbf{x}_l$ is reassigned to the cluster of its neighbor v_{r_u}, which will result in the maximum decrement in H_w. More specifically, for each node v_{r_i} chosen in a random sequence $R_1 = \{r_1, r_2, \ldots, r_N\}$, its cluster identifier x_{lr_i} is updated as follows.

$$x_{lr_i} = \arg \max_{x_{lu}} \left(-\Delta H_w(\mathbf{x}_l|_{x_{lr_i} \leftarrow x_{lu}}) \right), \quad u \in \{u | J_{r_i u} = 1\},$$

where $\Delta H_w(\mathbf{x}_l|_{x_{lr_i} \leftarrow x_{lu}})$ represents the decrement of H_w when node v_{r_i} is reassigned to the cluster of v_u. The node learning ends when the assignment of each node to the clusters of its neighbors cannot result in the decrement of H_w.

The node-level learning can converge to a good solution quickly. However, the node learning technique falls into a local optimum easily because it cannot traverse all solutions in the promising regions. By incorporating the neighborhood structure of cluster, we present a cluster learning technique to find a better solution around the solution $\mathbf{x}_l = \{s_1, s_2, \ldots, s_k\}$ generated by the node learning technique, where $s_i = \{v_{i_1}, v_{i_2}, \ldots, v_{i_n}\}$ is a cluster with n nodes.

Cluster learning. In the cluster learning, each cluster s_i updates its cluster identifier with that of its neighbor s_q, which will lead to the maximum decrement in H_w. More specifically, for each cluster s_i chosen in a random order $R_2 = \{r_1, r_2, \ldots, r_k\}$, its cluster identifier x_{ls_i} is updated as follows.

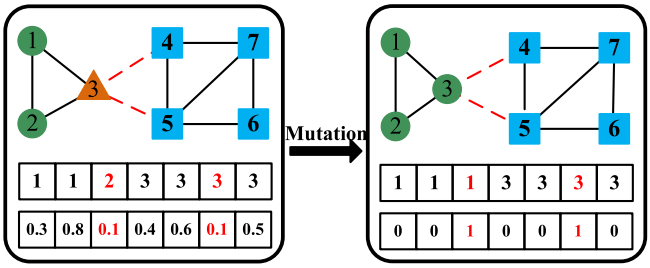


Fig. 8. Illustration of mutation operation with $p_m = 0.2$.

$$x_{ls_i} = \arg\max_{x_{ls_q}} \left(-\Delta H_w(\mathbf{x}_l|_{x_{ls_i} \leftarrow x_{ls_q}}) \right), \forall q \in \{q | A^+_{s_i s_q} \geqslant 1\}.$$

The updating process ends when the cluster identifier of each cluster keeps unchanged.

The cluster learning considers different neighborhood structures compared with the node learning, which makes it possible to find a better solution around the solution $\mathbf{x}_l$. However, the cluster learning is still easy to trap into a local optimal clustering division because it is hard to separate the merged nodes and communities again [33].

Partition learning. In order to overcome the constraint of cluster learning, we design the partition learning. The partition learning performs on two solutions $\mathbf{x}_l = \{x_{l1}, x_{l2}, \ldots, x_{lN}\}$ and $\mathbf{x}_g = \{x_{g1}, x_{g2}, \ldots, x_{gN}\}$, where $\mathbf{x}_l$ is the solution generated by the cluster learning and $\mathbf{x}_g$ is the best solution in the population, and it consists of two steps. The first step is to find the consensus neighbor $\mathbf{x}_f$ of $\mathbf{x}_g$ and $\mathbf{x}_l$. At this step, the nodes are divided into the same cluster if they have the same identifiers in both $\mathbf{x}_g$ and $\mathbf{x}_l$, but separated into different clusters if they have the same identifiers in $\mathbf{x}_g$ while different identifiers in $\mathbf{x}_l$. An illustration of this step is shown in Fig. 9. The second step uses the node and cluster learning to optimize the consensus neighbor $\mathbf{x}_f$, and generates a solution $\mathbf{x}'_l$.

The partition learning can learn the skeletal structure from $\mathbf{x}_l$ and $\mathbf{x}_g$ and evolve the consensus neighbor towards $\mathbf{x}_l$ or $\mathbf{x}_g$, or even a better solution.

Computation of ΔH_w. The ΔH_w of merging a node v_i from its cluster s_i into its neighboring cluster s_j can be quickly computed as

$$\Delta H_w(\mathbf{x}_l|_{x_{li} \leftarrow \mathbf{x}_{ls_j}}) = w \cdot \sum_{(i,q),q \in s_i} |J^+_{iq}| + (1-w) \cdot \sum_{(i,q),q \in s_i} |J^-_{iq}| - w \cdot \sum_{(i,p),p \in s_j} |J^+_{ip}| - (1-w) \cdot \sum_{(i,p),p \in s_j} |J^-_{ip}|. \quad (6)$$

The computational complexity of ΔH_w is $O(\bar{k})$, where $\bar{k}$ is the averaged degree of a signed network.

The reasons why the multilevel learning technique works are that the multilevel learning technique incorporates the network-specific knowledge into the search rules. Moreover, the node learning and the cluster learning can quickly converge to the optimal solutions in the promising regions discovered by GA. In addition, the partition learning overcomes the constraint of cluster learning. Finally, extensive experiments demonstrate the effectiveness of the multilevel learning technique.

3.2.7. Complexity analysis

It is time-consuming to implement the genetic operations and multilevel learning of MLMSB. In genetic operations, the two-way crossover and the mutation are implemented for $N_m/2$ and N_m independent times, respectively, where N_m represents the size of chosen solutions. The computational complexities of the crossover and the mutation are $O(\bar{k})$ and $O(\bar{k} + M^+)$, respectively, where $\bar{k}$ is the averaged degree and M^+ represents the number of positive edges. In the node and cluster learning, ΔH_w is computed M^+ times. Moreover, the formation of clusters needs $\bar{k}$ basic operations in the cluster learning. In addition, the identification of consensus neighbor needs N basic operations in the partition learning. Therefore, the computational complexities of the node learning, cluster learning and partition learning are $O(\bar{k}M^+), O(M\bar{k}^2)$ and $O(N + M^+\bar{k} + M\bar{k}^2)$, respectively. For MLMSB, its computational complexity is $O(\bar{k}^2 M g_{max})$, where g_{max} is the maximum number of iterations.

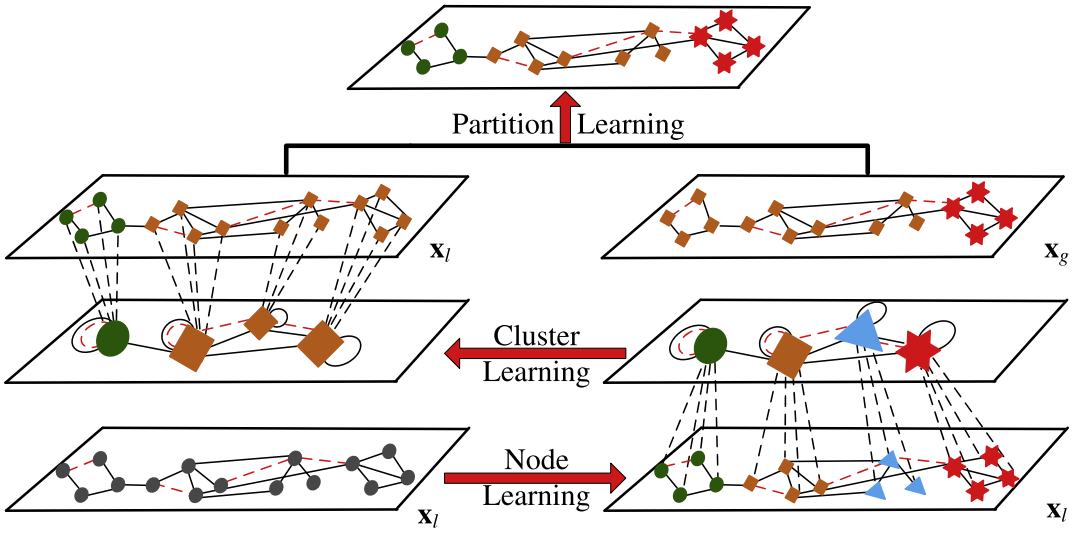


Fig. 9. Visualization of the multilevel learning on the best solution $\mathbf{x}_l$ in $\mathbf{Z}(t)$. The node learning technique updates $\mathbf{x}_l$ by merging any two neighboring nodes. The cluster learning technique consists of two phases: (1) build a new network whose nodes are the found clusters and (2) update $\mathbf{x}_l$ by merging any neighboring nodes in the new network. The partition learning technique evolves a consensus neighbor of two solutions $\mathbf{x}_l$ and $\mathbf{x}_g$ towards a better solution. The red dash lines are negative edges while the black dash lines represent mappings between nodes in a microscopic viewpoint and super-nodes in a macroscopic viewpoint. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2.8. Comparisons between MLMSB and our previous works

MLMSB and our previous works Meme-Net [32], MLCD [33] and Meme-SB [16] model the issue of complex systems as a network optimization problem, and use the network-specific MA to solve the modeled problem. However, they are different in the motivation, optimization model, GA operation, network-specific knowledge and local search.

As for the motivation, Meme-Net focuses on detecting the multi-resolution community structures of undirected networks. However, it is impossible for Meme-Net to detect communities in large-scale networks because of its high computational complexity [33]. MLMCD is devised to identify communities in large-scale networks. Meme-SB aims at computing the structural balance of signed networks in the strong definition. MLMSB is designed for the transformation of structural balance of signed networks.

In the optimization model, Meme-Net models the community detection in undirected networks as the optimization of modularity density (D_λ) [42]. Multi-resolution communities can be detected by maximizing D_λ with different λ. MLCD models the community detection of networks as the optimization of modularity (Q) [43]. The maximization of Q can detect communities with massive internal links and few external links. Meme-SB models the computation of structural balance as the optimization of energy function h while MLMSB solves the transformation of structural balance by optimizing a more general energy function H_w.

In the adopted GA operation, Meme-Net, MLCD and MLMSB have different representations from Meme-SB. Meme-Net, MLCD and MLMSB adopt an integer vector representation while Meme-SB uses a bit vector representation. Moreover, Meme-Net and MLMSB have different initializations from Meme-SB and MLCD. In the initialization, Meme-Net and MLMSB assign the cluster identifiers of nodes to those of their neighbors while Meme-SB and MLCD update the cluster identifiers of nodes with those of their neighbors. In addition, Meme-Net, MLCD and MLMSB have different crossover operators from Meme-SB. Meme-Net, MLCD and MLMSB use a two-way crossover while Meme-SB adopts a two-point crossover.

In the incorporated network-specific knowledge, Meme-Net uses the undirected links among nodes while MLCD adopts both the undirected connections of node and community. Meme-SB and MLMSB are used for tackling signed networks. Meme-SB uses the neighborhood of node while MLMSB adopts the neighborhoods of node, cluster and partition.

In the devised local search, Meme-Net and Meme-SB adopt a hill climbing and node learning as the local search, respectively. The hill climbing and the node learning optimize the clustering division of networks by updating the cluster identifier of nodes iteratively. MLCD and MLMSB adopt multilevel learning techniques as the local search. However, the detailed operations of the multilevel learning are different because MLCD and MLMSB tackle different optimization models and types of networks.

4. Experimental results

In this section, we test MLMSB on five real-world signed networks and compare it with nine classical algorithms.

4.1. Experimental networks

Five signed networks, including two small-scale international relationship networks, two large-scale online vote networks and a large-scale online blog network, are chosen in our experiments. The computation and transformation of the imbalances of two international relationship networks can reduce local and global conflicts. The computation and transformation of the imbalanced links of the three social online networks can find the missing and spurious operations of websites, collect authentic and reliable information and promote the development of social systems. The main properties of these networks are shown in Table 1. The extracted networks can be downloaded from our homepage http://see.xidian.edu.cn/faculty/mggong/publication.htm.

Gahuku-Gama Subtribes Network (GGS): The GGS network constructed by Read in [44] represents the cultures of highland New Guinea. The positive/negative links represent political unions/oppositions among 16 Gahuku-Gama subtribes (http://mrvar.fdv.uni-lj.si/sola/info4/andrej/prpart5.htm).

War network: The War network derived from the Correlates of War project [45] illustrates the international relations in the period from 2000 to 2010. The War network is extracted from the datasets "Militarized Interstate Disputes (v4.01)" and "Formal Alliances (v4.1)" in http://www.correlatesofwar.org/-datasets.htm, and it consists of 217 nodes and 1295 positive and 138 negative edges. Nodes represent countries, and positive (negative) links indicate conflict (alliance) relations.

Wiki-ele network: The Wiki-ele network consists of 7114 vertices and 78,792 positive and 21,529 negative links. Nodes represent users participating in the elections of administrators (either electors or candidates) and positive (negative) edges indicate positive (negative) elections. Detailed descriptions of the adopted Wiki-ele network are provided in Refs. [6,7] and http://snap.stanford.edu/data/wiki-Vote.html.

Wiki-rfa network: The Wiki-rfa network records the statistical requests from any Wikipedia member for Wikipedia editors to become administrators. A request can be represented by a supporting, neutral, or opposing vote. In the Wiki-rfa network, there are 11,402 users and 189,004 distinctly directed voter pairs including 139,312 supporting votes, 11,225 neutral votes and 38,467 opposing votes. Here, we consider a symmetrical signed Wiki-rfa network where nodes represent Wikipedia members and positive (negative) links indicate supporting (opposing) votes. After discarding the neutral votes, the signed undirected Wiki-rfa network is composed of 11,276 nodes and 132,962 positive and 37,972 negative links. Detailed descriptions of the Wiki-rfa network are given in http://snap.stanford.edu/data/wiki-RfA.html.

Slashdot network: Slashdot is a famous website for users to submit current primarily technology oriented news and for editors to evaluate these news. In Slashdot website, each user can be tagged by the other users as friends or foes. The dataset "s oc-sign-Slashdot081106" in http://snap.stanford.edu/data/#signnets is composed of 77,357 users and 516,575 tags. This dataset can be modeled as a Slashdot network with 77,357 nodes and 516,575 directed and signed links. Notes represent users and positive (negative) edges indicate friendly (hostile) tags among users [6]. The largest strongly connected component (SCC) of Slashdot network consists of 26,996 nodes and 337,351 signed directed edges. Here, its signed undirected version, which has 26,995 nodes and 217,563 positive and 71,075 negative undirected edges, is chosen in our experiments.

Table 1
Real-world signed networks. M^+ (M^-) represents the number of positive (negative) links in signed networks.

Networks	Nodes	Edges	M^+	M^-	M^-/M
GGS	16	58	29	29	0.5
War	216	1433	1295	138	0.0963
Wiki-ele	7114	100,321	78,792	21,529	0.2146
Wiki-rfa	11,276	170,934	132,962	37,972	0.2222
Slashdot	26,995	288,638	217,563	71,075	0.2462

Assuming that the directed and the undirected versions of SCC of Slashdot network are represented as J and J', respectively, we generate J' as follows.

$$J'_{ij} = \begin{cases} J_{ij} & \text{if } J_{ij} = J_{ji} \vee J_{ij} \neq 0 \wedge J_{ji} = 0, \\ J_{ji} & \text{if } J_{ij} = 0 \wedge J_{ji} \neq 0, \\ 0 & \text{if } J_{ij} = -J_{ji}. \end{cases} \quad (7)$$

4.2. Comparison algorithms

In this paper, MLMSB is compared with nine classical algorithms, namely GA, OLMSB, TLMSB, SLPAm, SBGLL, MODPSO, SN2013, PRE2014 and FEC. Their computational complexities are recorded in Table 2.

GA, OLMSB and TLMSB are the variant versions of MLMSB without the local search, the cluster and partition learning and the partition learning, respectively. Comparisons between MLMSB, GA, OLMSB and TLMSB are made to demonstrate the effectiveness of the devised node, cluster and partition learning techniques on the acceleration of MLMSB to an optimal solution.

SLPAm and SBGLL are the extended versions of classical greedy algorithms LPAm [46] and BGLL [47], respectively. LPAm detects the communities of unsigned networks by a modularity-specific label propagation while SLPAm computes the imbalances of signed networks by a H_w-specific label propagation. BGLL is a fast two-phase community detection algorithm based on the optimization of modularity [43,48]. In the first (second) phase, the node (community) chosen from the first one to the last one is assigned to its neighboring community when this change can increase modularity maximally. SBGLL focuses on the imbalance computation of signed networks based on the optimization of H_w. The reason why we choose them as our comparison algorithms is that we want to illustrate that the hybrid algorithm MLMSB can perform better than the classical greedy algorithms SPAm and SBGLL in terms of accuracy.

FEC [19] adopts an agent-based technique to identify the communities in signed networks. In FEC, communities are composed of a few nodes with dense positive but sparse negative links. The positive links across different communities and the negative links within the same community are considered as unbalanced links. A comparison between MLMSB and FEC is made to analyze the relations between the structural balance and the communities under different definitions in signed networks.

MODPSO [20] is a population-based multi-objective evolutionary algorithm by optimizing signed ratio association and signed ratio cut simultaneously. MODPSO starts from an initial population of solutions and it iteratively optimizes these solutions based on the framework of multi-objective discrete swarm optimization. MODPSO returns a set of solutions, each of which represents one possible balance transformation. Comparison between MLMSB and MODPSO is made to illustrate the superb performance of the proposed algorithm in terms of accuracy and stability compared with the population-based evolutionary algorithm.

PRE2014 [17] and SN2013 [25] are two relevant works on the computation of imbalance in signed networks. PRE2014 adopts a classical community detection algorithm Infomap [18] to detect communities in signed networks. SN2013 updates the cluster information of each node by: (i) the movement of the node from its cluster to another cluster and (ii) the interchange of two nodes between two clusters. PRE2014 and SN2013 evaluate the imbalance of signed networks by computing the number of positive links across the found communities and negative links within the same community.

Table 2
Parameters settings and computational complexity of the algorithms.

Algorithm	g_{max}	N_p	N_m	p_c	p_m	N_o	Complexity
MLMSB	100	100	10	0.9	0.1	2	$O(\bar{k}^2 M g_{max})$
GA	100	100	10	0.9	0.1	2	$O(N_m M g_{max})$
OLMSB	100	100	10	0.9	0.1	2	$O(\bar{k} M g_{max})$
TLMSB	100	100	10	0.9	0.1	2	$O(\bar{k}^2 M g_{max})$
SLPAm	100	–	–	–	–	–	$O(\bar{k} M g_{max})$
SBGLL	100	–	–	–	–	–	$O(\bar{k}^2 M g_{max})$
MODPSO	100	100	10	–	0.1	2	$O(N_P M g_{max})$
SN2013	300	–	–	–	–	–	$O(\bar{k} M g_{max})$
PRE2014	–	–	–	–	–	–	$O(\bar{k}^2 M)$
FEC	–	–	–	–	–	–	$O(M)$

4.3. Experiments for the computation of structural balance

Table 3 records the statistical results over 50 independent trials of all algorithms with parameters settings shown in Table 2. In Table 3, the $H_{min}, \overline{H}$ and H_{std} indicate the minimum, mean and standard deviation of H values, respectively. Moreover, the P-value is a statistical hypothesis (i.e., t-test) value, and it is the probability of getting the observed data when the null hypothesis is actually true. Here, the null hypothesis is expressed as that there is no difference between MLMSB and its comparison algorithm on the computation of structural balance, and the observed data are the statistical H values of the comparison algorithm over 50 trials. If the P-value is less than a significance level $\alpha = 0.05$, it indicates that the observed data are inconsistent with the assumption that the null hypothesis is true. It is noticed that a small P-value cannot imply that there is a meaningful difference of MLMSB with its comparison algorithm on the computation of structural balance. In addition, the symbols '>', '<' and '=' denote that MLMSB has better significance than, lower significance than and similar significance to its comparison algorithm, respectively, and they are generated based on the P-value of the comparison between MLMSB and the others algorithms and their $\overline{H}$ values. More specifically, if the P-value is smaller than 0.05 and MLMSB has larger (smaller) $\overline{H}$ value than its comparison algorithm, then the significance is labeled as '>' ('<'). If the P-value is larger than 0.05 and the $\overline{H}$ value of MLMSB is far larger than or far smaller than that of its comparison algorithm, then the significance is labeled as '>' or '<'. In the case of values with similarity, we label the significance as '='.

Table 3 shows/demonstrates that MLMSB can achieve the smallest H_{min} and $\overline{H}$ for all networks. Though some algorithms, e.g., OLMSB and TLMSB, can also yield the smallest H_{min} and $\overline{H}$ for the small-scale networks such as GGS and War, none of the comparison algorithms can effectively find the solutions with the smallest H_{min} and $\overline{H}$ values in the large-scale networks such as Wiki-ele, Wiki-rfa and Slashdot.

It is worthwhile to note that SBGLL, PRE2014 and FEC are deterministic algorithms which produce the same output for a particular input, whereas MLMSB, GA, OLMSB, TLMSB, SPLAm, MODPSO and SN2013 are non-deterministic algorithms whose output varies. Of the non-deterministic algorithms, MLMSB has the smallest H_{std} for the GGS, War, Wiki-ele, Wiki-rfa and Slashdot networks and GA, OLMSB and TLMSB have the smallest H_{std} values for the GGS network. The comparisons of H_{std} between MLMSB and its comparison algorithms demonstrate the stability of MLMSB on the computation of H.

From Table 3, we can see that the P-value of most comparisons are smaller than the significance level $\alpha = 0.05$. Table 3 also shows that the significance symbols of the comparisons between MLMSB, GA, OLMSB, TLMSB, SBGLL and PRE2014 are labeled as '=' for the small-scale GGS network. Furthermore, the significance symbols of the comparisons between MLMSB, GA, OLMSB, TLMSB, SLPAm,

Table 3
Statistic results of H over 50 independent trials on the six real-world signed networks. '–' represents that the algorithm cannot tackle it. '×' indicates that the observed data are the same as the tested data. The bold values correspond to the best results.

Networks	Indexes	MLMSB	GA	OLMSB	TLMSB	SLPAm	SBGLL	MODPSO	SN2013	PRE2014	FEC
GGS	H_{min}	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	4
	$\overline{H}$	**2**	**2**	**2**	**2**	2.720	**2**	2.620	2.560	**2**	4
	H_{std}	**0**	**0**	**0**	**0**	0.9648	**0**	1.638	1.057	**0**	**0**
	P-value	×	×	×	×	$3.3e^{-11}$	×	0.0003	$7.12e^{-7}$	×	0
	Significance	=	=	=	=	>	=	>	>	=	>
War	H_{min}	**40**	43	**40**	**40**	43	44	44	44	103.0	59
	$\overline{H}$	**41.23**	59.99	45.91	41.53	62.38	44	49.25	80.38	103.0	59
	H_{std}	0.6678	5.994	3.638	0.7582	5.299	**0**	4.693	16.42	**0**	**0**
	P_{value}	×	$9.9e^{-53}$	$1.2e^{-21}$	0.0887	$2.8e^{-63}$	$4.5e^{-52}$	$1.9e^{-29}$	$2.1e^{-42}$	$6e^{-185}$	$3e^{-131}$
	Significance	=	>	>	=	>	>	>	>	>	>
Wiki-ele	H_{min}	**13,838**	26,355	14,310	14,012	13,969	13,902	23,829	14,077	34,215	18,786
	$\overline{H}$	**13,841**	29,658	14,456	14,097	16,919	13,902	33,113	14,104	34,215	18,786
	H_{std}	2.201	2868	92.23	40.92	4954	**0**	7625	14.73	**0**	**0**
	P_{value}	×	$2.61e^{-8}$	$5.70e^{-9}$	$9.16e^{-9}$	0.0081	$1.6e^{-14}$	$2.23e^{-5}$	$2.3e^{-12}$	$3.2e^{-37}$	$1.1e^{-31}$
	Significance	=	>	>	>	>	>	>	>	>	>
Wiki-rfa	H_{min}	**25,972**	54,345	26,796	26,283	26,205	26,026	49,967	26,328	62,516	33,761
	$\overline{H}$	**25,974**	63,081	27,143	26,370	33,072	26,026	49,967	26,363	62,516	33,761
	H_{std}	1.429	6105	788.6	56.07	4310	**0**	**0**	12.98	**0**	**0**
	P_{value}	×	$1.29e^{-8}$	0.0011	$3.26e^{-9}$	0.0006	$3.6e^{-41}$	$1.5e^{-39}$	$6.8e^{-15}$	$3.5e^{-41}$	$3.8e^{-35}$
	Significance	=	>	>	>	>	>	>	>	>	>
Slashdot	H_{min}	**62,117**	185,421	62,433	62,290	62,321	62,338	–	62,530	70,843	70,628
	$\overline{H}$	**62,120**	189,225	62,604	62,310	62,370	62,338	–	62,534	70,843	70,628
	H_{std}	1.398	2577	113.6	16.41	36.59	**0**	–	4.243	**0**	**0**
	P_{value}	×	$9.3e^{-17}$	$2.81e^{-7}$	$2.4e^{-11}$	$8.5e^{-20}$	$3.0e^{-21}$	–	$2.1e^{-42}$	$1.1e^{-35}$	$1.4e^{-35}$
	Significance	=	>	>	>	>	>	–	>	>	>

SBGLL, SN2013 and PRE2014 are labeled as '>' for the large-scale Wiki-ele, Wiki-rfa and Slashdot networks.

From Table 3, we can verify the effectiveness of the proposed learning techniques easily. For the small-scale GGS network, GA, OLMSB, TLMSB and MLMSB have the same $H_{min}, \overline{H}$ and H_{std} values. However, for the small-scale War network and the large-scale Wiki-ele, Wiki-rfa and Slashdot networks, the $H_{min}, \overline{H}$ and H_{std} values found by GA are the largest. OLMSB which incorporates the node learning into GA has smaller $H_{min}, \overline{H}$ and H_{std} values than GA; TLMSB which incorporates the cluster learning into OLMSB has smaller $H_{min}, \overline{H}$ and H_{std} values than OLMSB; and MLMSB which incorporates the partition learning into TLMSB has smaller $H_{min}, \overline{H}$ and H_{std} values than TLMSB. It is noticed that the improvements of these algorithms are achieved at the cost of an acceptable increase in computational complexity.

Figs. 10 and 11 exhibit the clustering results of the GGS network and the War network with the smallest H, respectively. The clustering results reflect the potential community structures and the unbalanced links obviously. The GGS network is divided into three clusters, and it has two unbalanced links (e.g., the relations between 'MASIL' and 'NAGAM', 'MASIL' and 'UHETO'). The War network is mainly classified into two clusters, and it has 40 unbalanced links highlighted with black lines.

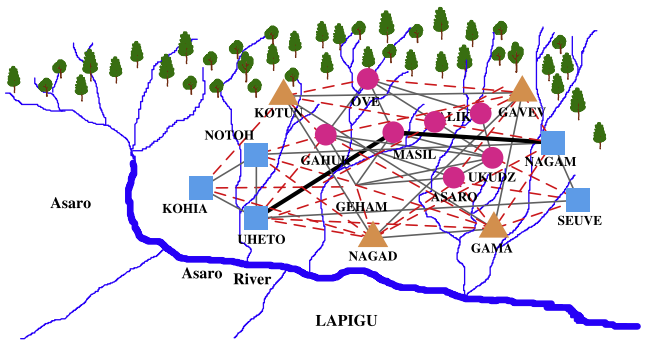


Fig. 10. Clustering result of the GGS network with $H = 2$. The unbalanced edges are highlighted in black.

Fig. 12 shows the variations of the H values obtained by the iterative optimization algorithms MLMSB, GA, OLMSB, TLMSB, SLPAm, SBGLL, MODPSO and SN2013 with the number of iterations on the War and Wiki-ele networks. The results illustrate that MLMSB, OLMSB, TLMSB, SLPAm, SBGLL and MODPSO converge within 100 generations while SN2013 converge within 300 generations. The results also show that the global search algorithm GA cannot converge within 100 generations, and that MLMSB, TLMSB and SBGLL can quickly converge to good solutions with low H values.

4.4. Experiments for the transformation of structural balance

The mean value of H_w of all algorithms is listed in Table 4, where H_w denotes the balance transformation cost. In this subsection, we investigate multiple balance transformations at minimum cost by varying the cost coefficient w from 0 to 1 with interval 0.1. Note that, only situations where $0 < w < 1$ are investigated since the cases where $w = 0$ and $w = 1$ hardly take place in practical applications.

The results in Table 4 show that for the small-scale GGS and War networks, MLMSB, OLMSB, TLMSB and SBGLL have smaller H_w values than SLPAm, MODPSO, SN2013, PRE2014 and FEC. For the large-scale Wiki-ele, Wiki-rfa and Slashdot networks, MLMSB, SBGLL and SN2013 have smaller H_w values than GA, OLMSB, TLMSB, SLPAm, MODPSO, PRE2014 and FEC. Furthermore, our algorithm MLMSB has the smallest H_w values for all networks under different w values but the Wiki-rfa network under $w = 0.1$.

From Table 4, we can see that MLMSB and its three variations can transform the unbalanced GGS network into balanced ones at minimum cost. For the War, Wiki-ele, Wiki-rfa and Slashdot networks, GA has the largest H_w values. OLMSB can effectively reduce

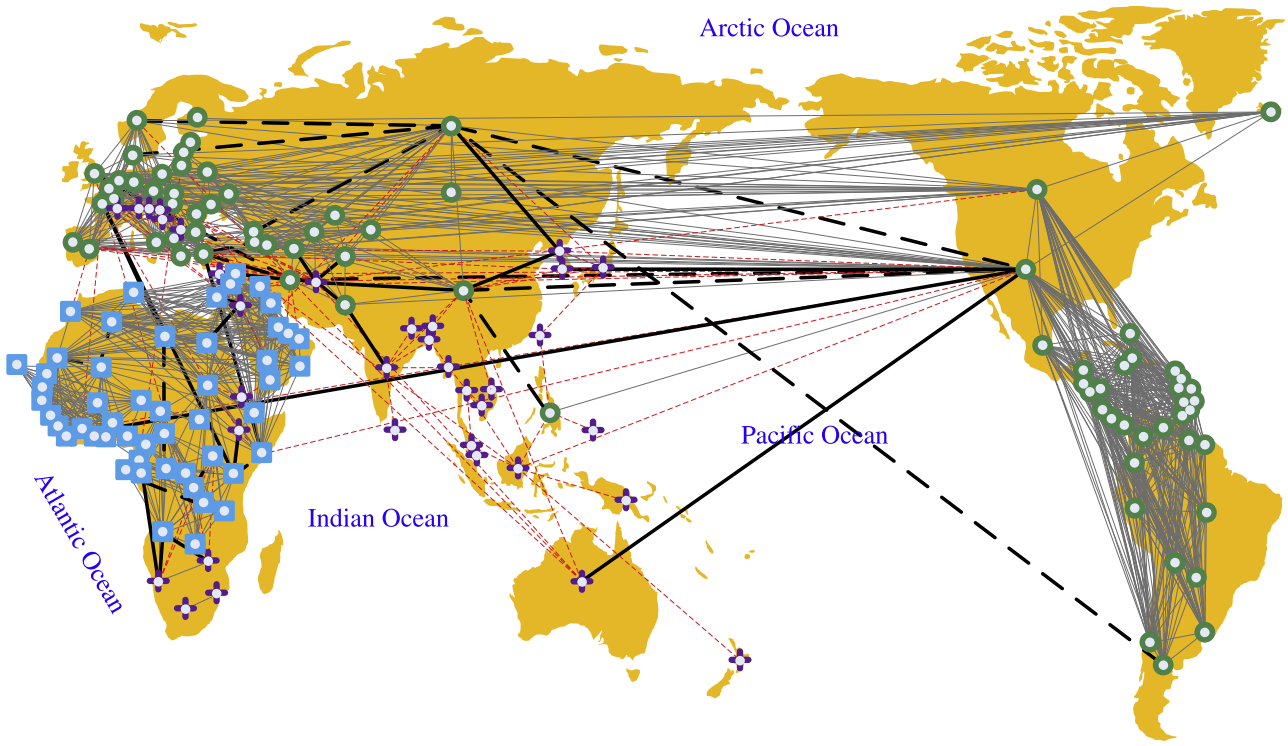


Fig. 11. Clustering result of the war network with $H = 40$. The nodes in the clusters with less than 3 nodes are marked with cross, and the unbalanced edges are highlighted in black.

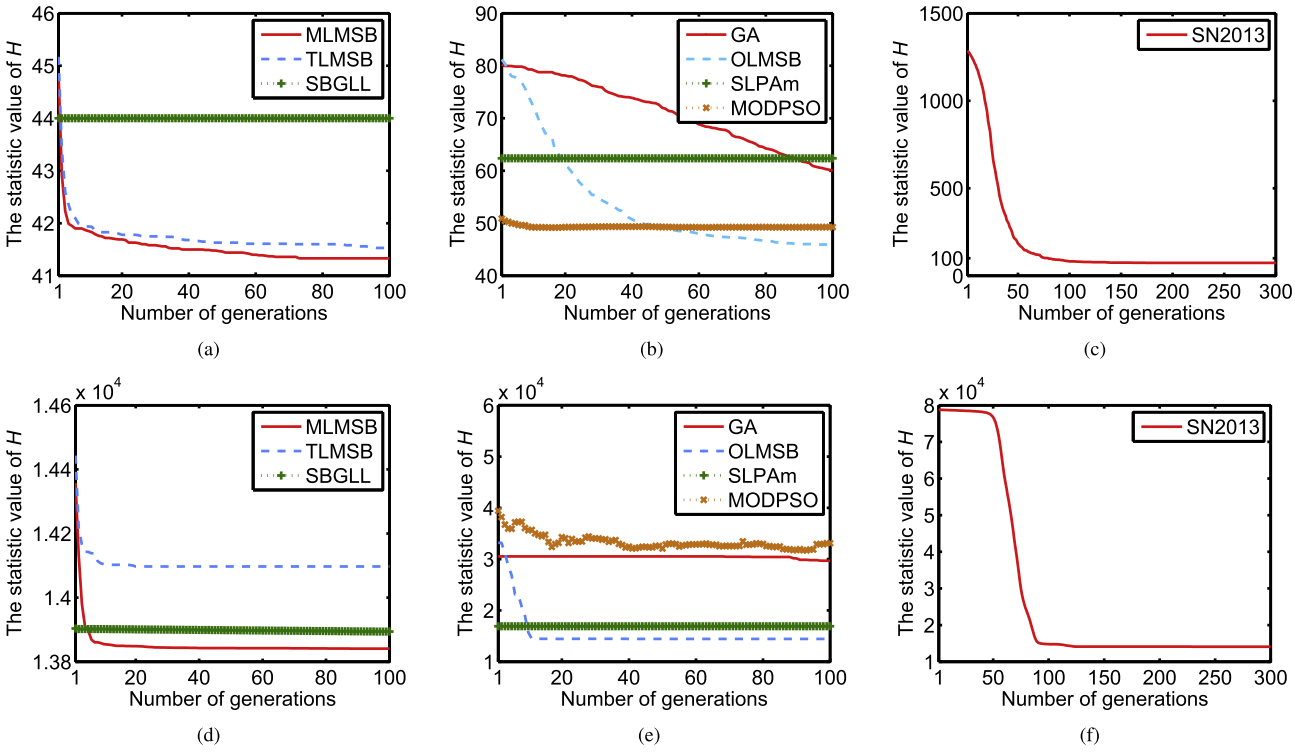


Fig. 12. Variation of H obtained by (a) MLMSB, TLMSB and SBGLL, (b) GA, OLMSB, SLPAm and MODPSO and (c) SN2013 with the number of generations on the war network and that by (d) MLMSB, TLMSB and SBGLL, (e) GA, OLMSB, SLPAm and MODPSO and (f) SN2013 with the number of generations on the Wiki-ele network.

the balance transformation cost of GA. More specifically, the maximum decrease of the cost can reach 45.86%, 90.99%, 92.95% and 95.84% for the War, Wiki-ele, Wiki-rfa and Slashdot networks when $0 < w < 1$, respectively. Moreover, TLMSB can reduce 57.82%, 3.41%, 3.01 % and 0.98% transformation cost for the War, Wiki-ele, Wiki-rfa and Slashdot networks, respectively. In addition, MLMSB can reduce the balance transformation cost of TLMSB. More specifically, H_w is maximally decreased by 1.64% for the

Table 4
Comparison results of H_w between MLMSB and the others algorithms on the five real-world networks. The bold values correspond to the best results.

Network	Alg.	$w=0$	$w=0.1$	$w=0.2$	$w=0.3$	$w=0.4$	$w=0.5$	$w=0.6$	$w=0.7$	$w=0.8$	$w=0.9$	$w=1$
GGS	MLMSB	**0**	**0.4**	**0.8**	**1.2**	**1.6**	**2**	**2**	**2.8**	**2.8**	**1.4**	**0**
	GA	**0**	**0.4**	**0.8**	**1.2**	**1.6**	**2**	**2**	**2.8**	**2.8**	**1.4**	**0**
	OLMSB	**0**	**0.4**	**0.8**	**1.2**	**1.6**	**2**	**2**	**2.8**	**2.8**	**1.4**	**0**
	TLMSB	**0**	**0.4**	**0.8**	**1.2**	**1.6**	**2**	**2**	**2.8**	**2.8**	**1.4**	**0**
	SLPAm	**0**	0.5560	1.104	1.632	2.208	2.720	2.920	3.458	4.372	4.794	5.040
	SBGLL	**0**	**0.4**	**0.8**	**1.2**	**1.6**	**2**	**2**	**2.8**	**2.8**	**1.4**	**0**
	MODPSO	1.680	1.868	2.056	2.244	2.432	2.620	2.808	2.996	3.184	3.372	3.560
	SN2013	**0**	0.5560	1.000	1.608	2.328	2.560	2.968	3.738	3.664	3.924	3.800
	PRE2014	**0**	**0.4**	**0.8**	**1.2**	**1.6**	**2**	2.4	**2.8**	3.2	3.6	4
	FEC	**0**	0.8	1.6	2.4	3.2	4	4.8	5.6	6.4	7.2	8
War	MLMSB	**0**	**13.83**	**24.04**	**30.83**	**36.40**	**41.23**	**41.00**	**36.03**	**28.40**	**16.40**	**0**
	GA	**0**	20.38	24.11	44.68	52.96	59.99	64.55	69.72	71.91	71.81	75.68
	OLMSB	**0**	14.78	25.47	33.80	40.31	45.91	48.85	46.73	42.10	38.88	33.56
	TLMSB	**0**	14.06	24.08	31.04	36.60	41.53	41.08	36.06	**28.40**	**16.40**	**0**
	SLPAm	**0**	17.29	29.95	42.07	52.21	62.38	69.34	76.85	85.06	90.85	95.36
	SBGLL	**0**	16	27.60	36.60	42.40	44	42.80	38.20	30.80	**16.40**	**0**
	MODPSO	40.24	44.14	46.82	48.03	48.76	49.25	48.65	41.35	33.44	25.42	17.40
	SN2013	**0**	21.29	37.98	51.03	66.04	80.38	86.09	89.73	110.7	112.4	107.4
	PRE2014	24	39.80	55.60	71.40	87.20	103.0	118.8	134.6	150.4	166.2	182.0
	FEC	22	29.40	36.80	44.20	51.60	59.00	66.40	73.80	81.20	88.60	96.00
Wiki-ele	MLMSB	**0**	**7475**	**11,706**	**13,997**	**14,537**	**13,841**	**12,128**	**9828**	**7022**	**3738**	**0**
	GA	3532	16,298	24,935	27,505	29,874	29,658	29,838	33,409	30,805	41,810	40,337
	OLMSB	**0**	7888	12,367	14,696	15,316	14,456	12,569	10,078	7143	3766	7
	TLMSB	**0**	7619	11,955	14,329	14,854	14,097	12,316	9935	7077	3746	**0**
	SLPAm	**0**	8017	12,263	14,560	15,325	16,919	14,084	11,357	7095	5740	39.40
	SBGLL	**0**	7617	11,864	14,147	14,630	13,902	12,184	9855	7036	3704	**0**
	MODPSO	10,980	20,643	29,530	31,157	32,135	33,113	34,091	35,069	36,039	36,993	37,948
	SN2013	**0**	7645	12,131	14,420	14,951	14,104	12,334	11,401	7154	3813	76.00
	PRE2014	20,106	22,923	25,750	28,571	31,393	34,215	37,036	39,858	42,680	45,502	48,324
	FEC	37,520	33,773	30,026	26,279	22,533	18,786	15,039	11,292	7545	3799	52
Wiki-rfa	MLMSB	**0**	14,263	**22,255**	**26,537**	**27,432**	**25,974**	**22,709**	**18,291**	**12,894**	**6762**	**0**
	GA	7030	28,554	45,581	53,161	58,627	63,081	72,800	69,620	79,809	96,478	99,156
	OLMSB	0.200	14,813	23,098	27,622	28,699	27,143	23,358	18,663	13,063	6804	7.400
	TLMSB	**0**	14,367	22,660	27,091	27,972	26,370	23,016	18,469	12,973	6778	**0**
	SLPAm	**0**	14,922	25,819	33,150	33,518	33,071	37,227	37,124	28,228	29,013	23,111
	SBGLL	**0**	**13,997**	22,374	26,657	27,559	26,026	22,760	18,316	12,920	6766	**0**
	MODPSO	99,934	89,941	79,947	69,954	59,960	49,967	39,974	29,980	19,987	9993	**0**
	SN2013	**0**	14,267	23,074	28,047	28,070	26,363	24,308	20,098	14,931	6827	52.40
	PRE2014	35,708	41,070	46,431	51,792	57,154	62,516	67,886	73,239	78,601	83,962	89,324
	FEC	67,402	60,674	53,946	47,217	40,489	33,761	27,033	20,305	13,576	6848	120
Slashdot	MLMSB	**0**	**29,346**	**51,498**	**63,277**	**66,153**	**62,120**	**52,804**	**40,985**	**27,770**	**13,944**	**0**
	GA	1873	41,894	81,432	118,385	155,100	189,225	226,890	262,687	295,975	334,869	370,136
	OLMSB	**0**	30,126	52,094	64,241	66,989	62,604	53,035	41,059	27,778	**13,944**	**0**
	TLMSB	**0**	30,420	52,001	63,982	66,624	62,310	52,883	41,008	**27,770**	**13,944**	**0**
	SLPAm	**0**	30,859	52,370	64,138	66,745	62,370	52,925	41,030	27,778	**13,944**	**0**
	SBGLL	**0**	29,399	51,738	63,698	66,511	62,338	53,167	41,070	27,806	**13,944**	**0**
	MODPSO	–	–	–	–	–	–	–	–	–	–	–
	SN2013	**0**	30,289	53,342	65,219	67,233	62,534	52,910	41,006	27,778	**13,944**	**0**
	PRE2014	134,982	122,154	109,326	96,499	83,671	70,843	58,015	45,187	32,360	19,532	6704
	FEC	141,256	121,730	113,004	98,879	84,754	70,628	56,502	42,377	28,251	14,126	**0**

War network, 2.32% for the Wiki-ele network, 2.04% for the Wiki-rfa network and 3.53% for the Slashdot network.

Fig. 13 records the variations of the balance transformations found by MLMSB with the cost parameter w. As shown in Fig. 13, there are multiple balance transformations for the GGS, War, Wiki-ele, Wiki-rfa and Slashdot networks. In general, more negative links and less positive links need to be changed of signs with the increase of w. To illustrate the balance transformations intuitively, we plot the clustering results of the GGS network with different w in Figs. 10 and 14, and the clustering results of the War network with $w=0.5$ and $w=0.4$ in Figs. 11 and 15.

Figs. 10 and 14 illustrate that it is necessary to transform 2 unbalanced positive links and 7 unbalanced negative links when $0.1 \leqslant w \leqslant 0.7$ and $0.8 \leqslant w < 1$, respectively. As shown in Fig. 10, the GGS network is divided into three clusters and the positive relations between 'MASIL' and 'UHETO', 'MASIL' and 'NAGAM' need to be transformed. As shown in Fig. 14, the GGS network is divided into two clusters and the seven negative links highlighted with black lines need to be transformed.

Fig. 11 shows that the War network is mainly divided into two clusters with 32 unbalanced positive links and 8 unbalanced negative links when $w=0.5$. Fig. 15 illustrates that the War network is mainly divided into three clusters with 35 unbalanced positive links and 7 unbalanced negative links when $w=0.4$.

It is noticed that in the GGS network, the two clusters drawn with blue[1] box and red circle in Fig. 10 are merged into the cluster drawn with blue box in Fig. 14. In the War network, the two clusters drawn with red triangle and green circle in Fig. 11 mainly split from the cluster drawn with green circle in Fig. 15. The interesting phenomenon indicates that different balance transformations may lead to the mergence and the separation of clusters in signed networks.

[1] For interpretation of color in Figs. 10 and 14, the reader is referred to the web version of this article.

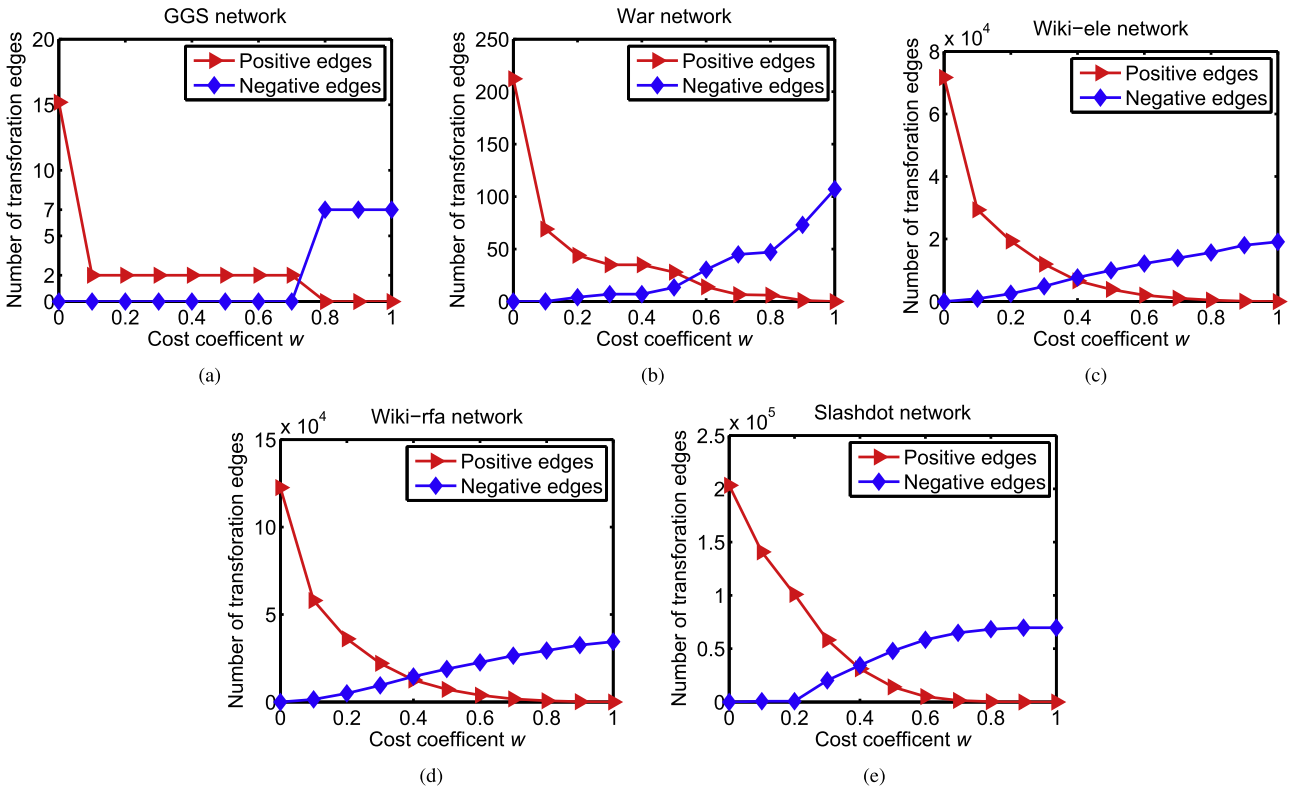


Fig. 13. Variation of the balance transformations with the cost parameter w for the (a) GGS, (b) War, (c) Wiki-ele, (d) Wiki-rfa and (e) Slashdot networks.

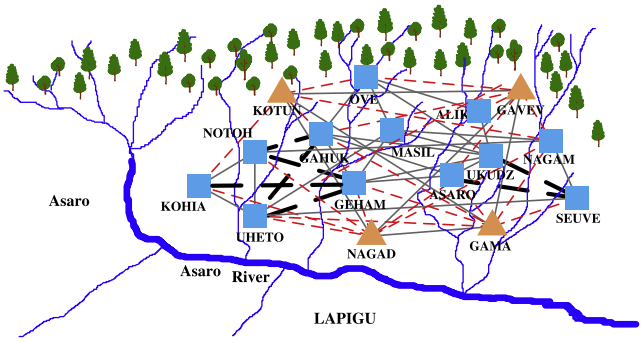


Fig. 14. Balance transformation of the War network based on the minimization of H_w with $0.8 \leqslant w < 1$.

In practical applications, if we know the unbalanced factors and the transformation cost in advance, we can reduce the potential conflicts of real systems by the transformation of the unbalanced factors. The MLMSB algorithm and transformation optimization models are devised to identify and reduce the imbalances in real-world signed systems.

4.5. Impacts of parameter settings

In our algorithm and its variations, there are six crucial parameters: the population size N_P, the parent population size N_m, the iteration number g_{max}, the pool size N_o, the crossover probability p_c and the mutation probability p_m. Fig. 16 records the variations of the statistic H values over 50 independent trials with different $N_P, N_m, g_{max}, N_o, p_c$ and p_m in the War network.

The curves in Fig. 16 show that MLMSB and TLMSB are more robust than GA and OLMSB to the parameters $N_P, N_m, g_{max}, N_o, p_c$ and p_m. Generally, the statistic H values of GA decrease as the N_P, N_m and g_{max} values increase. Note that, the computational complexity of GA is related to the N_P, N_m and g_{max} values. OLMSB is less sensitive than GA to $N_P, N_m, g_{max}, N_o, p_c$ and p_m. However, it may be trapped in local optima and cannot achieve convergence in a few iterations. The algorithms TLMSB and MLMSB are robust to $N_P, N_m, g_{max}, N_o, p_c$ and p_m and they converge to good solutions in a few generations (e.g., $g_{max} = 40$).

Fig. 16 also illustrates that GA, OLMSB, TLMSB and MLMSB have low H values when $N_P \geqslant 60, N_m \geqslant 10, g_{max} \geqslant 80, 1 \leqslant N_o \leqslant 3, 0.7 \leqslant p_c \leqslant 1$ and $0 \leqslant p_m \leqslant 0.4$. Considering both the consuming time and the performances, we set $N_p = 100, N_m = 10, g_{max} = 100, N_o = 2, p_c = 0.9$ and $p_m = 0.1$ for our method and its variations. The parameters settings in our method are the same as those in the population-based MODPSO algorithm [20].

5. Conclusions

The computation and transformation of structural balance are fundamental to unearth the potential conflicts and maintain the stability of complex systems. In this paper, we have extended the classical energy function so that it can compute the structural balance of signed networks both in strong and weak definitions. After that, the unbalanced links can be transformed by changing signs of unbalanced edges. Moreover, we have proposed a more general energy function to evaluate the transformation cost. In addition, by incorporating the neighborhoods of node, cluster and partition, we have devised a fast memetic algorithm to compute and transform structural balance in signed network. Experimental results have demonstrated the superior performance of the proposed method (MLMSB) compared with other state-of-the-art models in five real-world signed networks. The results also have illustrated the effectiveness of MLMSB on the pursuit of balance at minimum cost.

Despite of the promising performance of MLMSB, there are still some follow-up works. For instance, it is highly likely that multiple relations in complex systems are transformed to each other over

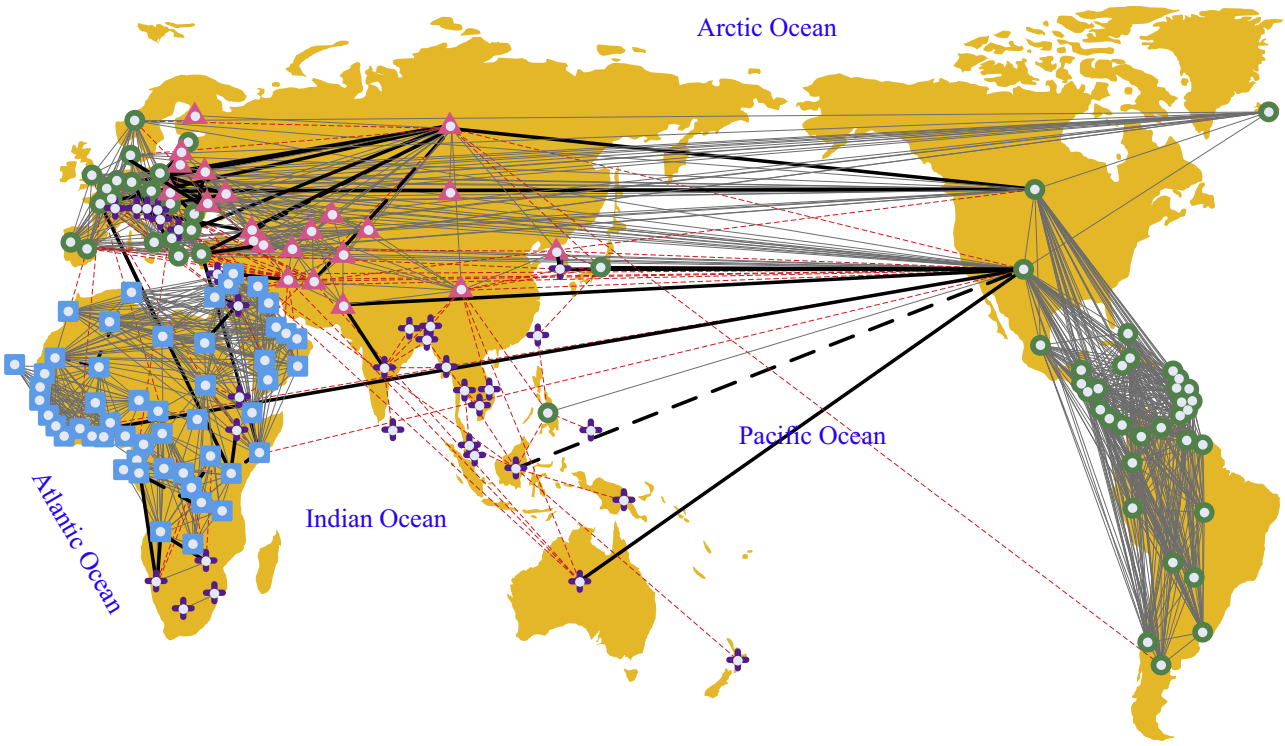


Fig. 15. Balance transformation of the War network based on the minimization of H_w with $w = 0.4$. The nodes in the clusters with less than 3 nodes are marked with cross, and the unbalanced edges are highlighted in black.

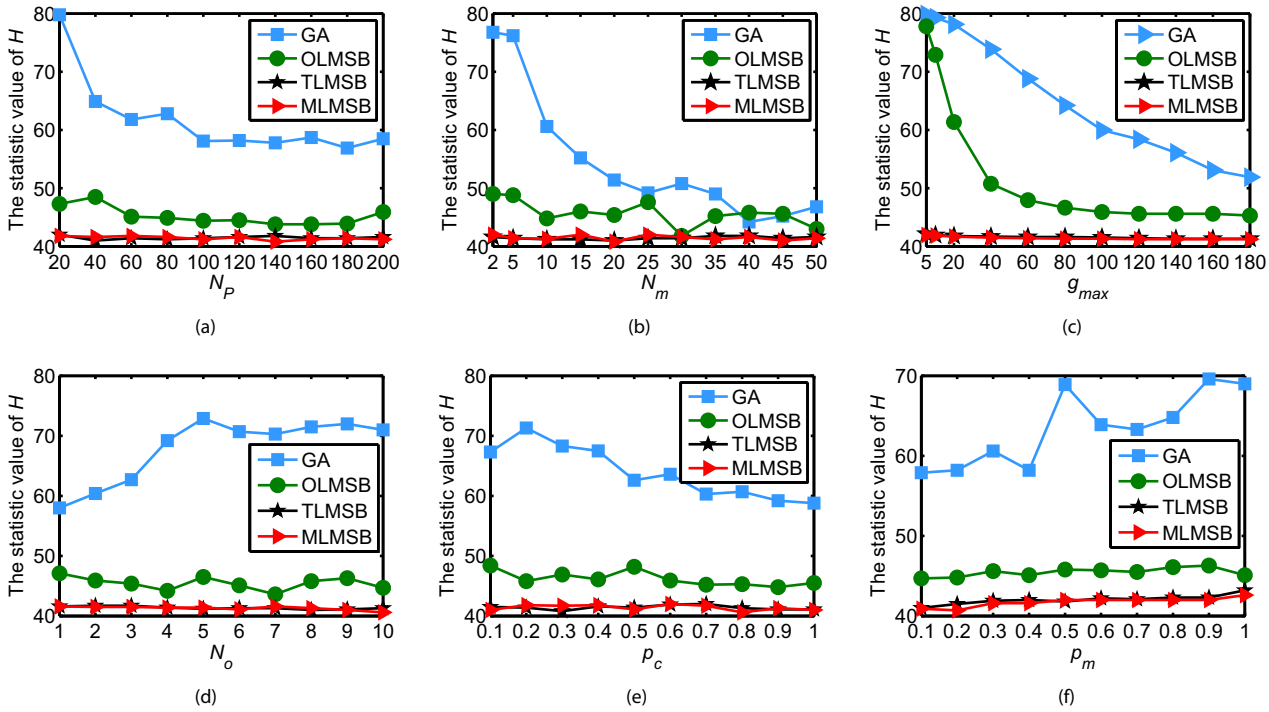


Fig. 16. Statistic values of H averaged over 50 independent runs for MLMSB and its three variations with different (a) population size N_P, (b) chosen population size N_m, (c) iteration numbers g_{max}, (d) pool size N_o, (e) crossover probability p_c and (f) mutation probability p_m.

time, i.e., dynamic properties. To cater this property, one of our future works will aim at transforming unbalanced structures to balanced ones in dynamic signed networks. Also, directed relationship may be more suitable for some real-world social systems [2,6], and therefore current work will be extended to signed directed networks.

Acknowledgements

The authors thank James Bradbury and Liyan Song for their careful modifications on the writings of this paper. The authors wish to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality. This work was supported by the National Natural Science Foundation of China (Grant Nos. 61273317, 61422209 and 61473215), Key Project from the National Science Foundation of China (Grant No. 12AZD110), National Program for Support of Top-notch Young Professionals of China, the Specialized Research Found for the Doctoral Program of Higher Education (Grant No. 20130203110011) and the Fundamental Research Fund for the Central Universities.

References

[1] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning about a Highly Connected World, Cambridge University Press, Cambridge, UK, 2010.
[2] M. Szell, R. Lambiotte, S. Thurner, Multirelational organization of large-scale social networks in an online world, Proc. Natl. Acad. Sci. 107 (31) (2010) 13636–13641.
[3] Y. Kim, M.A. Ahmad, Trust, distrust and lack of confidence of users in online social media-sharing communities, Knowl.-Based Syst. 37 (2013) 438–450.
[4] G. Palla, A.-L. Barabási, T. Vicsek, Quantifying social group evolution, Nature 446 (7136) (2007) 664–667.
[5] S.A. Marvel, S.H. Strogatz, J.M. Kleinberg, Energy landscape of social balance, Phys. Rev. Lett. 103 (19) (2009) 198701.
[6] J. Leskovec, D. Huttenlocher, J. Kleinberg, Signed networks in social media, in: CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 2010, pp. 1361–1370.
[7] G. Facchetti, G. Iacono, C. Altafini, Computing global structural balance in large-scale signed social networks, Proc. Natl. Acad. Sci. 108 (52) (2011) 20953–20958.
[8] S.A. Marvel, J. Kleinberg, R.D. Kleinberg, S.H. Strogatz, Continuous-time model of structural balance, Proc. Natl. Acad. Sci. 108 (5) (2011) 1771–1776.
[9] V.A. Traag, P. Van Dooren, P. De Leenheer, Dynamical models explaining social balance and evolution of cooperation, PloS One 8 (4) (2013) e60063.
[10] E. Estrada, M. Benzi, Walk-based measure of balance in signed networks: detecting lack of balance in social networks, Phys. Rev. E 90 (4) (2014) 042802.
[11] F. Heider, Attitudes and cognitive organization, J. Psychol. 21 (1) (1946) 107–112.
[12] N. Rovira-Asenjo, T. Gumí, M. Sales-Pardo, R. Guimera, Predicting future conflict between team-members with parameter-free models of social networks, Scientific Rep. 3 (1999) (2013) 1–6.
[13] S. Zhang, L. Chen, D. Hu, Y. Guo, Impacts of opinion propagation on social balance, J. Modern Phys. 4 (2) (2013) 130–136.
[14] J. Kunegis, A. Lommatzsch, C. Bauckhage, The slashdot zoo: mining a social network with negative edges, in: Proceedings of the 18th International Conference on World Wide Web, ACM, New York, NY, USA, 2009, pp. 741–750.
[15] G. Iacono, F. Ramezani, N. Soranzo, C. Altafini, Determining the distance to monotonicity of a biological network: a graph-theoretical approach, IET Syst. Biol. 4 (3) (2010) 223–235.
[16] Y. Sun, H. Du, M. Gong, L. Ma, S. Wang, Fast computing global structural balance in signed networks based on memetic algorithm, Physica A 415 (1) (2014) 261–272.
[17] P. Esmailian, S.E. Abtahi, M. Jalili, Mesoscopic analysis of online social networks: the role of negative ties, Phys. Rev. E 90 (4) (2014) 042817.
[18] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. 105 (4) (2008) 1118–1123.
[19] B. Yang, W.K. Cheung, J. Liu, Community mining from signed social networks, IEEE Trans. Knowl. Data Eng. 19 (10) (2007) 1333–1348.
[20] M. Gong, Q. Cai, X. Chen, L. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, IEEE Trans. Evol. Comput. 18 (1) (2014) 82–97.
[21] X. Zheng, D. Zeng, F.-Y. Wang, Social balance in signed networks, Inform. Syst. Front. 10 (2014) 1–19.
[22] T. Antal, P. Krapivsky, S. Redner, Dynamics of social balance on networks, Phys. Rev. E 72 (3) (2005) 036121.
[23] T. Antal, P.L. Krapivsky, S. Redner, Social balance on networks: the dynamics of friendship and enmity, Physica D: Nonlinear Phenom. 224 (1) (2006) 130–136.
[24] P. Doreian, A. Mrvar, Partitioning signed social networks, Social Netw. 31 (1) (2009) 1–11.
[25] P. Doreian, P. Lloyd, A. Mrvar, Partitioning large signed two-mode networks: problems and prospects, Social Netw. 35 (2) (2013) 178–203.
[26] K. Kułakowski, P. Gawroński, P. Gronek, The heider balance: a continuous approach, Int. J. Modern Phys. C 16 (05) (2005) 707–716.
[27] G. Facchetti, G. Iacono, C. Altafini, Exploring the low-energy landscape of large-scale signed social networks, Phys. Rev. E 86 (3) (2012) 036116.
[28] Y.S. Ong, A. Keane, Meta-lamarckian learning in memetic algorithms, IEEE Trans. Evol. Comput. 8 (2) (2004) 99–110.
[29] C. Blum, J. Puchinger, G.R. Raidl, A. Roli, Hybrid metaheuristics in combinatorial optimization: a survey, Appl. Soft Comput. 11 (6) (2011) 4135–4151.
[30] X. Chen, Y.-S. Ong, M.-H. Lim, K.C. Tan, A multi-facet survey on memetic computation, IEEE Trans. Evol. Comput. 15 (5) (2011) 591–607.
[31] E.R. Schneider, R.A. Krohling, A hybrid approach using topsis, differential evolution, and tabu search to find multiple solutions of constrained non-linear integer optimization problems, Knowl.-Based Syst. 62 (2014) 47–56.
[32] M. Gong, B. Fu, L. Jiao, H. Du, Memetic algorithm for community detection in networks, Phys. Rev. E 84 (5) (2011) 056101.
[33] L. Ma, M. Gong, J. Liu, Q. Cai, L. Jiao, Multi-level learning based memetic algorithm for community detection, Appl. Soft Comput. 19 (2014) 121–133.
[34] J. Wu, Z. Chang, L. Yuan, Y. Hou, M. Gong, A memetic algorithm for resource allocation problem based on node-weighted graphs [application notes], IEEE Comput. Intell. Mag. 9 (2) (2014) 58–69.
[35] S.N. Qasem, S.M. Shamsuddin, A.M. Zain, Multi-objective hybrid evolutionary algorithms for radial basis function neural network design, Knowl.-Based Syst. 27 (2012) 475–497.
[36] M. Zhou, J. Liu, A memetic algorithm for enhancing the robustness of scale-free networks against malicious attacks, Physica A 410 (2014) 131–143.
[37] F. Lin, C.-C. Yeh, M.-Y. Lee, The use of hybrid manifold learning and support vector machines in the prediction of business failure, Knowl.-Based Syst. 24 (1) (2011) 95–101.
[38] A. Ghanbari, S. Kazemi, F. Mehmanpazir, M.M. Nakhostin, A cooperative ant colony optimization-genetic algorithm approach for construction of energy demand forecasting knowledge-based expert systems, Knowl.-Based Syst. 39 (2013) 194–206.
[39] C. Ren, N. An, J. Wang, L. Li, B. Hu, D. Shang, Optimal parameters selection for bp neural network based on particle swarm optimization: a case study of wind speed forecasting, Knowl.-Based Syst. 56 (2014) 226–239.
[40] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, Cambridge university press, Cambridge, UK, 1994.
[41] M. Tasgin, A. Herdagdelen, H. Bingol, Community detection in complex networks using genetic algorithms, arXiv:0711.0491v1, 2007.
[42] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, L. Chen, Quantitative function for community detection, Phys. Rev. E 77 (3) (2008) 036109.
[43] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[44] K.E. Read, Cultures of the central highlands, new guinea, Southwestern J. Anthropol. 10 (1) (1954) 1–43.
[45] F. Ghosn, G. Palmer, S.A. Bremer, The mid3 data set, 1993–2001: procedures, coding rules, and description, Conflict Manag. Peace Sci. 21 (2) (2004) 133–154.
[46] J.B. Michael, W.C. John, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2) (2009) 026129.
[47] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. 2008 (10) (2008) P10008.
[48] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.