A Novel Label Propagation Method for Community Detection Based on Game Theory

Mengqin Ning[1], Jun Gong[2]*, Zhipeng Zhou[3]
Philosophy School, Beijing Normal University, Beijing, China[1]
Software School, Jiangxi Normal University, Nanchang, China[2, 3]

***Abstract*—Community is a mesoscopic feature of the multi-scale phenomenon of complex networks, which is the bridge to revealing the formation and evolution of complex networks. Due to high computational efficiency, label propagation becomes a topic of considerable interest within community detection, but its randomness yet produces serious fluctuations. Facing the inherent flaws of label propagation, this paper proposes a series of solutions. Firstly, this paper presents a heuristic label propagation algorithm named Label Propagation Algorithm use Cliques and Weight (LPA-CW). In this algorithm, labels are expanded from seeds and propagated based on node linkage index. Seeds are produced from complete subgraph, and node linkage index is related to neighboring nodes. This method can produce competitive modularity Q but not Normalized Mutual Information (NMI), and compensate with existing methods, such as Stepping Community Detection Algorithm based on Label Propagation and Similarity (LPA-S). Secondly, in order to combine the advantages of different algorithms, this paper introduces a game theory framework, design the profit function of the participant algorithms to attain Nash equilibrium, and build an algorithm integration model for community detection (IA-GT). Thirdly, based on the above model, this presents an algorithm, named Label Propagation Algorithm based on IA-GT model (LPA-CW-S), which integrates LPA-CW and LPA-S and solves the incompatibility between modularity and NMI. Fully tested on both computer-generated and real-world networks, this method gives better results in indicators such as modularity and NMI than existing methods, effectively resolving the contradiction between the theoretical community and the real community. Moreover, this method significantly reduces the randomness and runs faster.**

Keywords*—Community detection; label propagation; node linkage; complete subgraph; game theory

I. Introduction

Human beings are surrounded by systems that are unprecedentedly complicated. Behind each complex system, there is an intricate network that encodes the interactions among these system components. Among those networks, the most influential ones are social networks, communication networks, world wide web and cognitive neural networks etc. [1], which are characterized by small-world [2], scale-free [3], community structure [4]. Since Newman [4] put forward the problem of complex network community structure in 2002, domestic and foreign scholars have devoted themselves to studying the community nature of networks and proposed a large number of community discovery algorithms, which can be broadly divided into bottom-up community discovery and top-down community discovery [5]. The bottom-up method can efficiently expand the community gradually from nodes based on heuristic rules, which can be divided into three categories: modularity optimization class, local extension class and label propagation class. With the outstanding algorithm efficiency, the label propagation community discovery method has been widely concerned. However, the randomness of the existing methods cannot guarantee the stable and reliable results of community division, and each algorithm has its own advantages and disadvantages, so it cannot adapt to all scenarios alone.

Inspired by relevant theories in sociology, mathematics, biology and other fields, the idea of this paper is originated from three points. Fig. 1 is a vivid description of them.

- Inspiration 1: Social identity theory in sociology [6] and acquaintance model. In social relationship, the number of friends, the degree of intimacy and the co-neighbor relationship greatly affect the social relationship. Based on relational model in sociology, a node link relationship model is proposed in this paper, which provides a theoretical basis for label selection strategy.
- Inspiration 2: Classical game theory in mathematics and evolutionary game theory in biology [7]. There exist conflict, competition and cooperation among nodes in complex network, and their microscopic dynamic evolution mechanism can be described by game model.
- Inspiration 3: The contradiction exists between the theoretical community division and the real-world division. Community detection algorithms are mainly based on graph theory and quality function, and the results of community detection are usually refined, but small and large communities are not effective for communication, Therefore, community scale in the real world tend to be greater.

These ideas improve the initialization, propagation, and convergence processes in the new algorithm. Meantime, this paper introduces game theory to explain the cooperation and competition among nodes in complex networks, and propose an algorithmic integration model for community detection, and then a novel algorithm is proposed based on this model. Fig. 2 shows the theoretical framework and technical route of this paper. The main contributions can be summarized in the following three points:

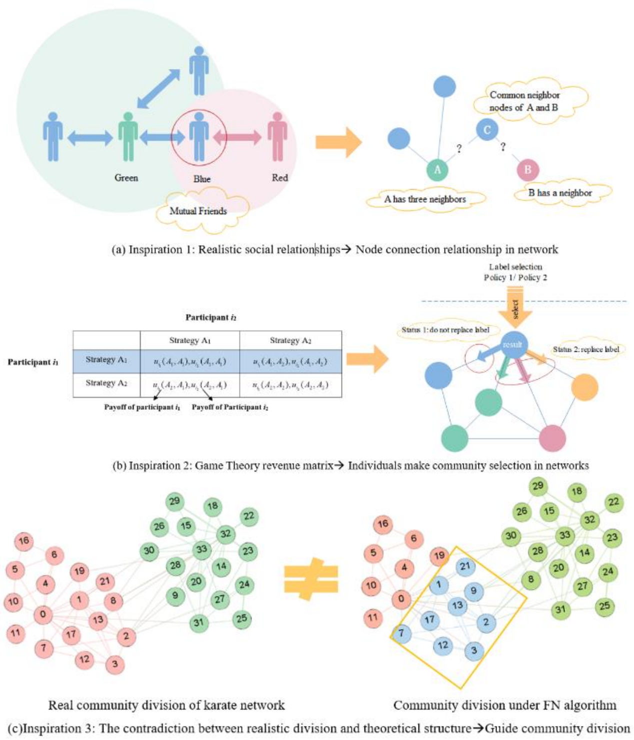

Fig. 1. Source of inspirations for this paper.

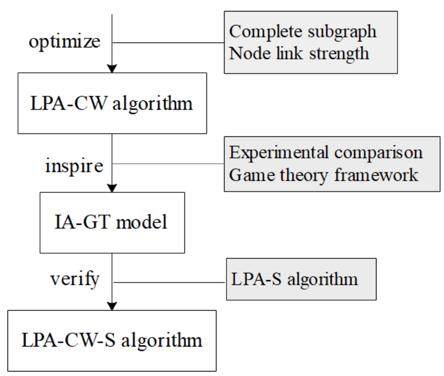

Fig. 2. The theoretical framework of this paper.

- **Contribution 1:** This paper proposes a label propagation community detection algorithm (LPA-CW) based complete subgraphs and Node Link Strength. In the initialization phase, non-overlapping complete subgraphs are introduced as the seed community, and Node Link Strength based mechanism is introduced in the propagation phase, which improves the stability and accuracy of the community division, suitable for large-scale networks.

- **Contribution 2:** This paper proposes an integrated model of community detection algorithms, IA-GT model, based on game theory. The model sets the payoff function of the participant algorithm, derives and verifies the Nash equilibrium under the mixed strategy, which can theoretically realize the complementary advantages of different algorithms and hence has strong scalability.

- **Contribution 3:** Based on this new IA-GT model, LPA-CW algorithm and LPA-S algorithm are selected as the combination objects, and then LPA-CW-S algorithm is proposed. Experiments have proved that the game mechanism effectively takes into account the contradiction between the theoretical community structure and the real community division, and performs well in metrics such as modularity and NMI, which improves the efficiency and decreases the randomness and volatility.

The remainder of this paper is organized as follows. Section II introduces the research status of label propagation algorithm and the research status of introducing game theory framework to solve the problem of community discovery. Section III and IV respectively describe the LPA-CW algorithm, IA-GT game theory model and LPA-CW-S integrated algorithm. Section V is the experimental setting and result analysis, and Section VI gives conclusions.

II. Related Work

This section introduces the research status of community detection algorithm based on label propagation and game theory, and then proposes the existing limitations and research direction.

A. Label Propagation Algorithm

In 2002, Zhu et al. first proposed the label propagation algorithm (LPA), which predicts the label information of unlabeled nodes with the labeled nodes. In 2007, Raghavan et al. [8] applied LPA to community detection for the first time. LPA can detect community structure in a near-linear time, which is greatly attractive. In 2009, Barber et al. [9] redefined LPA as an equivalent optimization problem, expanding the scope of application, and proposed the LPAm by modifying the objective function, which is applicable to both two-part network and single-part network. In 2017, Li et al. [10] proposed Stepping-LPA-S (short for LPA-S) aiming at reducing side effects of community merging by introducing a new quality function.

This search for LPA optimization has attracted much interest in recent years due to the side effects of propagation instability. Liu et al. [11] introduced node influence, network propagation update and node attribute characteristics in LPA. SUN et al. [12] transformed node partition problem into link partition problem in LPA. Kouni et al. [13] introduced node aggregation coefficient in LPA to evaluate node importance. ZHANG et al. [14], from the perspective of human society and radar transmission, defined four node capabilities (propagation, attraction, emission and reception), label influence and a novel label propagation mechanism to cope with instability and efficiency. In the field of community detection, LPA optimization is still a research hotspot, but it is trapped in homogeneity without breakthrough.

- **Limitation 1:** Most algorithms emphasize on network structure, ignoring node importance during network formation.

- **Limitation 2:** The improvement mainly focuses on the label selection stage, ignoring the overall consideration for other stages, such as initialization and convergence.

Inspired by clique percolation and sociological relationship theory, this paper optimizes most stages including initialization, propagation and convergence, and proposes a novel algorithm.

B. Game Theory

Since the publication of Von Neumann's article "Theory of Parlor Games" [15], game theory has been widely applied to psychology, economics, sociology, politics and many other fields. However, game theory rarely appears in the field of community detection. Chen et al. [16] first introduced game theory framework to solve the problem of community detection, which reflects the formation of real-world communities and detects communities by the gain and loss function. Ioana et al. [17] proposed a dynamic community detection method based on game theory elements and extremum optimization, but its experimental results had unobvious advantage compared with existing algorithms, and there were few related studies since then.

Since 2019, there has been a series of new improvements. For example, Hesamipour et al. [18] used the Adamic/Adar (AA) index to detect the local host node and extended its surrounding community based on the game theory. ZHOU et al. [19] proposed an edge weight calculation method for computing node and alliance Shapley values in combination with game theory for community detection. Kumar et al. [20] used a game theory method (Dynamic Clustering game, DCFG) to analyze the clustering problem of attribute graphs, and provided a solution for balancing topology structure and node attributes. Obviously, there are some new entry points to combine game theory with community detection, but there are still some limitations.

- **Limitation 1:** Most game theory based community detection algorithms can't get satisfactory results in benchmark experiments, and improvement bottlenecks still exist compared with classical algorithms.
- **Limitation 2:** The game selection didn't effectively solve the contradiction between the theoretical optimization of community division and the real world needs, and didn't take full advantage of game theory.

Considering the above problems, this paper views some community detection algorithms as participants, and proposes an extensible model.

III. A Community Detection Algorithm for Label Propagation Based on Complete Subgraphs and Node Link Strength

A. Problem Formulation

Given $G=(V,E)$ represents a complex network, where $V=\{v(i)\mid i=1,2,...,n\}$ represents the set of nodes in the network, and $E=\{e(i)\mid i=1,2,...,m\}$ represents the set of edges. Communities are subsets of nodes highly linked among themselves but loosely connected to the rest of the network. Communities are believed to play a central role in the functional properties of complex structures. They are represented by $C=\{C_1,C_2,...,C_K\}(1\le K\le|V|)$ and K is the number of network communities.

- **Definition 1:** Direct link strength. It denotes the direct contribution to the link strength of edge $e_{v(1),v(2)}$ of nodes $v(1)$ and $v(2)$ concerning their neighbor nodes, which is marked as $DL_{v(1),v(2)}$ and the equation is as follows:

$$DL_{v(1),v(2)}=\frac{1}{d_{v(1)}+d_{v(2)}} \tag{1}$$

In which $d_{v(1)}$ and $d_{v(2)}$ respectively represent the number of neighbor nodes of nodes $v(1)$ and $v(2)$, namely the degree. The larger the node degree is, the more important the node is, but the smaller contribution to the link intensity value.

- **Definition 2:** Indirect link strength. It denotes the indirect contribution degree of the link strength of edge $e_{v(1),v(2)}$ between nodes $v(1)$ and $v(2)$ according to the common neighbor nodes, which is marked as $IL_{v(1),v(2)}$ and the equation is as follows:

$$IL_{v(1),v(2)}=\frac{\left(|N\left(v(1)\right)\cap N\left(v(2)\right)|+1\right)}{d_{v(1)}+d_{v(2)}} \tag{2}$$

In which $N\left(v(1)\right)$ and $N\left(v(2)\right)$ denote neighbor node sets of nodes $v(1)$ and $v(2)$, respectively, $N\left(v(1)\right)\cap N\left(v(2)\right)$ represents common neighbor node sets of nodes $v(1)$ and $v(2)$. The more common neighbors between nodes are, the stronger their connectivity is.

- **Definition 3:** Node link strength. It denotes the joint contribution of nodes $v(1)$ and $v(2)$ to the link strength of edges $e_{v(1),v(2)}$ via direct link and indirect link strength, which is marked as $LS_{v(1),v(2)}$ and the equation is as follows:

$$LS_{v(1),v(2)}=\begin{cases} DL_{v(1),v(2)} & if\ d_{v(1)}=1\quad or\quad d_{v(2)}=1 \\ DL_{v(1),v(2)}+2*IL_{v(1),v(2)} & if\ d_{v(1)}>1\quad and\quad d_{v(2)}>1 \end{cases} \tag{3}$$

According to neighbor nodes between node $v(1)$ and $v(2)$, if exists, the direct link and indirect link strength are calculated, and if not, only the direct link strength is calculated.

B. Description of LPA-CW Algorithm

1) Label initialization based on non-overlapping complete subgraph: In the traditional LPA, a unique label is assigned to each node during initialization, which causes the scattered labels. In the subsequent propagation process, it is hard to converge and easy to lead to fluctuation. The node-link relationship tells us that the community structure is highly similar to complete subgraph. Therefore, identifying the non-overlapping complete subgraph first can improve label propagation.

The label initialization process is as follows, sorting all the unassigned nodes in descending order of degree, then looking for a complete subgraph starting from the node with greatest degree, and then process continuously cycles until each node have been assigned to some subgraph, finally assigning the same label to nodes in the same complete subgraph. Approximation strategy ensures algorithm efficiency, making subsequent label propagation process converge faster. The pseudocode of this procedure is as follows:

Algorithm 1 Find Nonoverlapping Complete Subgraph

```
Data: Graph
Output: nonoverlapping cliques
node ←nodes of G sorted by degree in descending order
done ← [0 for each i in node] #0:unassigned, 1:assigned
   to some clique
c ← [[] for each i in node]
t ← 0 #record the number of cliques
while node is not empty do
    n ← node[0] #node with the largest degree among
          unassigned nodes
    done[n] ← 1
    node.remove(n)
    c[t].append(n)
    for each j in neighbors of n sorted by degree in
          descending order do
        if done[j] = = 0 and j have edge(i,j) for each i in c[t]
              then
            done[j] ← 1
            node.remove(j)
            c[t].append(j)
        end if
    end for
    t++
end while
c ← c[:t]
return c
```

2) Label update strategy based on node link strength: When a node has multiple neighbor labels with the same highest frequency, one of these labels will be randomly selected as its own label, which is the traditional problem of LPA. This randomness greatly reduces accuracy and stability of community division.

To minimize randomness, this paper uses the node link strength to set a new label update strategy. Label update rules are defined as follows: (1) calculate the link strength $LS_{v(i),v(j)}$ between any two nodes in the network, according to the random access order, for the current nodes $v(i)$, find out all the node sets R_{labeln} with the same label in its neighbor $r(j)$ (according to Equation 4). Then calculate the link strength cumulative sum between node $v(i)$ and each node set R_{labeln} (according to Equation 5). Find the label $label_n$ corresponding to the accumulated and largest node set, and the node $v(i)$ label is updated to $label_n$.

$$R_{labeln} = \{r(1), r(2), ..., r(|R_{labeln}|)\}$$

$$R = \{R_{label1}, R_{label2}, ..., R_{labeln}\} \tag{4}$$

$$L = \{\sum_{j=1}^{|R_{label1}|} LS_{v(i),r(j)\in R_{label1}}, \sum_{j=1}^{|R_{label2}|} LS_{v(i),r(j)\in R_{label2}}, ..., \sum_{j=1}^{|R_{labeln}|} LS_{v(i),r(j)\in R_{labeln}}\} \tag{5}$$

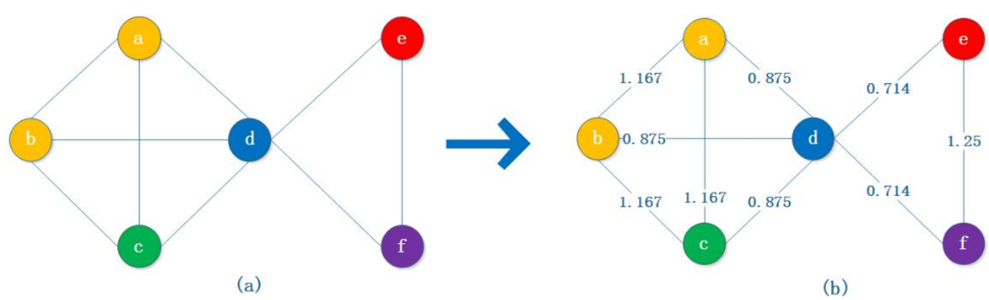

Fig. 3. A simple example for node link strength.

As shown in Fig. 3, calculate node link strength of $v(a)$ - $v(b)$ and $v(a)$ - $v(d)$ according to Equation 3. The calculation steps for the above two are as follows: $LS_{v(a),v(b)} = 1/(3+3)+2*(2+1)/(3+3) = 1.167$, $LS_{v(a),v(d)} = 1/(3+5)+2*(2+1)/(3+5) = 0.875$. Here node color denotes node label, set current node = $v(d)$, and its neighbor node set with identical labels is $R = \{\{r(a), r(b)\}, \{r(c)\}, \{r(e)\}, \{r(f)\}\}$. Then accumulate node link strength with the same labels and the result is calculated as L={1.75， 0.875， 0.714， 0.714}, so the maximal value 1.75 is selected and node $v(d)$ is updated for the yellow label.

3) Algorithm procedure: The algorithm procedure of LPA-CW is shown in Fig. 4, and its pseudo-code is described in Algorithm 1.

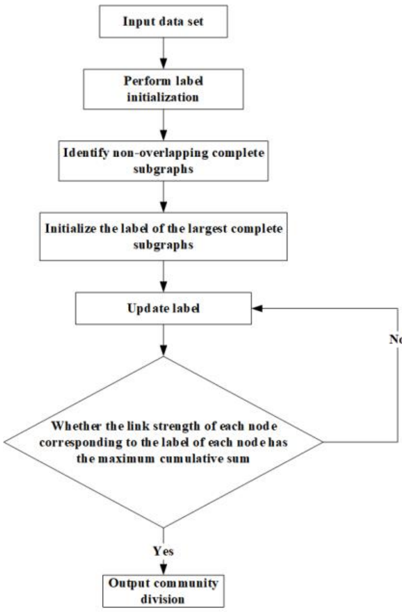

Fig. 4. Flow chart of LPA-CW algorithm.

Algorithm 1 Pseudocode of Label Propagation Algorithm use Cliques and Weight

```
Input: Graph
Output: partition
addWeight(G)
labeling ← v : k for k, v in enumerate(G)
cliques ← getCliques(G)
maxc ← maxLength(cliques)
for i in cliques do
  if length(cliques[i]) == maxc then
    maxLabel = max(i)
    for j in i do
      labeling[j] ← maxLabel
    end for
  end if
end for
while not labelCompleteEdgeWeight(labeling, G) do
  for n in G.nodes() do
    updateLabelEdgeWeight(n, labeling, G)
  end for
end while
partition ← getResult(labeling)
return partition
```

4) Algorithm analysis: Assuming that the network has n nodes and m edges, the average degree of nodes is denoted by $\bar{k}$, and the number of non-overlapping complete subgraphs searched in III*A* is denoted by λ.

The running time is mainly consumed during two stages. The first stage is the initialization phase, and it takes $O(\bar{k}n\log n)$ to identify non-overlapping complete subgraphs, then label assignment to the subgraph is $O(\lambda)$, and calculating node link strength is $O(n+2m)$. The second stage is the label propagation process, and it takes $O(n+2m)$ to access neighbor nodes, and takes $O(r\cdot(n+2m))$ to loop iteration r times. The value of r is related to data sets, when the scale of the data set increases or the average degree increases, r will also increase. Generally, r is in [3,6].

In summary, the time complexity of the new algorithm is $O(r\cdot(n+2m))$. In the subsequent experiment section, it can be proved that this algorithm is prone to faster convergence.

IV. Community Detection Algorithm Integration Model based on Game Theory

A. Problem Formulation and Basic Definitions

According to the analysis of experimental data in Section V*B* of this paper, LPA-CW algorithm has obvious advantages in modularity Q, which shows that community partition is of high quality and refinement; the comparison algorithm LPA-S [10] has complementary advantages in standard mutual information NMI, and has high accuracy in comparison with real community partition. Therefore, this paper introduces the game model to explain the individual choice and overall stability maintenance in the process of community formation. Firstly, in this paper, the combination of label propagation algorithms, namely LPA-CW algorithm and LPA-S algorithm, is selected.

The strategic game G is represented by the set $G=\{N,\{A_i\}_{i=1}^{N},\{u_i\}_{i=1}^{N}\}$, where $N=\{1,...,N\}$ is the set of participants, $A_i=\{A_1,A_2,...,A_n\}$ is the set of strategies available to the participant *i*, and u_i is the payoff of the participant *i*. When constructing the model, this paper uses the payoff matrix and mixed strategy in game theory.

B. IA-GT Community Detection Model Construction and Verification

The IA-GT (Integration Algorithm-Game Theory) community detection model is divided into three stages. Fig. 5 shows the frame diagram of the model construction. This model is an extensible model that can be flexibly replaced and combined. In the first step, the selection of participants must be based on the principle of the algorithm itself and the game theory, which has theoretical integration and practical significance. The second step is to define the payoff function of the participant according to the selected strategy and the core parameters of the algorithm. In the third step, the Nash equilibrium solution result needs to be verified by algorithm experiment.

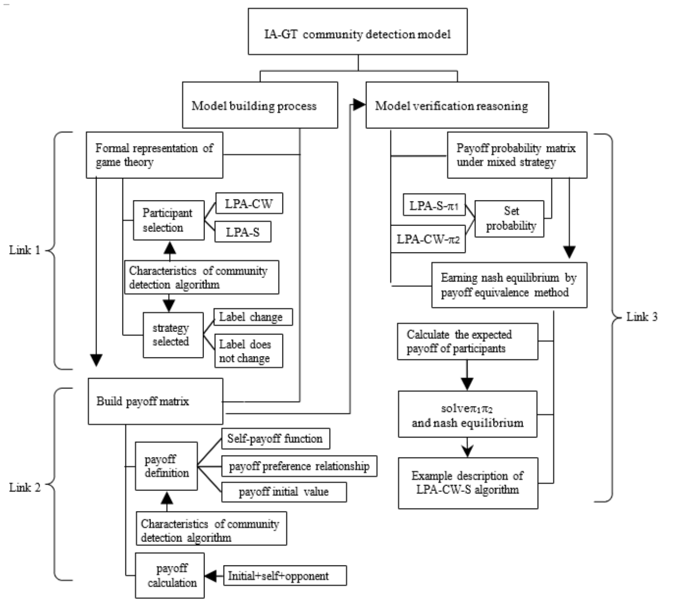

Fig. 5. IA-GT community detection model.

1) Integrated model building process: According to the game theory, this paper uses the payoff matrix to construct the model, with LPA-CW algorithm and LPA-S algorithm as two participants. Combining the core principles of the two algorithms, first initialize the network label, and then update the node label according to the game theory selection algorithm. The community label of each node has two states "changed" or "unchanged". In this model, the strategy of the two participants is to "change (c)" or " don't change (d)" the label of the current processing node. Symbolizing the above game process can be obtained:

$$G = \{\{\, LPA\text{-}S,\ LPA\text{-}CW\},\{A_{LPA-S}, A_{LPA-CW}\},\{u_{LPA-S}, u_{LPA-CW}\}\}$$

Participants: $N = \{\, LPA\text{-}S,\ LPA\text{-}CW\}$ represents two algorithms;

Strategy: $A_{LPA-S} = A_{LPA-CW} = \{c, d\}$ use c and d to indicate change and don't change.

Payoff: $u_{LPA-S}(c,c), u_{LPA-CW}(c,c)$; $u_{LPA-S}(c,d), u_{LPA-CW}(c,d)$;

$u_{LPA-S}(d,c), u_{LPA-CW}(d,c)$; $u_{LPA-S}(d,d), u_{LPA-CW}(d,d)$;

In game theory, the payoff function needs to be defined according to the problem domain. For example, for the strategy group $A(d,c)$, $u_{LPA-S}(d,c)$ is the payoff of the participant LPA-S under this set of strategies, not only related to its own strategy choice, but also related to the opponent (LPA-CW) strategy choice and initial profit value during the interaction. Combining the label update rules of LPA-S and LPA-CW, define the participant's own benefits and the preference relationship between participants, under the selected strategy.

- **Definition 4:** Participant LPA-S's own benefits under the c strategy: The maximum value of similarity between the current processing node *v*(*i*) and its neighbors minus the maximum value of similarity between node *v*(*i*) and the neighbors with the same label as node *v*(*i*), the equation denoted as I_{LPA-S} is as follows.

$$\begin{aligned} I_{LPA-S} &= \max\{S_1\} - \max(S_2) \\ S_1 &= \{s_{v(i),r(1)}, s_{v(i),r(2)}, \dots, s_{v(i),r(j)}\}, r(j) \in N\left(v(i)\right) \\ S_2 &= \{s_{v(i),r(1)}, s_{v(i),r(2)}, \dots, s_{v(i),r(h)}\}, r(h) \in R_{label\,i} \end{aligned} \quad (6)$$

Where $s_{v(i),r(1)}$ is the similarity between the node *v*(*i*) and its neighbors [21], $N\left(v(i)\right)$ is the set of all neighbors of node *v*(*i*), R_{labeli} is the set of neighbors with the same label as node *v*(*i*), S_1 is the set of similarity between node *v*(*i*) and all its neighbors *r*(*j*), S_2 is the set of similarity between node *v*(*i*) and its neighbor *r*(*h*) with the same label as node *v*(*i*). The LPA-S algorithm changes the label according to the neighbor with the greatest similarity, and the label doesn't need to be changed if it is the same. The overall value of I_{LPA-S} is between $(-1,1)$, When $I_{LPA-S} \in (-1,0]$, it is required to treat it as $I_{LPA-S} = 0$ uniformly, and the label is not changed; when $I_{LPA-S} \in (0,1)$, the greater the difference, the higher the payoff gained from label changes.

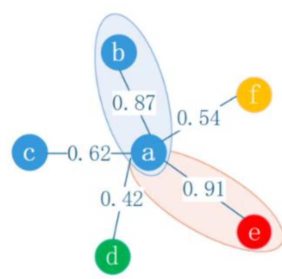

Fig. 6. Example of label selection for LPA-S.

Fig. 6 shows an example, the neighbor with the largest similarity of current processing node *v*(*a*) is node *r*(*e*). Among neighbors *r*(*b*) and *r*(*c*) that have the same label as node *v*(*a*), the similarity of *r*(*b*) is greater. According to the calculation, $S_1 = \{0.87, 0.62, 0.42, 0.91, 0.54\}$, $S_2 = \{0.87, 0.62\}$ and $I_{LPA-S} = 0.91 - 0.87 = 0.04$.

- **Definition 5:** Participant LPA-CW's own benefits under the *c* strategy: The maximum value in the cumulative sum of Link Strengths between the current processing node *v*(*i*) and each neighbor node set $R_{label\,n}$ minus the cumulative sum of Link Strengths between the node *v*(*i*) and its neighbors with the same label, the equation denoted as I_{LPA-CW} is as follows.

$$I_{LPA-CW} = \max\{L\} - \sum_{h=1}^{|R_{labeli}|} LS_{v(i),r(h)\in R_{labeli}} \quad (7)$$

Where the Node Link Strength $LS_{v(i),r(h)}$, set L , $R_{label\,n}$ are defined in detail in section III*B2)*. The LPA-CW algorithm changes the label according to the neighbor set with the largest accumulation of Node Link Strength, and the label doesn't need to be changed if it is the same. By performing standard normalization treatment on I_{LPA-CW} to get $I_{LPA-CW}{}'$, When $I_{LPA-CW} \in (-1,0]$, the label is not changed; when $I_{LPA-CW} \in (0,1)$, the greater the difference, the higher the payoff gained from label changes.

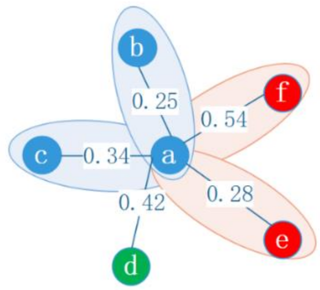

Fig. 7. Example of label selection for LPA-CW.

Fig. 7 shows an example, the current processing node *v*(*a*), all node sets with the same label in its neighbors are $R = \{\{r(b), r(c)\}, \{r(d)\}, \{r(e), r(f)\}\}$,the corresponding Node Link Strength accumulation is L={0.59， 0.42， 0.82}, $\max\{L\} = 0.82$.Where the neighbors with the same label as node *v*(*a*) are *r*(*b*) and *r*(*c*), the cumulative sum of their Node link strengths is 0.59, according to the calculation, $I_{LPA-CW} = 0.82 - 0.59 = 0.23$.

- **Definition 6:** The preference relationship between payoff of IA-GT community detection model: Under the selected strategy group, the influence of opponent's choice on the payoff of participants [22].

Combined with the idea of label propagation algorithm, the convergence result of community division is that the labels will no longer change, so it is assumed that the initial payoff value of each node is 0. When the participant chooses the "change" strategy, its own payoff is calculated according to definitions 4 and 5. At this time, if the opponent also chooses the "change" strategy, the participants' overall payoff minus 1, but if the opponent chooses the "don't change" strategy, there will be no impact. When participants choose "don't change" strategy, its own payoff is 0, at this time if the opponent choose "change" strategy, the participants' overall payoff minus 1, if the opponent also choose "don't change" strategy, the participants' overall payoff plus 1.

According to definitions 4, 5, and 6, calculate the overall payoff of the participants LPA-S and LPA-CW under the selected strategy group, and construct the payoff matrix of the IA-GT model as shown in Fig. 8.

$A(c,c)$: $u_{LPA-S}(c,c) = I_{LPA-S} - 1$; $u_{LPA-CW}(c,c) = I_{LPA-CW}{}' - 1$

$A(c,d)$: $u_{LPA-S}(c,d) = I_{LPA-S}$; $u_{LPA-CW}(c,d) = 0 - 1 = -1$

$A(d,c)$: $u_{LPA-S}(d,c) = 0 - 1 = -1$; $u_{LPA-CW}(d,c) = I_{LPA-CW}{}'$

$A(d,d)$: $u_{LPA-S}(d,d) = 0 + 1 = 1$; $u_{LPA-CW}(d,d) = 0 + 1 = 1$

LPA-CW

LPA-S		Change(*c*)	Don't change(*d*)
	Change(*c*)	$I_{LPA-S} - 1, I_{LPA-CW}{}' - 1$	$I_{LPA-S}, -1$
	Don't change(*d*)	$-1, I_{LPA-CW}{}'$	1,1

Payoff of LPA-S **Payoff of LPA-CW**

Fig. 8. Payoff matrix of IA-GT community detection model.

2) Reasoning verification of integrated model: After the payoff matrix is determined, the participants will randomly choose different strategies according to a certain probability distribution, at this time, and then Nash equilibrium under mixed strategies can be solved. Based on the integration needs of the two algorithms, this paper chooses the payoff equivalent method to calculate.

When two participants choose different strategies, for the current processing node *v*(*i*), it is assumed that the probability of community label change is π_1 when LPA-S algorithm is applied, and the probability is π_2 when LPA-CW algorithm is applied. Fig. 9 shows the payoff probability matrix of IA-GT community detection model under mixed strategy.

LPA-CW

LPA-S		Change(*c*), π_2	Don't change(*d*), $1-\pi_2$
	Change(*c*), π_1	$I_{LPA-S} - 1, I_{LPA-CW}{}'$	$I_{LPA-S}, -1$
	Don't change(*d*), $1-\pi_1$	$-1, I_{LPA-CW}{}'$	1,1

Fig. 9. Payoff probability matrix of IA-GT community detection model under mixed strategy.

According to game theory, the participants's expected payoff is evaluated by the probability of opponents choosing strategies, that is, the expected payoff of LPA-CW algorithm is evaluated by π_1. For LPA-CW algorithm:

Expected payoff from choosing "change" strategy: $Eu_{LPA-CW}(c) = (I_{LPA-CW}{}' - 1) \times \pi_1 + I_{LPA-CW}{}' \times (1 - \pi_1)$;

Expected payoff from choosing "don't change" strategy: $Eu_{LPA-CW}(d) = -1 \times \pi_1 + 1 \times (1 - \pi_1)$;

According to the Nash Equilibrium Payoff Equivalence Method:

$Eu_{LPA-CW}(c) = Eu_{LPA-CW}(d)$, $\pi_1 = 1 - I_{LPA-CW}{}'$;

The same can be obtained $\pi_2 = 1 - I_{LPA-S}$.

Both π_1 and π_2 are the probability when the label changes, so only the case of $I_{LPA-CW}{}', I_{LPA-S} \in (0,1)$ need to be considered, because when the two are less than 0, the label doesn't change.

According to Equation 6 and 7:

$$\pi_1 = 1-(\max\{L\} - \sum_{h=1}^{|R_{labe\ln}|} LS_{v(i),r(h)\in R_{labe\ln}})' \tag{8}$$

$$\pi_2 = 1-\left(\max\{S_1\} - \max(S_2)\right) \tag{9}$$

$$p = \{\{\pi_1, 1-\pi_1\}, \{\pi_2, 1-\pi_2\}\} \tag{10}$$

π_1 and π_2 are related to the payoff of the two participants' algorithms, and they are both solved. The result of mixed strategy game is mixed strategy Nash equilibrium, so Nash equilibrium is obtained, as shown in Equation 10. This paper combines the game theory with the community detection problem, and constructed the IA-GT community detection algorithm integration model.

C. Model Application: Description of LPA-CW-S Algorithm

In the previous section, this paper has theoretically proved the effectiveness of the IA-GT model. Through dynamic analysis, with the update of each node's label by community detection, the payoff size of algorithm implementation and strategy selection probability will also change correspondingly. According to the game payoff matrix, at this time the algorithm needs to choose the optimal strategy for processing. Fig. 10 shows an application example of the model, namely LPA-CW-S algorithm, which proves the feasibility of the model from the experimental point of view.

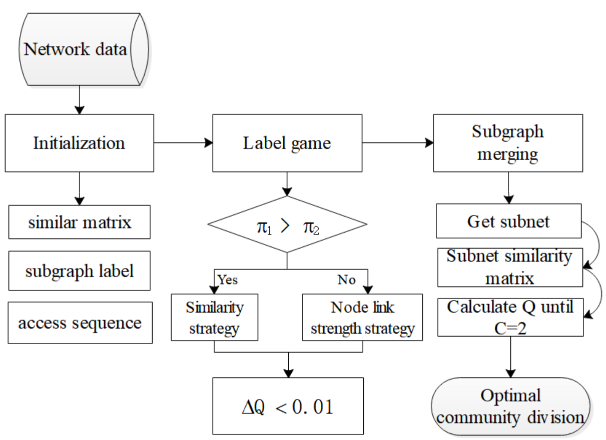

Fig. 10. The framework of LPA-CW-S.

1) Algorithm steps: According to the above theory, LPA-CW-S integration algorithm steps are as follows:

a) Initialization stage: Initialize all labels, search for the non-overlapping minimum complete subgraph in the graph, and let the clique with the largest number of nodes assign the same label. Initialize the similarity matrix, and refer to LPA-S for calculation; creating a random access sequence of nodes.

b) Label game operation stage: According to the obtained access sequence, the following values are calculated one by one: π_1, applying LPA-CW algorithm, the probability of community label change; π_2, applying LPA-CW algorithm, the probability of community label change; If $\pi_1 > \pi_2$, the label is updated by similarity strategy, otherwise, it is updated by Node Link Strength strategy. After each traversal, the modular Q is calculated once, and if the difference between the two times is less than 0.01, the first operation is finished.

c) Sub-graph merging operation stage: Preparation stage: get subnets from the current G, and then obtain the similarity matrix between subnets, save the current partition results. Operation stage: Initialize the subnet random access sequence; Calculate modularity Q once every time a subnet is updated. If Q is larger than the Q of previous partition results, save the current partition results to obtain the optimal solution, until there are two communities left.

d) Complete and return the optimal partition result: According to the above steps, the LPA-CW-S algorithm is divided into three stages. In the first stage, Algorithm 2-1 for the pseudo-code of initialization; in the second stage, Algorithm 2-2 for the pseudo-code of label game; and in the third stage, Algorithm 2-3 for the pseudo-code of subgraph merging.

Algorithm 2-1 Initialization

```
Data: A network G = (V, E)
G1 ← G.copy()
initializeLabel(G)
addWeight(G)
sDict ← initializeSimilarityMatrix(G)
nodeOrder ← initializeNodeOrder(G)
cliques ← getCliques(G)
maxc ← maxLength(cliques)
for i in cliques do
  if length(cliques[i]) == maxc then
    maxLabel = max(i)
    for j in i do
      G.nodes[i][label] ← maxLabel
    end for
  end if
end for
```

Algorithm 2-2 Propagation step one

```
state1 ← False
oldQ ← 1
while state1 == False do
  for i in nodeOrder do
    pi1 ← getPi1(G, i)
    pi2 ← getPi2(G, i)
    if pi1 ≥ pi2 then
      updateNodeLabelUseSimilarity(G, sDict, i)
    else
      updateNodeLabelUseEdgeWeight(G, i)
    end if
  end for
  partition ← getCurrentPartition(G)
  newQ ← modularity(G1, partition)
  changeQ ← abs(oldQ − newQ)
  if changeQ ≤ 0.01 then
    state1 ← True
  end if
end while
```

Algorithm 2-3 Propagation step two

Result: Communities $P = \{C1, C2, ..., Cn\}$

```
labelForNetwork ← getSubNetwork(G)
sSubDict ← initializeSubSimilarityMatrix(G)
partition ← getResult(labelForNetwork)
state2 ← False
while state2 == False do
  if len(partition) == 2 then
    break
  end if
  labelOrder ← initializeLabelOrder(labelForNetwork)
  curPartition ← getResult(labelForNetwork)
  maxQ ← modularity(G1, curPartition)
  for label in labelOrder do
    updateNetworkLabel(labelForNetwork, label)
  end for
  curPartition ← getResult(labelForNetwork)
  currentQ ← modularity(G1, curPartition)
  if currentQ ≥ maxQ then
    partition ← curPartition
  end if
  if len(labelForNetwork) == 2 then
    state2 ← True
    break
  end if
end while
return partition
```

2) Algorithm complexity analysis: According to the step analysis of LPA-CW-S algorithm, the running time is mainly used in three stages. The first stage is the initialization stage, the complexity of identifying non-overlapping complete subgraphs is $O(\bar{k}\log n)$, the complexity of initializing the label assignment to the subgraphs is $O(\lambda)$, and the complexity of initializing the similarity matrix is $O(n^2)$. The second stage is the first label propagation stage, which calculates that the Link Strength and similarity complexity of nodes are both $O(n+2m)$,the number of loop iterations is r_1 ,so the complexity is $O(r_1 \cdot (n+2m))$. The third stage is the second step of label propagation, if the number of iterations is r_2, the time complexity is $O(r_2 \cdot (n+2m))$.

Because of the influence of the network size, average degree and other factors, the final time complexity of the LPA-CW-S algorithm takes the highest value among the above three stages depending on the specific situation.

V. Experimental Results

In this paper, the algorithm is implemented in Python3.9, and the experiment is carried out on the Windows 10 desktop with a 4-core i5@2.4GHz CPU and 16G memory.The LPA-CW, LPA-CW-S, LPA [8], LPAm [23], LPA-S [10], CNM [24] algorithms will be contrasted on 10 real network and 9 artificial network, which have different parameter settings. This part analyzes the experimental results from the perspectives of community division, modularity, stability, and time efficiency to verify the superiority of the algorithm proposed in this paper.

A. Datasets and Evaluation Index

1) Real network datasets: In this paper, four commonly used labeled network data sets, such as Karate, and six unlabeled network data sets, such as Lesmis, are selected. It contains real networks with different scales of nodes and different practical application scenarios, which can comprehensively evaluate the performance of the algorithm. Its parameter characteristic are shown in the following Table I.

TABLE I. Basic Structural Parameters of Real Network

Network	**Reference**	***N***	***M***	***c***	**<*k*>**
Karate	[25]	34	78	2	4.588
Dolphins	[26]	62	159	2	5.129
Football	[4]	115	613	12	10.661
Polbooks	[27]	105	441	3	8.400
Lesmis	[28]	77	254	-	6.579
Jazz	[29]	198	2742	-	27.697
Sandi	[30]	674	613	-	1.819
Netscience	[31]	1589	2742	-	3.451
Facebook	[32]	4039	88234	-	43.691
Power	[33]	4941	6594	-	2.669

2) Artificial network: Artificial network is generated by benchmark of Lancichinetti et al. [34] LFR benchmark can generate networks with real network characteristics based personal demand. Its parameter characteristic is shown in the following Table II.

TABLE II. Parameter Description of LFR Benchmark Artificial Network Generation

Parameter	**Meaning**
N	number of nodes
k	averange degree
maxk	the maximum degree of nodes
mu	mixing parameter
t1	power law distribution index of node degree
t2	power law distribution index of community size
minc	the minimum community size
maxc	the maximum community size

Mu represents the probability that the node linking with the community outside. N represents the number of nodes. The bigger the mu, the less obvious the boundary of the community and the difficult it is to detect the community structure. LFR-1 to LFR-5 are set to N to 1000, mu to 0.1 to 0.5 arithmetic increments. LFR-6 to LFR-9 are set to 2000 to 5000 arithmetic increments, mu to 0.3. Their parameter setting is listed in the following Table III.

TABLE III. PARAMETER SETTING OF ARTIFICIAL NETWORK DATASET

Network	*N*	*k*	*maxk*	*mu*	*minc*	*maxc*
LFR-1	1000	10	40	0.1	30	60
LFR-2	1000	10	40	0.2	30	60
LFR-3	1000	10	40	0.3	30	60
LFR-4	1000	10	40	0.4	30	60
LFR-5	1000	10	40	0.5	30	60
LFR-6	2000	10	40	0.3	30	60
LFR-7	3000	10	40	0.3	30	60
LFR-8	4000	10	40	0.3	30	60
LFR-9	5000	10	40	0.3	30	60

3) Evaluation index: The evaluation indicators about the community detection mainly include the following two, both of which are scientifically evaluated and have different focuses, and can comprehensively evaluate the performance of the algorithm from many aspects such as graph theory structure and real division.

Modularity, proposed by Newman and Girvan [27,35], this evaluation index does not have a priori requirements for the internal structure of the community, and only needs to count the total number of edges inside and outside the community as shown in Equation 11:

$$Q = \frac{1}{2m}\sum_{c=1}^{n}[2lc - \frac{dc^2}{2m}] \tag{11}$$

Among the equation, c is the community number, n is the number of communities, lc is the number of edges in community c, dc is the sum of node degrees in community c, and m is the number of all edges in the entire network. The bigger the Q value, the better the effect of community division. The value of Q ranges in [-0.5, 1). When the value of Q is in [0.3, 0.7], it indicates that the quality of community clustering is great.

NMI (normalized mutual information), proposed by DanonL [36] in 2005 is generally used to measure the difference between the community structure divided by the algorithm and the result of the real community division. This indicator can evaluate the accuracy and stability of the community discovery algorithm as shown in Equation 12:

$$NMI(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} N_{ij}\log(\frac{N_{ij}N}{N_i N_j})}{\sum_{i=1}^{c_A} N_i \log(\frac{N_i}{N}) + \sum_{j=1}^{c_B} N_j \log(\frac{N_j}{N})} \tag{12}$$

Among the equation, c_A represents the number of real community divisions, c_B represents the number of algorithmic community divisions, the sum of the *i*-th row of the matrix N_{ij} is denoted as N_i, and the sum of the *j*-th column is denoted as N_j. The value range of the NMI is [0, 1], and the bigger the NMI value, it indicates that the detected community structure is closer to the real community division.

B. Experimental Result of LPA-CW

1) Real network experiment comparison: The labeled network has the real division of the community, and the algorithm performance can be compared through the Q value of the algorithm community division result and the NMI index. The unlabeled network does not have the real community division, and only the Q value can be used to compare the algorithm performance.

a) Labeled network: As can be seen from the modularity comparison curve in Fig. 11, the value of the division result of the LPA-CW algorithm on the labeled real network data set is generally higher than that of other comparison algorithms, indicating that the algorithm has the quality and stability of community division. The superiority is precisely because the sub-graph structure and node link strength guidance are added to the algorithm, which makes the divided community structure stronger and the clustering quality higher. By observing the comparison curve in the figure, it can be found that the NMI index of LPA-CW algorithm is not ideal, while the NMI index of LPA-S algorithm is much higher than other comparison algorithms. Comparing the Q value and NMI index of the two algorithms in Table IV, it can be seen that the LPA-CW algorithm and the LPA-S algorithm are complementary. This paper considers combining these two algorithms. The above-mentioned experimental phenomena and the label selection characteristics of label propagation algorithms provide experimental basis for the combination of algorithms.

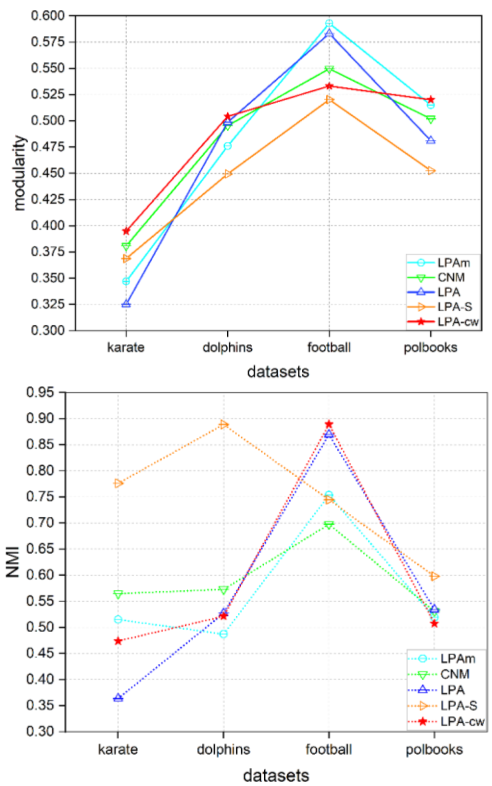

Fig. 11. Comparison of experimental results Q and NMI on labeled real network.

TABLE IV. COMPARISON OF ALGORITHM INDEX DATA ON LABELED REAL NETWORK

Network	Criterion	LPA	LPAm	LPA-S	CNM	LPA-CW(ours)
Karate	Q	0.3251	0.3470	0.3688	0.3807	**0.3949**
	NMI	0.3636	0.5150	**0.7760**	0.5646	0.4738
Dolphins	Q	0.4986	0.4760	0.4494	0.4955	**0.5042**
	NMI	0.5270	0.4870	**0.8888**	0.5727	0.5214
Football	Q	0.5831	**0.5930**	0.5203	0.5497	0.5331
	NMI	0.8697	0.7536	0.7447	0.6977	**0.8892**
Polbooks	Q	0.4811	0.5150	0.4525	0.5020	**0.5201**
	NMI	0.5341	0.5190	**0.5979**	0.5308	0.5075

b) Unlabeled network: At present, most networks in the real world are unlabeled networks and do not yet have real division results. Therefore, it is very important to continuously improve the modularity and time efficiency of community detection algorithms, which can be used to further guide community activities and behaviors in the real world. As shown in Table V, since CNM is based on modularity optimization algorithms, it has more advantages in modularity comparison, but it is also susceptible to the limitation of modularity resolution [37]. The real world network is complex and the data is huge. In most cases, the LPA-CW algorithm has obtained better community division results. The Q value is higher than that of similar label propagation algorithms and has higher time efficiency. It has great adaptability for large-scale network community detection.

TABLE V. COMPARISON OF Q RESULTS ON UNLABELED REAL NETWORK

Network	LPA	LPA-S	CNM	LPA-CW(ours)
Lesmis	0.5267	0.4492	0.5006	**0.5312 (1)**
Sandi	0.7796	0.8037	**0.9313**	0.8439 **(2)**
Jazz	0.2816	0.1719	**0.4389**	0.2822 **(2)**
Facebook	0.7369	0.6977	0.7774	**0.7885 (1)**
Power	0.6271	0.6012	**0.9346**	0.6478 **(2)**
Netscience	0.9074	0.8102	**0.9551**	0.8589 **(3)**

2) Artificial network experiment comparison: The analysis of real network experiment results has proved that compared with other classic community discovery algorithms, the LPA-CW algorithm has better community division quality and is complementary to the LPA-S algorithm. The algorithm is based on the optimization of the classic LPA algorithm, so in order to further verify the internal performance of the LPA-CW algorithm, this paper adopts the artificial network dataset of control variables (see section V*A2)* for R1-R5 parameters), and verify the LPA-CW through experiments compared with the LPA algorithm, whether the LPA-CW algorithm can effectively reduce the randomness and instability of label selection in the process of label propagation.

As shown in Fig. 12 and Table VI, under the same conditions, the modularity Q and the NMI index are decreasing when the mu is from 0.1 to 0.5. The greater the coincidence, the more difficult it is to identify the characteristics of the community structure. The modularity Q and NMI index obtained by the division results of the LPA-CW algorithm are mostly higher than those of the LPA algorithm. At the same time, it also solves the problem of the lower resolution of the LPA algorithm as the community boundary in the network becomes less obvious. The improvement measures of the algorithm on the LPA algorithm have obvious effects, and the resolution of network recognition with unobvious community boundaries has been improved.

C. Analysis of Experimental Results of LPA-CW-S Algorithm

Based on the experiment in Section V*B*, it can be seen that the LPA-CW algorithm has a higher modularity and a lower NMI index and the LPA-S algorithm has a higher NMI index and a lower modularity, because LPA-CW-S algorithm is obtained by combining game theory model. The experiment in this section will prove whether the performance of the algorithm after the combination is improved based on both, and the advantages are complementary.

As shown in Fig. 13 and Table VII, it can be found that the LPA-CW-S algorithm that combines the two games has shown great results in terms of modularity Q value. The modularity of the Dolphins and Football datasets is higher than that of the LPA-CW algorithm. The modularity of the other two data sets is also almost close to the LPA-CW algorithm, and higher than the LPA-S algorithm, which verifies that the probability game of adding similarity and node link strength in the label propagation process is preferential. Similarly, in terms of NMI index, the LPA-CW-S algorithm that combines the two games is much higher than the LPA-S algorithm and the LPA-CW algorithm on the Football dataset. On the Polbooks dataset, the NMI has reached LPA-S algorithm level, the NMI on the other two data sets is also significantly improved compared to the LPA-CW algorithm, verifying that the subgraph merging in the second stage of the algorithm can make the final community division result better.

TABLE VI. COMPARISON OF ALGORITHM INDEX DATA ON ARTIFICIAL NETWORK

Network	Criterion	LPA	LPA-CW(ours)
R1	Q	**0.8330**	0.8321
	NMI	0.9712	**0.9848**
R2	Q	0.7100	**0.7378**
	NMI	0.9230	**0.9878**
R3	Q	0.5713	**0.6242**
	NMI	0.8041	**0.9424**
R4	Q	**0.4892**	0.4363
	NMI	0.7662	**0.7684**
R5	Q	0	**0.3470**
	NMI	0	**0.6991**

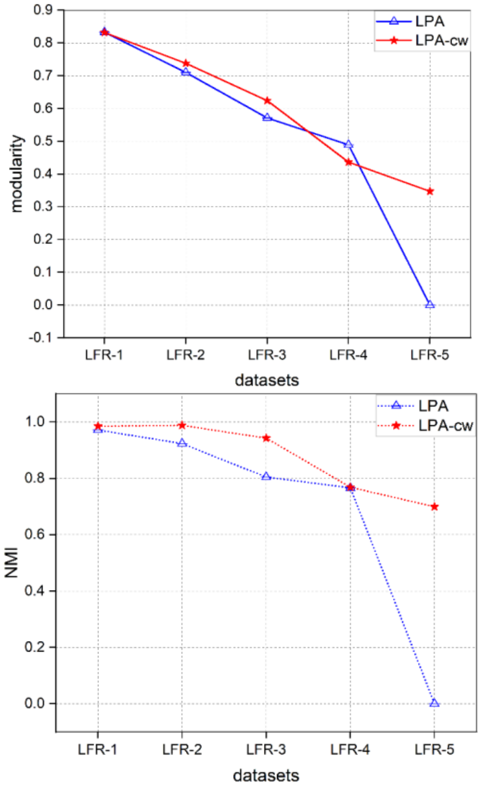

Fig. 12. Comparison of Q and NMI when N=1000 and μ changes from 0.1 to 0.5.

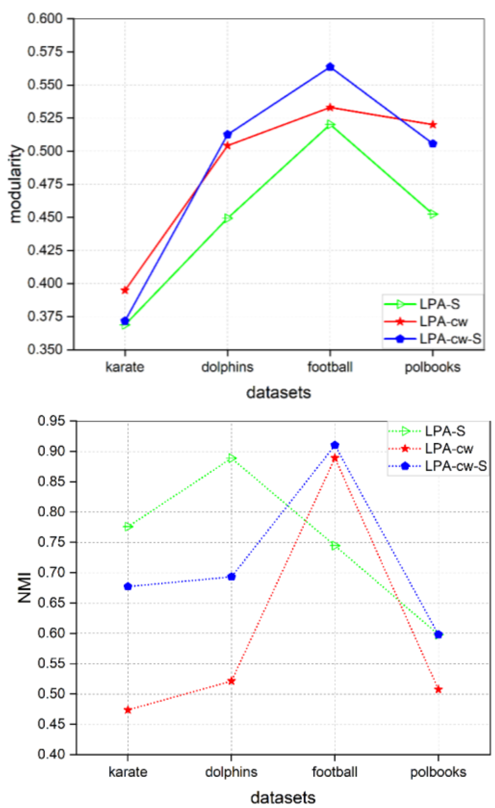

Fig. 13. Comparison of experimental results Q and NMI on real network.

TABLE VII. COMPARISON OF ALGORITHM EVALUATION INDEX RESULTS ON REAL NETWORK

Network	Criterion	LPA-S	LPA-CW	LPA-CW-S(GT)
Karate	Q	0.3688 (3)	**0.3949 (1)**	0.3718 (2)
	NMI	**0.7760 (1)**	0.4738 (3)	0.6772 (2)
Dolphins	Q	0.4494 (3)	0.5042 (2)	**0.5126 (1)**
	NMI	**0.8888 (1)**	0.5214 (3)	0.6932 (2)
Football	Q	0.5203 (3)	0.5331 (2)	**0.5637 (1)**
	NMI	0.7447 (3)	0.8892 (2)	**0.9102 (1)**
Polbooks	Q	0.4525 (3)	**0.5201 (1)**	0.5056 (2)
	NMI	**0.5979 (1)**	0.5075 (3)	**0.5979 (1)**

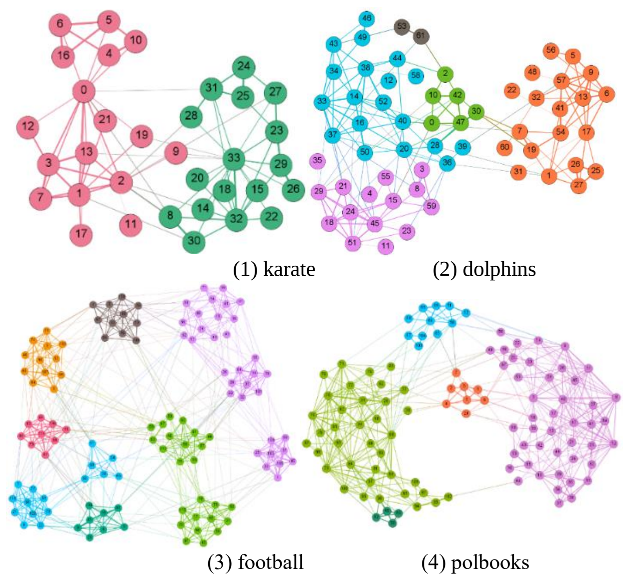

Fig. 14. Community division results of real network under LPA-CW-S algorithm.

Fig. 14 shows the visualization of the algorithm results. The network partition is complete, the clustering effect is obvious and clear, the nodes within the community are tightly connected, and the nodes between the communities are sparsely connected. From the perspective of the community structure, the division results obtained by the LPA-CW-S algorithm are reasonable and effective.

In summary, in the comparison of the overall modularity Q of the above dataset and the NMI, the game theory LPA-CW-S algorithm is more global than the two participant algorithms. The experiment proves the feasibility and rationality of the IA-GT model in this paper. It shows that game selection and subgraph merging can well neutralize the contradiction between the theoretical community structure and the actual community division, which helps to improve the accuracy and rationality of the community division results.

D. Analysis of Stability

In statistical description, variance [38] is an important equation in statistics, used to measure the stability of a set of data, the smaller the variance, the more stable the set of data, on the contrary, the more unstable the set of data. As shown in Equation 13.

TABLE VIII. COMPARISON OF ALGORITHM STABILITY DATA

	Karate	Dolphins	Football	Polbooks	Lesmis	Sandi	Jazz	Facebook	Power	Netscience
LPA-S	0.0003	0.0010	0.0004	0.0009	0.0002	0.0003	0.0012	0.0096	0.0125	0.0073
LPA-CW	0	0	0	0	0	0	0	0	0	0
LPA-CW-S	0.0002	0.0002	0.0003	0.0006	0.0001	0.0001	0.0009	0.0013	0.0078	0.0065

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(\bar{x} - x_i)^2 \tag{13}$$

In the above equation, s^2 represents the variance, x_i represents the modularity value during the *i*-th run and $\bar{x}$ represents the average value of this group of modularity Q values.

Considering the combination of the LPA-CW algorithm and the LPA-S algorithm, this article evaluates the stability of the algorithm partitioning results. Table VIII shows the variance of the modularity Q calculated by the LPA-S algorithm, the LPA-CW algorithm, and the LPA-CW-S integrated algorithm running on the real network dataset for 17 times on average.

Table VIII shows that the stability of the LPA-S algorithm is worse than that of the LPA-CW algorithm proposed in this article. The LPA-CW-S algorithm after the game combination neutralizes the algorithm of the two participants, and the stability is improved compared with the LPA-S algorithm. As the number of nodes continues to increase and the scale of the network continues to increase, the stability improvement becomes more obvious. The above experimental results prove that through the combination of game theory framework while retaining the advantages of algorithm community division, it also reduces the instability and volatility of the algorithm.

E. Analysis of Time Efficiency

Label propagation algorithms are widely used due to their close to linear time complexity. To understand the time efficiency of the LPA-CW algorithm and the game theory LPA-CW-S algorithm proposed in this paper, Table IX shows its comparison with the comparison algorithm. The comparison of time complexity is analyzed from the perspective of orders of magnitude. The CNM algorithm is a modular optimization algorithm based on the improvement of the FN algorithm that uses the heap data structure to calculate and update the network. It is close to linear time complexity and the LPA-S algorithm time complexity is at the square level. The LPA-CW algorithm proposed in this paper is complex. The degree is also close to linear. The game theory LPA-CW-S algorithm proposed in this paper is a three-stage algorithm and its time complexity is the highest among the three stages according to the data set size, which is between linear and square.

TABLE IX. COMPARISON OF TIME COMPLEXITY OF ALGORITHMS

Algorithm	Complexity
CNM	$O(n\log^2 n)$
LPA	$O(n+m)$
LPA-S	$O(n^2)$
LPA-CW	$O(r\cdot(n+2m))$
LPA-CW-S	$O(n^2)\,/\,O(r_1\cdot(n+2m))\,/\,O(r_2\cdot(n+2m))$

At the same time, in order to further verify the superiority of the algorithm in this paper, as shown in Table X, the experimental point of proof is given. Under the same condition of mu=0.3, the number of nodes increases from 1000 to 5000. The time efficiency of the LPA-CW algorithm proposed is similar to that of the LPA algorithm and it is also close to linear time complexity. The LPA-CW algorithm is obviously more efficient than the CNM algorithm.

The time efficiency of the LPA-CW-S algorithm is between the LPA-CW and LPA-S algorithms, but it is much higher than the LPA-S algorithm. Although it is no longer linear complexity, it is better than some modular optimization algorithms. Algorithms such as the GN and FN algorithms are faster. At the same time, the experiments in the last two sections also prove that the algorithm in this paper has obvious advantages in the accuracy of community division. Therefore, the LPA-CW-S algorithm achieves a compromise between time cost and accuracy, and this computational complexity is acceptable in practice.

TABLE X. COMPARISON OF TIME EFFICIENCY DATA OF ALGORITHMS (S)

Network	CNM	LPA	LPA-S	LPA-CW	LPA-CW-S(GT)
LFR-3	0.8846	0.0747	14.7138	0.2753	1.7833
LFR-6	2.2111	0.1536	61.1946	0.5585	5.5451
LFR-7	3.8703	0.2533	130.5576	0.8487	10.826
LFR-8	6.6011	0.2942	332.1598	0.9365	18.5639
LFR-9	9.355	0.4159	430.2665	1.4511	30.6681

VI. CONCLUSION

In this paper, LPA- CW algorithm is proposed to reduce the initialization time of labels by identifying non-overlapping holograms. The label update strategy based on Node Link Strength reduces the randomness in label propagation, and improves the accuracy of community division results. Combined with game theory, this paper proposes an IA-GT community detection algorithm integration model to simulate individual community selection behavior in complex networks. From a new modeling point of view, this paper reasonably explains the individual's choice and the overall stability maintenance in the process of community formation. This paper also puts forward LPA-CW-S algorithm for model verification, experiments show that the contradiction between theoretical community structure and real community division can be well neutralized by game selection, which can reduce the volatility and randomness of the algorithm, and improve the time efficiency. This compromise strategy will better adapt to the real community detection application scenario.

Through the double verification of theoretical model and experimental algorithm, this paper holds that complex networks and game theory are naturally combinable. The future work is as follows:

- Optimize the balance of various evaluation indexes through game theory, and explore the combination of more community detection algorithms by combining the integrated model of community detection algorithms (IA-GT) proposed in this paper, and extend it to the field of overlapping community detection.
- Apply game theory to other directions in complex networks, such as evolutionary networks. Or it can be applied to spatio-temporal dynamic network [39] in combination with spatio-temporal location, providing solutions for more practical application scenarios.

REFERENCES

[1] A. L. Barabási, Network science. Cambridge University Press, New York, 2014.

[2] J. S. Kleinfeld, "The small world problem," Society, vol. 39, pp. 61–66, 2002.

[3] A. L. Barabási, R. Albert, "Emergence of scaling in random networks," Science, vol. 286, no. 5439, pp. 509-512, 1999.

[4] M. Girvan, M. E. J. Newman, "structure in social and biological networks," PNAS, vol. 99, no. 12, pp. 7821-7826, 2002.

[5] S. Souravlas, A. Sifaleras, S. Katsavounis, "A classification of community detection methods in social networks: a survey," International Journal of General Systems, vol. 50, no. 1, pp. 63-91, 2021.

[6] H. Taijfel, "Experiments in intergroup discrimination," Scientific American, vol. 223, no. 5, pp. 96-102, 1970.

[7] S. Z. Guo, Z. M. Lu, "The basic theory of complex network," Science Press, Beijing, pp. 292-303, 2012.

[8] U. N. Raghavan, R. Albert, S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," Physical review E., vol. 76, 036106, 2007.

[9] M. J. Barber, J. W. Clark, "Detecting network communities by propagating labels under constraints," Physical review E., vol. 80, 026129, 2009.

[10] W. Li, C. Huang, M. Wang, X. Chen, "Stepping community detection algorithm based on label propagation and similarity," Physica A, vol. 472, pp. 145–155, 2017.

[11] S. C. Liu, F. X. Zhu, L. Gan, "Overlapping community discovery algorithm based on label propagation probability," Journal of Computer Science, vol. 39, no. 04, pp. 717-729, 2016.

[12] H. L. Sun, J. Liu, J. B. Huang, G. T. Wang, X. L. Jia, Q. B. Song, "LinkLPA: A Link-Based label propagation algorithm for overlapping community detection in networks," Computational Intelligence, vol. 33, no. 2, pp. 308-331, 2017.

[13] I. B. E. Kouni, W. Karoui, L. B. Romdhane, "Node importance based label propagation algorithm for overlapping community detection in networks," Expert Systems With Applications, vol. 162, 113020, 2019.

[14] Y. Zhang, Y. Liu, J. Zhu, C. Yang, W. Yang, S. Zhai, "NALPA: A node ability based label propagation algorithm for community detection," IEEE Access, vol. 8, pp. 46642-46664, 2020.

[15] J. V. Neumann, O. Morgenstern, "Theory of Game and Economic Behavior," Journal of the American Statistical Association, New York, 1944.

[16] W. Chen, Z. M. Liu, X. R. Sun, Y. J. Wang, "A game-theoretic framework to identify overlapping communities in social networks," Data Mining and Knowledge Discovery, vol. 21, no. 2, pp. 224-240, 2010.

[17] R. I. Lung, C. Chira, A. Andreica, "Game theory and extremal optimization for community detection in complex dynamic networks," Plos One, vol. 9, no. 2, e86891, 2014.

[18] S. Hesamipour, M. A. Balafar, "A new method for detecting communities and their centers using the Adamic/Adar Index and game theory," Physica A, vol. 535, 122354, 2019.

[19] X. Zhou, S. Cheng, Y. Liu, "A cooperative game theory-based algorithm for overlapping community detection," IEEE Access, vol. 8, 68417-68425, 2020.

[20] M. Kumar, R. Gupta, "Overlapping attributed graph clustering using mixed strategy games," Applied Intelligence, vol. 51, pp. 5299–5313, 2021.

[21] T. Zhou, L. Lü, Y. C. Zhang, "Predicting missing links via local information," Eur. Phys. J. B., vol. 71, pp. 623–630, 2009.

[22] X. Zhang, Z. Y. Xia, S. W. Xu, J. D. Wang, "Ensemble method: Community detection based on game theory," International Journal of Modern Physics B, vol. 28, no. 30, 1450211, 2014.

[23] M. J. Barber, J. W.Clark, "Detecting network communities by propagating labels under constraints," Physical review E., vol. 80, 026129, 2009.

[24] A. Clauset, M. E. J. Newman, C. Moore, "Finding community structure in very large networks," Physical review E., vol. 70, 066111, 2004.

[25] W. W. Zachary, "An information flow model for conflict and fission in small groups," Journal of Anthropological Research, vol. 33, no. 4, pp. 452–473, 1977.

[26] D. Lusseau, "The emergent properties of a dolphin social network," Royal Society, vol. 270, 0057, 2003.

[27] M. E. J. Newman, M. Girvan, "Finding and evaluating community structure in networks," Physical review E., vol. 69, 026113, 2004.

[28] D. E. Knuth, "The Stanford GraphBase: A platform for combinatorial computing," ACM-SIAM symposium, pp. 41-43, 1993.

[29] P. M. Gleiser, L. Danon, "Community structure in jazz," Advances in Complex Systems, vol. 6, pp. 565–573, 2003.

[30] V. Batagelj, A. Mrvar, Pajek datasets, 2006.

[31] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," Physical review E., vol. 74, 036104, 2006.

[32] J. McAuley, J. Leskovec, "Learning to discover social circles in ego networks," NIPS, pp. 539-547, 2012.

[33] D. J. Watts, S. H. Strogatz, "Collective dynamics of small-world networks," Nature, vol. 393, pp. 440–442, 1998.

[34] A. Lancichinetti, S. Fortunato, F. Radicchi, "Benchmark graphs for testing community detection algorithms," Physical review E, vol. 78, no. 4, 046110, 2008.

[35] M. E. J. Newman, "Modularity and community structure in networks," PNAS, vol. 103, no. 23, pp. 8577-8582, 2007.

[36] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, “Comparing community structure identification,” Journal of Statistical Mechanics: Theory and Experiment, P09008, 2005.

[37] S. Fortunato, M. Barthélemy, “Resolution limit in community detection,” PNAS, vol. 104, no. 1, pp. 36-41, 2007,

[38] R. A. Fisher, “The Correlation between relatives on the supposition of mendelian inheritance,” Transactions of the Royal Society of Edinburgh, vol. 52, no. 2, pp. 399-433, 1919,

[39] T. Zhou, Z. K Zhang, G. R. Chen, X. F. Wang, et al., “Opportunities and challenges of complex network research,” Journal of University of Electronic Science and Technology of China, vol. 43, no. 1, pp. 1-5, 2014.