Efficient Community Detection Algorithm with Label Propagation using Node Importance and Link Weight

Mohsen Arab, Mahdieh Hasheminezhad*
Department of Computer Science
Yazd University, Yazd, Iran

***Abstract*—Community detection is a principle tool for analysing and studying of a network structure. Label Propagation Algorithm (LPA) is a simple and fast community detection algorithm which is not accurate enough because of its randomness. However, some advanced versions of LPA have been presented in recent years, but their accuracy need to be improved. In this paper, an improved version of label propagation algorithm for community detection called WILPAS is presented. The proposed algorithm for community detection considers both nodes and links important. WILPAS is a parameter-free algorithm and so requires no prior knowledge. Experiments and benchmarks demonstrate that WILPAS is a pretty fast algorithm and outperforms other representative methods in community detection on both synthetic and real-world networks. More specifically, experiments show that the proposed method can detect the true community structure of real-world networks with higher accuracy than other representative label propagation-based algorithms. Finally, experimental results on the networks with millions of links reveal that the proposed algorithm preserve nearly linear time complexity of traditional LPA. Therefore, the proposed algorithm can efficiently detect communities of large-scale social networks.**

***Keywords*—Label propagation; node importance; link weight**

I. Introduction

Many complex systems can be modelled as networks with nodes for entities and edges for the connections between them. Many real-world networks have community structure. Communities can be found in many complex systems such as social and biological networks, the internet, food webs and so on. Nodes of a community have often several characteristics in common.

By now, many different methods have been proposed for community detection. In 2002, Newman and Girvan devised a divisive algorithms using centrality indices to find community boundaries [1]. This index called edge betweenness and it refers to the number of shortest paths between all pairs of nodes that run along the edge. The edge with highest edge betweenness is removed in iterative steps until no edges remain. This process takes $O(m^2n)$ which makes it impractical to be run on the networks with more than 30,000 nodes. In 2004, a measure called modularity was introduced to evaluate a given partition of a network into communities [2]. So many methods were presented for modularity optimization [3], [4], [5]. Aside from modularity optimization, a variety of different algorithms such as graph partition-based methods [6], [7], [8] and density-based methods[9], [10] and label propagation algorithm (LPA) [11] have been presented for community detection.

Among all the community detection methods, LPA is one of the fastest algorithms. LPA algorithm is simple and its time complexity is nearly linear time. However because of randomness, the detected communities have poor stability. That is, LPA may find different communities in different runs. In some runs, small communities are merged with big ones forming "monster" communities which is a drawback of LPA [12].

The LPA can be described as follows. Initially, each node is assigned a unique numeric label. At each iterative step, each node updates its label to the most frequent label from its neighbours in a random order. When there are multiple most frequent labels, the node will randomly pick one of them. Relabeling continues until the label of each node is its most frequent label among its neighbours. Finally, the nodes with the same label are considered in the same community. In fact, there are two sources of randomness in LPA which make it unstable and inaccurate. First source is random update order of nodes and the second one is randomly selecting one label when there are multiple most frequent labels to choose.

In this paper, a novel label propagation method for community detection called WILPAS is introduced. WILPAS algorithm has two stages. Let $l(v)$ be the label of node v. In the first stage, two sources of randomness of LPA are eliminated to increase accuracy. That is, firstly, random node sequence for label updating of LPA is replace by one specific update order. Secondly, WILPAS presents a novel label updating mechanism based on both node importance and link strength which makes the second source of randomness very unlikely to happen. The first stage of WILPAS is called weighted importance label propagation algorithm (WILPA).

Resulted communities from the first stage (WILPA) might be sub-communities of real ones. Therefore, in stage two of WILPAS, detected labels of nodes during the first stage are injected as a seed into a method called LPA_d. In fact, LPA_d is the same as traditional LPA in using random update order and the traditional label updating formula, but with one difference. When half of the neighbours of a node v have label $l(v)$, LPA_d does not update its current label $l(v)$. As it will be shown later, this change can avoid possible label oscillations in stage two of WILPAS.

Extensive experimental studies demonstrate that WILPAS is a pretty fast algorithm and it can get better community

detection results comparing with several label propagation based algorithms on both synthetic and real-world networks.

This paper is structured as follows. In Section II, related works in the field are listed. Some notions are defined in Section III. In Section IV the proposed method (WILPAS) is presented. The time complexity of proposed method is stated in Section V. Experimental results of comparing the proposed method with some famous methods in this area are discussed in Section VI. Finally, conclusion is given in Section VII.

II. Related Works

In 2007, Raghaval et al.[11] proposed Label Propagation Algorithm (LPA) for community detection. LPA can be summarized as four following steps:

1) Initialize every node with a unique label.

2) Arrange the nodes in a random order.

3) For every node in that random order, set its label with the one which is the most frequent label among its neighbours.

4) If every node has a label that the maximum number of their neighbours have, then stop the algorithm; else go to step 2.

The formula of label updating for LPA is as follows:

$$l(v) = \underset{l}{\operatorname{argmax}} \sum_{u \in N^l(v)} 1 \ , \tag{1}$$

where $N^l(v)$ indicates the set of neighbours of node v with label l. This is LPA's asynchronous version. Since synchronous version has potential label oscillations as discussed in [11], this version is not considered. As discussed earlier LPA has two types of randomness. Unfortunately, randomness of LPA may result in missing small communities and even getting trivial solution in which all nodes are assigned the same label [12]. Moreover, it makes the algorithm unstable such that different communities may be detected in different runs of the algorithm.

Zhang et al. generalized LPA to weighted networks by calculating the probability value of every label [13]. The label updating formula in this case is changed as follows:

$$l(v) = \underset{l}{\operatorname{argmax}} \sum_{u \in N^l(v)} w_{vu} \ , \tag{2}$$

where w_{vu} indicates the weight of the edge between nodes v and u.

Barber and Clark proposed modularity-specialized algorithm (LPAm) to constrain the label propagation process [14]. Their algorithm is near-linear time, but it may get stuck in poor local maxima in the modularity space. To scape local maxima, Liu et al. introduced an advanced modularity-specialized label propagation algorithm called LPAm+ [15]. LPAm+ combines LPAm with multistep greedy agglomerative algorithm to get higher modularity values. Thus, LPAm+ doest not guarantee near-linear time complexity [16]. Xing et al. presented a node influence based label propagation algorithm called NIBLPA [17]. NIBLPA defines two concepts node influence and label influence for specifying node orders and label choosing mechanism respectively. Zhang et al. proposed a label propagation algorithm with prediction of percolation transition named LPAp [16]. They transformed the process of label propagation into network construction process. Using this prediction process of percolation transition, they tried to delay the occurrence of trivial solutions. Sun et al. proposed a centrality-based label propagation called CenLP [18]. They presented a new measure for computing the centrality of nodes. Based on these centrality values, one specific update order in addition to node preference values are specified in order to improve traditional LPA.

III. Terminology

Let $G = (V, E)$ be an undirected network. The number of nodes and links of G is denoted by n and m, respectively. Let d_v be the degree of node v in the network. Degrees of node v within and outside of its community are denoted by d_v^{in} and d_v^{out}, respectively. Mixing parameter μ for each node v is defined as $\frac{d_v^{out}}{d_v}$. The set of all neighbours of node v is denoted by $N(v)$. Internal and external links respectively refers to the links within and between communities.

IV. Proposed Method (WILPAS)

The proposed algorithm has two stages. At first stage, in order to increase the quality of detected communities of LPA, one specific node order for label updating and a novel formula for selecting labels for nodes is introduced. The novel formula for label updating is based on the weights of links and importance of nodes. Therefore, the first stage of the proposed algorithm using these two modifications in traditional LPA is called weighted importance label propagation algorithm (WILPA). The detected communities resulted from stage one might be sub-communities of real ones. Therefore, in stage two, found labels of nodes resulted from stage one (WILPA) will be injected as a seed into a method similar to traditional LPA. The second stage which has a slight difference with traditional LPA is called LPA_d. By presenting these two stages, the proposed method is completed. Since detected labels of WILPA algorithm are injected as a seed into LPA_d algorithm, the proposed method is called WILPAS.

A. Weighting Measure for Links

There are several normalized similarity measure to assign weights to an edge (u,v) such as cosine [19]. Cosine similarity measure between two nodes u and v is defined as follows:

$$cosine(u, v) = \frac{\mid N(u) \cap N(v) \mid}{\sqrt{\mid N(u) \mid\mid N(v) \mid}}, \tag{3}$$

Where $||$ indicates the cardinality of a set. Using cosine may result in assigning zero values to some links. Thus, instead an extended version of cosine [18] is chosen to assign non-zero weights to links. This measure is called structural similarity and is defined as follows:

$$\sigma(u, v) = \frac{\mid \Gamma(u) \cap \Gamma(v) \mid}{\sqrt{\mid \Gamma(u) \mid\mid \Gamma(v) \mid}}, \tag{4}$$

where $\Gamma(u) = N(u) \cup \{u\}$.

B. Stage One of WILPAS (WILPA)

The stage one of WILPAS (WILPA) improves traditional LPA with two modifications: The first modification is presenting one specific node order for updating labels instead of random order. The second one is presenting a novel formula for selecting new labels of nodes. This novel label updating formula considers both importance values of neighbour nodes and weights of neighbour links of a node to select its new label.

1) Specific update order: In the proposed method, nodes are rearranged such that important nodes update their labels first. The degree of each node v (i.e d_v) is chosen as its importance value. Among several nodes with equal degrees, those whose neighbours have higher degrees are more important. Thus, an extended version of importance value of each node v ($EI(v)$) is defined as follows:

$$EI(v) = d_v + \sum_{u \in N(v)} d_u \tag{5}$$

Therefore, order of nodes for label updating in the proposed method is specified in descending order of extended importance values of nodes.

2) Novel label updating formula: In WILPA, instead of selecting the most frequent label among neighbours of a node as its new label, a novel label choosing mechanism is adopted. This mechanism considers both node importance and link importance for selecting the new label.

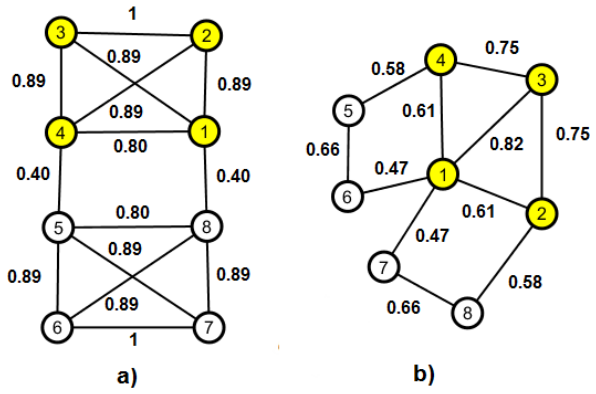

Fig. 1. Two sample networks.

Consider Fig. 1 a with two real communities $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$. Suppose that during traditional label propagation process (using label updating formula 1) nodes 1-4 have label 1 and nodes 5-8 have a label equal to their own numbers. That is, denoting label of node v by $l(v)$: $l(1) = l(2) = l(3) = l(4) = 1$ and $l(5) = 5$, $l(6) = 6$, $l(7) = 7$, $l(8) = 8$. Moreover, suppose that the update order of nodes is 8, 5, 6,7. That is, node 8 should update its label first, then node 5 and so on. Node 8 has four neighbours with labels 1, 5, 6 and 7. If node 8 selects label 1 randomly, then labels of nodes 5, 6 and 7 will be 1 as well, since label 1 will be the most frequent neighbour label for them. Thus, trivial solution (forming one big community) will be obtained.

To avoid trivial solution, the chance of propagation of labels between different communities should be decreased. One way to do this is using weights of links in the label propagation process. The idea behind using weights is that two endpoints of an internal link share more common friends to each other than two endpoints of an external link. Thus, if weight of a link is defined based on the ratio of common friends between its two endpoint nodes, then internal links are more likely to get higher weight than external ones. Thus, by considering the weights of links in label propagation process, one can expect that propagation of labels between two different communities will be less likely.

In Fig. 1(a), let this time take into account the structural similarity weight 4 of links and choose formula 2 as label updating mechanism. In this situation, the weights of the links connecting labels 1, 5, 6 and 7 to node 8 are 0.40, 0.80, 0.89 and 0.89. Therefore, node 8 will select one of two labels 6 or 7, because their corresponding weight 0.89 is maximum. Either of two labels 6 or 7 is chosen by node 8, the other three nodes 5, 6 and 7 will choose that label as well. Therefore, two real communities will be detected correctly.

However, using weighted label propagation can decrease the chance of propagation of labels between different communities, but in sparse real-world networks, this strategy may cause real communities to break apart into several sub-communities. For example, consider the network in Fig. 1(b) with one single community. Like previous example, let consider nodes 1-4 have label 1 and other nodes have label equal to their own numbers. Using weighted label propagation strategy will result in finding three communities $\{1, 2, 3, 4\}$, $\{5, 6\}$, $\{7, 8\}$. This is because the weights of two links $(5, 6)$ an $(7, 8)$ are greater than their neighbour links. Therefore, nodes 5 and 6 will choose the labels of each other. Similarly, both nodes 7 and 8 will adopt the same label 7 or 8 as their final label. Therefore, weighted label propagation strategy may divide some communities of a real-world network into several sub-communities.

To resolve the mentioned problem, one idea is to consider degrees of nodes as their importance values in label updating formula. This solution is based on this intuitive idea that in each network, there are some important nodes with high degree which play crucial role in spreading information, viral marketing, etc. Therefore, nodes with higher degrees are more likely to be centers of communities [18]. It is obvious that in social networks, a famous person or a celebrity with more friends and connections has more impact on each of his friends than a person with just a few friends.

Therefore, on the one hand, with weighted label propagation external links would have low effect in spreading labels between different communities. Thus, this idea can reduce the formation of monster communities. On the other hand, most important nodes (such as nodes with high degrees) play very crucial role in formation of communities. Therefore, the degrees of nodes should be considered in label updating formula as well. By taking into consideration both weights of links and degrees of nodes, the label updating formula of the proposed method is defined as follows:

$$l(v) = \underset{l}{\operatorname{argmax}} \sum_{u \in N^l(v)} w_{vu} * d_u \quad (6)$$

Therefore, each neighbour u of node v has an impact in defining $l(v)$ based on its importance value (d_u) and the weight of corresponding link (w_{vu}). As discussed above, the degree of each node is considered as its importance value. Therefore, the first stage of the proposed method is completed. The pseudo-code of WILPA is presented in Algorithm 1.

C. Stage Two of WILPAS (LPA_d)

Resulted communities from the first stage (WILPA) might be sub-communities of real ones. Therefore, in stage two of WILPAS, detected labels of nodes during the first stage are injected as a seed into a method similar to original label propagation algorithm. To be more accurate, LPA_d is the same as original LPA in using random update order and label updating formula, but with one difference. When half of the neighbours of a node v is the same as $l(v)$, LPA_d keeps its current label.

Consider the network in Fig. 2. WILPA algorithm as the stage one of WILPAS method detects two communities on this network which are shaded with colors green and yellow. If original LPA is applied on these found labels, final labels of two nodes 6 and 11 will be either green or yellow. This is because of the fact that two nodes 6 and 11 are connected to two different communities with equal number of links. In this situation, if LPA_d algorithm is used instead of traditional LPA, then labels of two nodes 6 and 11 will be fixed as green. Hence, LPA_d algorithm by avoiding possible label oscillations and unnecessary iterations can increase stability of detected communities and reduce the number of iterations of the proposed method.

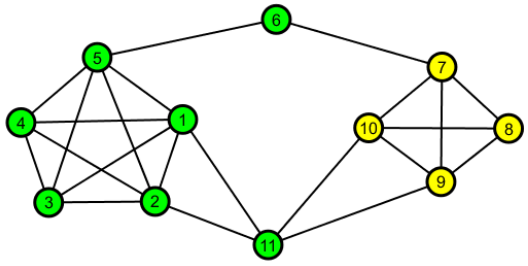

Fig. 2. A network with two detected communities by stage one of WILPAS.

D. Pseudo-code for WILPAS

Pseudo-code of WILPA, LPA_d and WILPAS are presented in Algorithms 1, 2 and 3, respectively.

V. Time Complexity

In this section, the time complexity of the proposed method is discussed.

```
1 For each node v set l(v) = v /* Initialization of labels. */
2 Arrange nodes based on descending order of their EI values
    (formula 5) and put them into array X.
3 repeat
4     foreach vertex v in X do
5         Update l(v) using novel proposed label updating
            formula 6
6     end
7 until labels of nodes do not change any more;
```

Algorithm 1: WILPA

```
1 Arrange nodes randomly and put them into array Y.
2 repeat
3     foreach vertex v in Y do
4         if |N^{l(v)}(v)| < d_v/2 then
5             Update l(v) using traditional label updating
                formula 1
6         end
7     end
8 until labels of nodes do not change any more;
```

Algorithm 2: LPA_d algorithm

```
1 Compute structural weights of all links using formula 4
    (e.g. Algorithm 4 can be used for computing weights).
2 WILPA()( Algorithm 1 ) /* Stage 1*/
3 LPA_d() (Algorithm 2) /* Stage 2*/
```

Algorithm 3: WILPAS Algorithm

A. Time complexity of weighting all links

For computing the weight of a link, the number of common friends between its two endpoints should be counted. Algorithm 4 is a simple algorithm to do that. As it can be seen from the algorithm, for computing the weight of link (v, u) it is enough to explore the set C of u's neighbours and then count the number of nodes in C which are neighbours to v as well. This is done in $O(d_u)$. Since there are d_u neighbour links for u, computing the weights of all of its neighbour links can be done in $O(d_u^2)$. Therefore, total time complexity of this simple weighting algorithm is

$$\sum_{u=1}^{n} d_u^2 \quad (7)$$

Space complexity of this algorithm using adjacency list is

$$O(m) \quad (8)$$

Checking whether two nodes z and v are neighbours can be done in $O(d_z)$ using adjacency list. But, in order to do that in $O(1)$, an extra array named 'mark' is used as follows. For each node v, at first, in line 3 of Algorithm 4, each of its neighbour u is marked as v. Then, in line 8 the adjacency of two nodes z and v is checked in $O(1)$ by comparing the content of 'mark' array of index z with v.

B. Time Complexity of Two Stages of WILPAS

In the first stage (WILPA), at first, all nodes are arranged based on their EI values. This can be done with time complexity $O(n \log n)$. Time complexity of each iteration of label

```
foreach vertex v do
    foreach neighbor u of v do
        mark[u]=v
    end
    foreach neighbor u of v do
        Cfriends=0;
        foreach neighbor z of u do
            if mark[z]==v then
                ++Cfriends;
            end
        end
        /* compute weight of edge(u,v) using equation 4 */
    end
end
```

Algorithm 4: A Simple Weighting Algorithm

updating in WILPA is the same as traditional LPA which is $O(m)$ [11]. This is because time complexity of computing new label $l(v)$ in formula 6 and formula 1 is the same. Therefore, time complexity of WILPA is

$$O(n \log n) + O(R_1 m), \tag{9}$$

where R_1 is the number of iterations of WILPA. Similarly, each iteration of LPA_d requires $O(m)$ time. Thus, time complexity of LPA_d is

$$O(R_2 m), \tag{10}$$

where R_2 is the number of iterations of LPA_d.

C. Total Time Complexity of WILPAS

Total time complexity of WILPAS is the summation of time complexities of computing weights, WILPA and LPA_d which is as follows:

$$O(n \log n) + O(R_1 m) + O(R_2 m) + O(\sum_{u=1}^{n} d_u^2) \tag{11}$$

It is important to note that in practice in most cases both R_1 and R_2 are less than 10. Moreover, real networks are often sparse, i.e. $m = O(n)$. In addition, as it will be shown in experiments section, the weighting Algorithm 4 consumes less than 25 seconds for finding the weights of all links of a network with $500,000$ nodes and around 10 million links. Therefore, as it will be demonstrated later, WILPAS is pretty fast in practice, even with existing term $\sum_{u=1}^{n} d_u^2$.

VI. Experiments

This section evaluates the effectiveness and the efficiency of the proposed algorithm. Several experiments on both synthetic networks and well-known real-world networks are conducted. Moreover, the performance of WILPAS with LPA, CenLP, LPAp, LPAm and NIBLPA are compared. All the simulations are carried out in a desktop pc with Pentium Core2, 1.8 GHZ processor and 4GB of RAM under Windows 8.1 OS.

In this paper, normalized mutual information (NMI) [20] is used as the evaluation measure which is currently widely used in measuring the quality of detected communities. NMI allows us to measure the amount of information common to two different network partitions. Accordingly, if a network has a known community structure, one can explore the efficacy of the algorithm by comparing known real partition with the partition found by that algorithm. When the found partition matches the real one, then NMI=1, and when two partitions are independent of each other, then NMI=0.

A. Test on Synthetic Networks

In this section, LFR benchmark networks [21] are chosen which are currently the most commonly used synthetic networks in community detection. The parameters of LFR benchmark networks are as follows: number of nodes n, the average degree k, maximum degree $maxk$, mixing parameter μ. Moreover, $minc$ and $maxc$ refer to the minimum and maximum values for community sizes, respectively.

Three ranges for different community sizes are used which are indicated by the letters S (stays for small), B (stays for big) and VB (stays for very big). The ranges of community sizes for three letters S, B and VB are $[cmin, cmax] = [10, 50]$, $[cmin, cmax] = [20, 100]$ and $[cmin, cmax] = [200, 1000]$, respectively. For each type of networks, 10 samples are generated and on each sample, each tested label propagation-based algorithm is run 10 times. Then, the average of these 100 NMI values are reported as output. In this paper for all the networks with $n \geq 100,000$, the average degree $k = 40$ and the letter VB are used, i.e. community sizes of these networks range between 200 and 1000 where average degree of nodes is 40.

Fig. 3 and 4 show the accuracy of the mentioned methods on the networks with size of 1000. One can observe that for $n = 1000$, when $\mu \leq 0.50$ three methods WILPAS, LPAm and CenLP find communities pretty well. However, when communities are big, for $\mu > 0.50$, LPAm gets better results (see Fig. 4).

Fig. 5 and 6 show the accuracy of methods when $n = 10,000$. From these two figures it can be observed that when $n = 10,000$, WILPAS outperforms other methods. CenLP is the second most accurate method for community detection on this network. On this network, NIBLPA shows poor performance in community detection.

Fig. 7 demonstrates the NMI results for the three most accurate tested label propagation methods i.e. WILPAS, CenLP and LPAm for a network with $n = 100,000$, $k = 40$, $[cmin, cmax] = [200, 1000]$. As it can be observed from this figure, WILPAS achieves higher NMI values than CenLP and LPAm. CenLP shows more accuracy than LPAm except for $\mu = 0.70$. The detailed information about the results is displayed in Table I.

TABLE I. NMI Results of Three Methods $WILPAS$, $CenLP$ and $LPAm$ on the Network with $n = 100,000$

μ	LPAm	CenLP	WILPAS
0.40	0.9963	1	1
0.45	0.9955	1	1
0.50	0.9946	1	1
0.55	0.9927	1	1
0.60	0.9818	0.9999	1
0.65	0.9527	0.9970	1
0.70	0.8277	0	0.9997

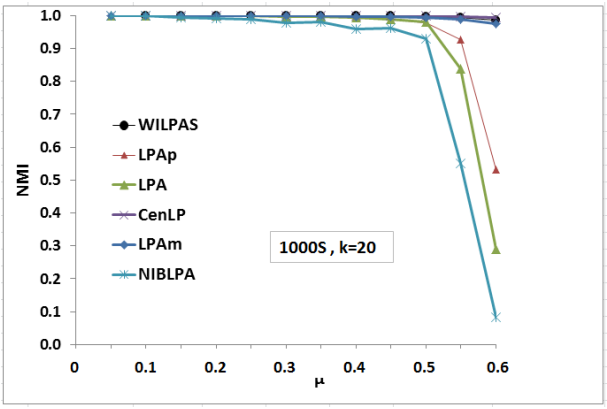

Fig. 3. Comparing different label propagation-based algorithms on the network with $n = 1,000$ where communities are small.

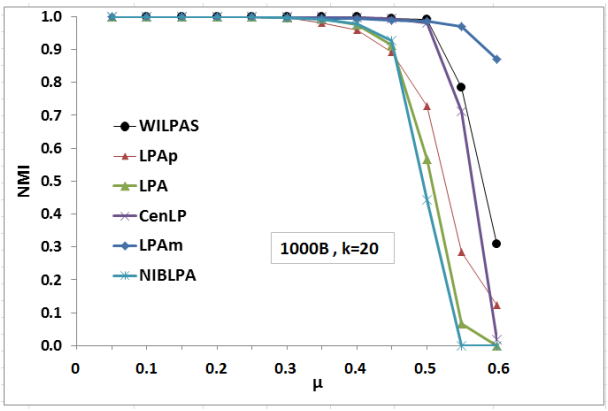

Fig. 4. Comparing different label propagation-based algorithms on the network with $n = 1,000$ where communities are big.

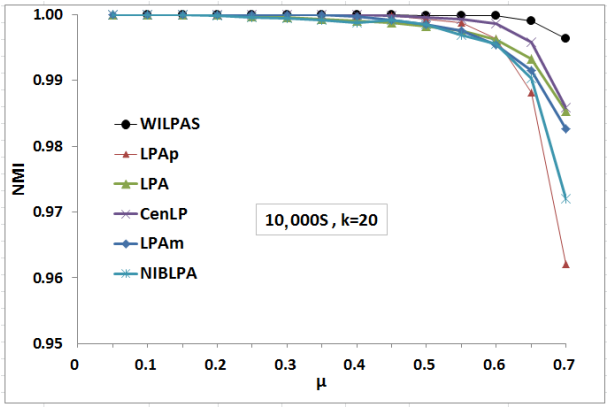

Fig. 5. Comparing different label propagation-based algorithms on the network with $n = 10,000$ where communities are small.

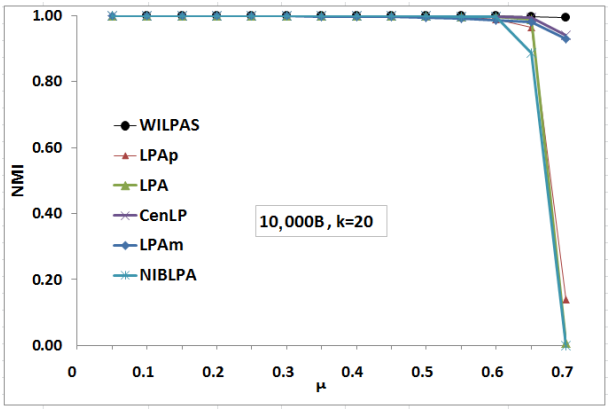

Fig. 6. Comparing different label propagation-based algorithms on the network with $n = 10,000$ where communities are big.

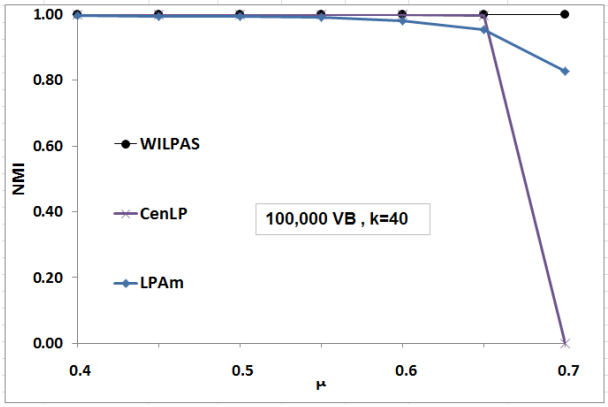

Fig. 7. Comparing different label propagation-based algorithms on the network with $n = 100,000$ where communities are very big.

B. Experiment on Real-world Networks

In this section, the evaluation of the above methods on real-world networks which their communities are already known is discussed. Zachary Karate club [22], American college football [1], Dolphin social network [23] and Polblog [24] are four famous networks in the field. The details of these networks are shown in Table II. The NMI results of all tested label propagation-based methods are displayed in Table III.

Each method is run 10 times on each real network, then the average NMI results are reported. The number in the {} for CenLP, NIBLPA and WILPAS in Table III shows the number of found communities by these three deterministic methods. Since LPA, LPAp and LPAm detect different partitions on the same network for each run, they are ignored. The maximum resulted NMI values on each network has been bold in Table III.

TABLE II. REAL-WORLD NETWORKS WITH KNOWN COMMUNITY STRUCTURES

Network	Nodes	Links	Communities
Karate [22]	34	78	2
Dolphin [23]	62	159	2
Football [1]	115	615	12
Polblog [24]	1490	16715	2

1) Zachary Karate club: The well-known Karate club network of Zachary [22] is a standard benchmark for community detection. Zachary observed 34 members of a karate club in the United States over two years. Because of a disagreement between administrator and instructor of the club, a new club was formed by the instructor by taking about the half of the

original club members. The edge between nodes (members) of this network represent the social interactions between the members outside the club. These two original communities are specified with the shapes 'square' and 'circle' in Fig. 8.

As it can be observed from Table III, WILPAS is the only method that finds exactly the two real communities of Karate club network with NMI=1. CenLP is the second best method with NMI=0.60 with finding four communities. NIBLPA has poor performance on Karate network with NMI=0.21. The sets of sizes of detected communities by WILPAS, CenLP and NIBLPA are $\{16, 18\}$, $\{12, 5, 4, 13\}$ and $\{2, 3, 29\}$, respectively. Fig. 8 shows the two detected communities by WILPAS on Karate club network with different colors.

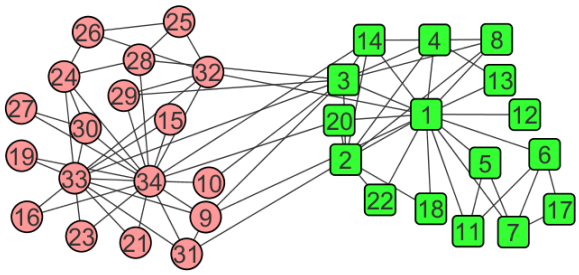

Fig. 8. Result of community detection by WILPAS on Karate network. Two real communities are specified with shapes 'circle' and 'square'. Two found communities are shaded with different colours. WILPAS method detects the two real communities exactly as it is.

2) American college football: Another well known benchmark for community detection is American college football network compiled by Girvan and Newman [1]. This network represents Division I games for the 2000 season. Nodes represent teams and the edges represent the games between teams. The teams belong to the conferences with 8 to 12 teams each. Since, games between the teams of the same conference are usually more frequent than the games between the teams of different communities, this network has community structure. As one can see from Table III both WILPAS and CenLP finds 13 communities on this network. In fact, both WILPAS and CenLP gain the maximum NMI value 0.90 on this network. After these two methods, LPAm is the third accurate method with NMI=0.89.

3) Dolphin social network: Dolphin network [23] shows the frequent associations between 62 dolphins living in Doubtful Sound, New Zealand. Nodes are dolphins and the edges between nodes shows that the two corresponding dolphin were seen together more than expected by chance. After leaving one of dolphins, they separated in two communities. Two original communities are specified with shapes 'circle' and 'square' in Fig. 9. Three communities which are detected by WILPAS are specified with different colors.

From Table III one can observe that WILPAS achieves higher NMI value than other methods on the Dolphin network. Moreover, the number of detected communities by WILPAS is more close to two real communities of Dolphin network. LPAm fails to detect true communities with getting lowest NMI value 0.45.

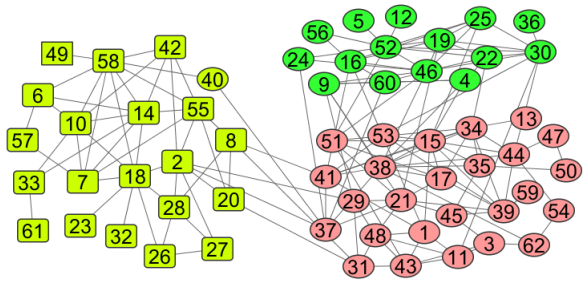

Fig. 9. Result of community detection by WILPAS on Dolphin network. Two real communities are specified with shapes 'circle' and 'square'. Three found communities are shaded with different colours.

4) Polblogs network: This network represents the links between weblogs about US politics preceding the US Presidential Election of 2004 [24]. The links were automatically extracted from a crawl of the front page of the weblogs. Each blog is labelled with '0' or '1' to indicate whether they are "liberal" or "conservative". This network can be considered both directed or undirected. In this paper, the undirected version of this network which has 1490 nodes and 16715 links is considered. Since nodes with degree zero makes this network disconnected, when comparing the performance of methods, these nodes are ignored. Thus, by removing 266 nodes with degree zero, the resulted network with 1224 nodes is considered for testing and comparing community detection methods. By doing this, the sizes of two real communities of Polblog are 588 and 636.

CenLP and WILPAS achieve NMI values 0.71 and 0.70 on Polblog network respectively. However CenLP gets a little more NMI value than WILPAS, but the number of detected communities of WILPAS is more close to two real communities of Polblog network. The sets of sizes of detected communities by WILPAS and CenLP are $\{552, 2, 670\}$ and $\{559, 2, 4, 659\}$, respectively.

In summary, when dealing with community detection on real networks, WILPAS outperforms other methods on Karate and Dolphin network, while CenLP has a little better accuracy than WILPAS on Polblog network. Both of these two methods has the same accuracy on Football network with finding 13 communities. The superiority of WILPAS on Karate and Dolphin networks is remarkable while superiority of CenLP on Polblog network is negligible. Moreover, while both of these two methods find 13 communities on Football network, the numbers of found communities of WILPAS on Karate, Dophin and Polblog networks are more close to the numbers of real communities of these networks. These show that the proposed method WILPAS is the most accurate label propagation method in comparison to other tested methods for community detection on the real networks.

TABLE III. NMI RESULTS OF THE METHODS ON FOUR REAL NETWORKS WITH KNOWN COMMUNITY STRUCTURES

Networks/ methods	LPAm	LPAp	WILPAS	LPA	NIBLPA	CenLP
Karate	0.55	0.56	**1**,{2}	0.70	0.21 {3}	0.60, {4}
Dolphin	0.45	0.55	**0.66**,{3}	0.52	0.50 {5}	0.61,{4}
Polblog	0.45	0.61	0.70,{3}	0.70	0.20 {9}	**0.71** ,{4}
Football	0.89	0.88	**0.90**,{13}	0.87	0.78 {9}	**0.90**{13}

C. Efficiency Analysis

To illustrate the running time of the proposed algorithm WILPAS and compare it with other algorithms, 10 networks using LFR software are produced, where the number of nodes $n = 100,000$ and the average degree $k = 40$ and $[minc, maxc] = [200, 1000]$. Fig. 10 plots the average running time of the proposed method WILPAS on these 10 synthetic networks compared with other five label propagation algorithms: LPA, LPAm, LPAp, CenLP and NIBLPA. As one can see from Fig. 10, method WILPAS is faster than LPAm but slower than LPA, LPAp and NIBLPA. In addition, it has a comparative execution time with CenLP.

Fig. 11 illustrates the running time of the weighting Algorithm 4 where n ranges from 100,000 to 500,000. As it can be seen from this figure, the weighting Algorithm 4 consumes less than 3.1 seconds for finding weights of this network with 100,000 nodes and around 2 million links. With increasing the number of node n to 500,000, the consumed time increase near linearly. Therefore, finding weights of all links of a network with 500,000 nodes and around 10 million links requires less than 25 seconds.

Similarly, for evaluating the scalability of WILPAS, the average running time of WILPAS on 10 LFR networks is reported where n ranges from 100,000 to 500,000. From Fig. 12 one can observe that the execution time of WILPAS scales approximately linearly with n, while it is less than double of execution time of LPA. As one can see from Fig. 12, WILPAS consumes less than 104 seconds for community detection on the network with $500,000$ nodes and around 10 million links. This shows the efficiency and scalability of WILPAS in community detection.

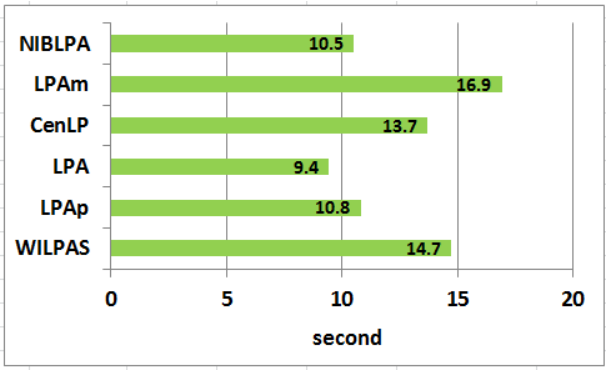

Fig. 10. The execution times of different methods on a network with n=100,000, k=40, $[minc, maxc] = [200, 1000]$.

VII. Conclusion

In this paper, a new label propagation algorithm called WILPAS is proposed. WILPAS presents specific update order and a novel label choosing formula in order to increase the accuracy of community detection. WILPAS is parameter-free that requires no prior knowledge. Experimental results on both synthetic and real-world tested networks demonstrate that WILPAS is the most accurate label propagation algorithm, while it is pretty fast. Moreover, finding communities of networks with around 10 million links in less than two minutes

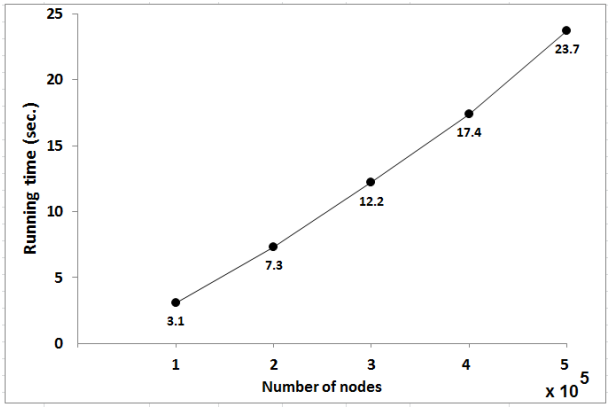

Fig. 11. The execution time of simple weighting Algorithm 4 with increasing n. The average degree k=40, $[minc, maxc] = [200, 1000]$. The number of nodes n ranges from 100,000 to 500,000.

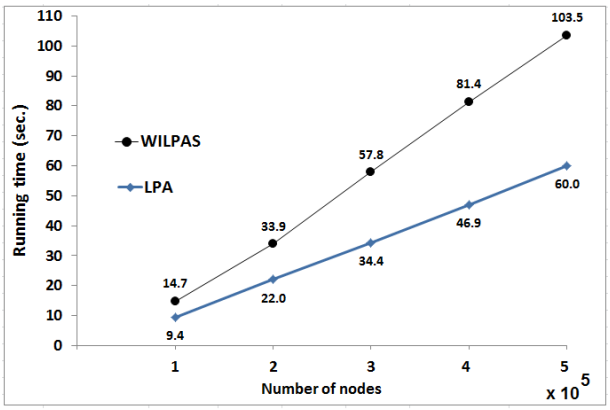

Fig. 12. The execution time in second for LPA and WILPAS on LFR benchmark with k=40, $[minc, maxc] = [200, 1000]$. The number of nodes n ranges from 100,000 to 500,000.

shows its scalability. Finally, experiments on several well-known real-world networks demonstrate that WILPAS outperforms other tested label propagation algorithms in finding true community structures of networks. In this paper, the communities should be distinct from each other. As future work, this algorithm can be extended to be used for overlapping (or fuzzy) community detection where each node may belong to several different communities.

References

[1] M. Girvan, M. E. Newman, Community structure in social and biological networks, Proceedings of the national academy of sciences 99 (12) (2002) 7821–7826.

[2] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical review E 69 (2) (2004) 026113.

[3] G. Agarwal, D. Kempe, Modularity-maximizing graph communities via mathematical programming, The European Physical Journal B 66 (3) (2008) 409–418.

[4] L. Bennett, S. Liu, L. Papageorgiou, S. Tsoka, A mathematical programming approach to community structure detection in complex networks, in: Symposium on Computer, no. June, 2012, pp. 17–20.

[5] M. Arab, M. Afsharchi, Community detection in social networks using hybrid merging of sub-communities, Journal of Network and Computer Applications 40 (2014) 73–84.

[6] B. W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell system technical journal 49 (2) (1970) 291–307.

[7] G. W. Flake, S. Lawrence, C. L. Giles, Efficient identification of web communities, in: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2000, pp. 150–160.

[8] S. White, P. Smyth, A spectral clustering approach to finding communities in graph., in: SDM, Vol. 5, SIAM, 2005, pp. 76–84.

[9] X. Xu, N. Yuruk, Z. Feng, T. A. Schweiger, Scan: a structural clustering algorithm for networks, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2007, pp. 824–833.

[10] H. Sun, J. Huang, J. Han, H. Deng, P. Zhao, B. Feng, gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 481–490.

[11] U. N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical review E 76 (3) (2007) 036106.

[12] I. X. Leung, P. Hui, P. Lio, J. Crowcroft, Towards real-time community detection in large networks, Physical Review E 79 (6) (2009) 066107.

[13] ping Zhang, A., Ren, G., Cao, H., zhu Jia, B. and bin Zhang, S., 2013, May. Generalization of label propagation algorithm in complex networks. In Control and Decision Conference (CCDC), 2013 25th Chinese (pp. 1306-1309). IEEE.

[14] Barber, M.J. and Clark, J.W., 2009. Detecting network communities by propagating labels under constraints. Physical Review E, 80(2), p.026129.

[15] Liu, X. and Murata, T., 2010. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. Physica A: Statistical Mechanics and its Applications, 389(7), pp.1493-1500.

[16] Zhang, A., Ren, G., Lin, Y., Jia, B., Cao, H., Zhang, J. and Zhang, S., 2014. Detecting community structures in networks by label propagation with prediction of percolation transition. The Scientific World Journal, 2014.

[17] Xing, Y., Meng, F., Zhou, Y., Zhu, M., Shi, M. and Sun, G., 2014. A node influence based label propagation algorithm for community detection in networks. The Scientific World Journal, 2014.

[18] Sun, H., Liu, J., Huang, J., Wang, G., Yang, Z., Song, Q. and Jia, X., 2015. CenLP: A centrality-based label propagation algorithm for community detection in networks. Physica A: Statistical Mechanics and its Applications, 436, pp.767-780.

[19] Z. Liu, P. Li, Y. Zheng, M. Sun, Community detection by affinity propagation, Tech. rep., Technical Report (2008).

[20] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment 2005 (09) (2005) P09008.

[21] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical review E 78 (4) (2008) 046110.

[22] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of anthropological research (1977) 452–473.

[23] D. Lusseau, M. E. Newman, Identifying the role that animals play in their social networks, Proceedings of the Royal Society of London B: Biological Sciences 271 (Suppl 6) (2004) S477–S481.

[24] Adamic, L.A. and Glance, N., 2005, August. The political blogosphere and the 2004 US election: divided they blog. In Proceedings of the 3rd international workshop on Link discovery (pp. 36-43). ACM.