Hindawi
Complexity
Volume 2019, Article ID 6946189, 16 pages
https://doi.org/10.1155/2019/6946189

Research Article

Community Detection with Self-Adapting Switching Based on Affinity

Ning-Ning Wang[1,2,3] **Zhen Jin**[1,2] **and Xiao-Long Peng**[1,2]

[1]*Complex Systems Research Center, Shanxi University, Taiyuan 030006, China*
[2]*Shanxi Key Laboratory of Mathematical Techniques and Big Data Analysis on Disease Control and Prevention, Shanxi University, Taiyuan 030006, China*
[3]*School of Systems Science, Beijing Normal University, Beijing 100875, China*

Correspondence should be addressed to Zhen Jin; jinzhn@263.net

Received 14 April 2019; Revised 20 August 2019; Accepted 23 September 2019; Published 13 November 2019

Academic Editor: Sergio Gómez

Community structures in complex networks play an important role in researching network function. Although there are various algorithms based on affinity or similarity, their drawbacks are obvious. They perform well in strong communities, but perform poor in weak communities. Experiments show that sometimes, community detection algorithms based on a single affinity do not work well, especially for weak communities. So we design a self-adapting switching (SAS) algorithm, where weak communities are detected by combination of two affinities. Compared with some state-of-the-art algorithms, the algorithm has a competitive accuracy and its time complexity is near linear. Our algorithm also provides a new framework of combination algorithm for community detection. Some extensive computational simulations on both artificial and real-world networks confirm the potential capability of our algorithm.

1. Introduction

The continuing advance of network science plays a prominent role in deepening the understanding of complex systems in the real world [1–3]. Among others, one salient property commonly observed in many complex networks is the community structure, i.e., the organization of nodes in different groups, with many edges connecting nodes of the same group and comparatively fewer connections among nodes of different groups [4–7]. For instance, in a scientific citation network, communities are sets of scientific papers on the same topic or in a similar research field [8], while in protein-protein interaction networks, proteins working in the same biological process (or being in the same cellular component) interact with each other. Moreover, the community structure has been shown to have strong impacts on epidemic dynamics [9, 10] and link prediction. Therefore, with the acquisition of the real network data, one should pay careful attention to the community structure, which is of value to further investigations of complex networks.

For a deep understanding to the community structure, it is necessary to define what a community is. In general, there are three types of definitions: local definition, global definition, and definition based on vertex similarity [6], including the definition based on modularity and the topological structure, such as the self-referring definition and comparative definition [11]. However, there are few definitions that quantitatively describe the community structure. In 2003, Radicchi et al. provide the community definitions in both the strong and weak sense with the quantitative description [12]: the subgraph C is a community in a strong sense if

$$k_i^{\text{in}}(C) > k_i^{\text{out}}(C), \quad \forall i \in C, \tag{1}$$

and in a weak sense if

$$\sum_{i \in C} k_i^{\text{in}}(C) > \sum_{i \in C} k_i^{\text{out}}(C), \quad \forall i \in C. \tag{2}$$

The above quantitative definitions mean that the degrees inside of all, or most, nodes are more than the degrees

outside, where the degree inside $k_i^{\text{in}}(C)$ is the number of node's neighbors in the same community and the degree outside $k_i^{\text{out}}(C)$ is the number of node's neighbors in other communities. Thereafter, another quantitative definition is defined by Hu et al. [11] as follows: subnetworks (or subgraphs) $C_1, C_2, \ldots, C_m$ are said to be m communities of a network (or graph) G if and only if they satisfy that $\cup_{l=1}^{l=m} C_l = G$, and for any node $j \in C_{l_0}$, $l_0 \in \{1, 2, \ldots, m\}$, one has

$$\sum_{i \in C_{l_0}} A_{i,j} \geq \max\left\{ \sum_{i \in C_t} A_{i,j},\ t = 1, 2, \ldots, m \right\}, \tag{3}$$

where A is the adjacency matrix of the graph G. Unlike the consideration by Zhan et al. [13], we regard this definition as the generalized definition, since it allows that each node degree outside can be more than degree inside, and only need the node which has the largest number of neighbors with its own community. In this paper, we use this definition as our standard for community detection and it is remarkable that the overlap of node is not considered and node belongs to only one community based on the detection result.

In order to accurately describe the quantitative relation between the degree inside and outside of communities, Lancichinetti et al. introduce a mixing parameter μ_i for each node i to denote that the node i shares a fraction μ_i of its links with external nodes and a fraction $1 - \mu_i$ with internal nodes, i.e., $\mu_i = k_i^{\text{out}}/(k_i^{\text{in}} + k_i^{\text{out}})$ [14, 15]. In this paper, we consider that the mixing parameter of each node is less than 0.5 in *strong* communities, and contrarily, it is more than 0.5 in *weak* communities, and these two kinds of communities all satisfy the definition of Hu et al.

There have been various kinds of algorithms designed for community detection. For example, the Kernighan–Lin algorithm, spectral bisection method, k-means clustering method, and the spectral clustering algorithm are traditional algorithms derived from graph theory or statistics. With the development of computers, large-scale computing is becoming widely available, so it is feasible to increase the calculation complexity and network scale. These advances enable researchers to develop many optimized algorithms, including the greedy algorithms based on modularity [16] and betweenness [4, 17]. Meanwhile, there are some algorithms which are based on dynamical methods [18–23] and similarity or affinity [24, 25]. However, ignoring difference between the *strong* and *weak* communities is a major drawback to some algorithms based on node affinity or similarity, which makes the detection accuracy of these algorithms low for *weak* communities. Thus, we design a self-adapting switching (SAS) algorithm based on single affinity and combination of two affinities.

The evaluation criterions for the performance of community detection can be determined by two kinds of approaches. One is to compute the topology-based metrics, including the coverage, conductance, and modularity metrics. The other is to calculate the knowledge-driven measurements, such as the Precision metrics, Jaccard index, and the normalized mutual information (NMI) [26]. We adopt NMI index as the evaluation criterion for the performance of algorithms in some real-world networks, the Lancichinetti–Fortunato–Radicchi (LFR) benchmark networks (heterogeneous networks) [14], the Girvan–Newman (GN) benchmark networks (homogeneous networks) [4], and the nonuniform popularity similarity optimization (nPSO) benchmark networks (heterogeneous networks) [27]. Based on the results, we find that our algorithm has an advantage over some state-of-the-art algorithms and is more suitable for heterogeneous networks with larger power-law exponent. This paper is outlined as follows. In Section 2, we design the principle of our algorithm and discuss its complexity. Tests and results are presented in Section 3. Conclusions are summarized in Section 4.

2. Structural Analysis and Algorithm

In this section, we will present an analysis about the community structure and design the affinity-based SAS algorithm for community detection, and then its complexity is discussed at last.

2.1. The Analysis of Community Structure. Some studies indicate that the node degrees generally obey the power-law distribution [28, 29] or [30] log-normal distributions in real-world networks, where the nodes with large degree are known as *hub* nodes and have strong degree centrality, such as the network in Figure 1. Although the number of *hub* nodes in real-world networks is relatively small, their vital roles in communities and networks have been repeatedly mentioned in some literature studies [13, 31, 32]. The identification of the *hub* nodes is usually considered as the breakthrough point for heuristic algorithms. In these algorithms, a single affinity is often deficient for community detection, especially for the *weak* communities. Therefore, we design a new algorithm that combines two affinities in the detection of *weak* communities.

As is well known, the ultimate aim of the community detection algorithms that are based on affinity or modularity is to find the global maximum of such indices and to guarantee the minimum number of connections between different communities. Both of them are nondeterministic polynomial hard problems. Putting aside these problems, our algorithm is heuristic and its detection process is based on the affinity between the nodes being detected and having been detected, rather than between two single nodes. Motivated by the different affinities, i.e., the common neighbors (CN), *hub* depressed (HD), and *hub* promoted (HP) indices summarized by Zhou et al. [33], we provide two definitions of affinity for node j and node set P as follows, and some important notations are shown in Table 1.

The first affinity $s_P^{(j)}$ between any node j and node set P is as follows:

$$s_P^{(j)} = \left| N_j \cap P \right|. \tag{4}$$

The second affinity $S_P^{(j)}$ between any node j and node set P is given by

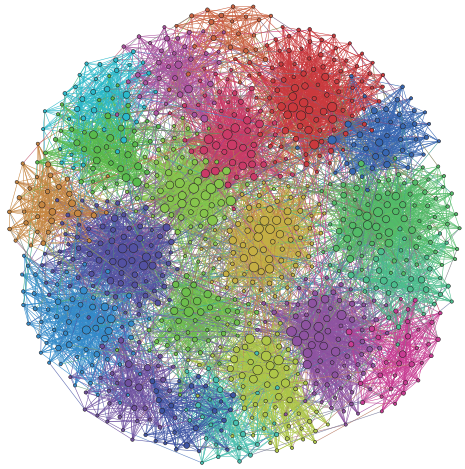


Figure 1: In heterogeneous networks, *hub* nodes are scattered in almost every communities. These *hub* nodes are the key starting point in community detection. Communities are distinguished by different colors, where node degree is marked by different sizes.

Table 1: Description of main symbols.

Notation	Meaning
$\|A\|$	The number of elements in set A
$C_m(t)$	The set of nodes detected at t^{th} step that belong to the m^{th} community, where t is the detection step
C_m^*	The set of nodes that belong to the m^{th} community when $C_m(t)$ no longer changes
$C_m^{\text{new}}(t)$	The new detected nodes that belong to the m^{th} community at step t, $C_m^{\text{new}}(t) = C_m(t) - C_m(t-1)$
N_i	The set of neighbors of node i
$s_P^{(j)}$	The *first affinity* between node j and P, where P is a set of nodes
$S_P^{(j)}$	The *second affinity* between node j and P

$$S_P^{(j)} = \frac{s_P^{(j)}}{k_j}, \tag{5}$$

where k_j is the degree of node j.

These two affinities have different emphases: the first one focuses on the absolute number of the common neighbors and the second is the relative affinity. Our heuristic algorithm is implemented from the *hub* node and then detects other nodes belonging to the same community based on these affinities.

Generally, the affinities between nodes in one community are larger than those between nodes in different communities, while this is hard to be satisfied sometimes, especially for the *weak* communities. To illustrate this point, we calculate the first and second affinity between the *hub* node and its neighbors in LFR benchmark graphs. First, the second affinity between the *hub* node's neighbors in the same community and other communities is shown in Figure 2(a). We discover that the second affinities of nodes in *strong* communities have obvious differences, but they are mixed together in the *weak* communities when $\mu > 0.5$. We conduct a similar experiment on the *weak* communities with the first affinity to observe its distinction ability. Since the first affinity is the absolute number of common neighbors, we normalize it and only pay attention to its normalization form $\gamma_{N_i}^{(j)}$ in Figure 2(b), where the notation of *hub* node is i, the notation of its neighbors set is N_i, and node j is a neighbor of *hub* node:

$$\gamma_{N_i}^{(j)} = \frac{s_{N_i}^{(j)}}{\max_{j \in N_i}\left\{s_{N_i}^{(j)}\right\}}, \quad j \in N_i. \tag{6}$$

From the statistical results, we find that, for the *strong* communities, the second affinity has effective distinction ability. However, it is not enough to detect the *weak* communities and need to work with the first affinity. Moreover, the detection method of *strong* communities is not suitable to *weak* communities and may detect many communities composed of several nodes or even a single node, which can be a trigger principle of the switch condition in our SAS algorithm. So our algorithm is divided into two parts, which we name in short as SAS-1 and SAS-2, respectively. Next, we will describe the algorithm and its principle in detail.

2.2. The Algorithm. Here, we will introduce the two parts of our algorithm including its core principles and pseudocodes and then analyze its complexity. Some important notations are also shown in Table 1.

2.2.1. The Strong Community Method SAS-1. In this method, each community, its nodes and the edges of these nodes, will be gradually deleted from the network after the end of its detection. So we denote the network as $G_m = (V_m, E_m)$ after the $(m-1)^{\text{th}}$ $(m > 1)$ community has been detected, where V_m and E_m are the sets of nodes and edges, respectively. In order to describe the algorithm generally, we will use the example of the detection of m^{th} community.

The first step: at step $t = 1$, the method selects one node $i \in V_m$ as the *hub* node, whose degree is maximal in G_m. At this step, the *hub* node i and its neighbors, satisfying $S_P^{(j)} \geq 0.5$, are the detected nodes belonging to $C_m(1)$, where the node set P consists of the node i and its neighbors, node $j \in N_i$.

The second step: at step $t = 2$, the method searches the nodes in $C_m(1)$, and then these nodes' neighbor j is substituted into this community if and only if it satisfies the condition

$$S_{C_m(1)}^{(j)} \geq 0.5, \tag{7}$$

where the value 0.5 is confirmed by the definition of the *strong* community, and then the community from $C_m(1)$ to $C_m(2)$ is updated.

The t^{th} step: similarly, when $t \geq 3$, in order to reduce the complexity, the method searches the nodes in $C_m^{\text{new}}(t)$ and only detects these nodes' undetected neighbors. Then, neighbor j is substituted into $C_m(t)$ if and only if it satisfies the condition

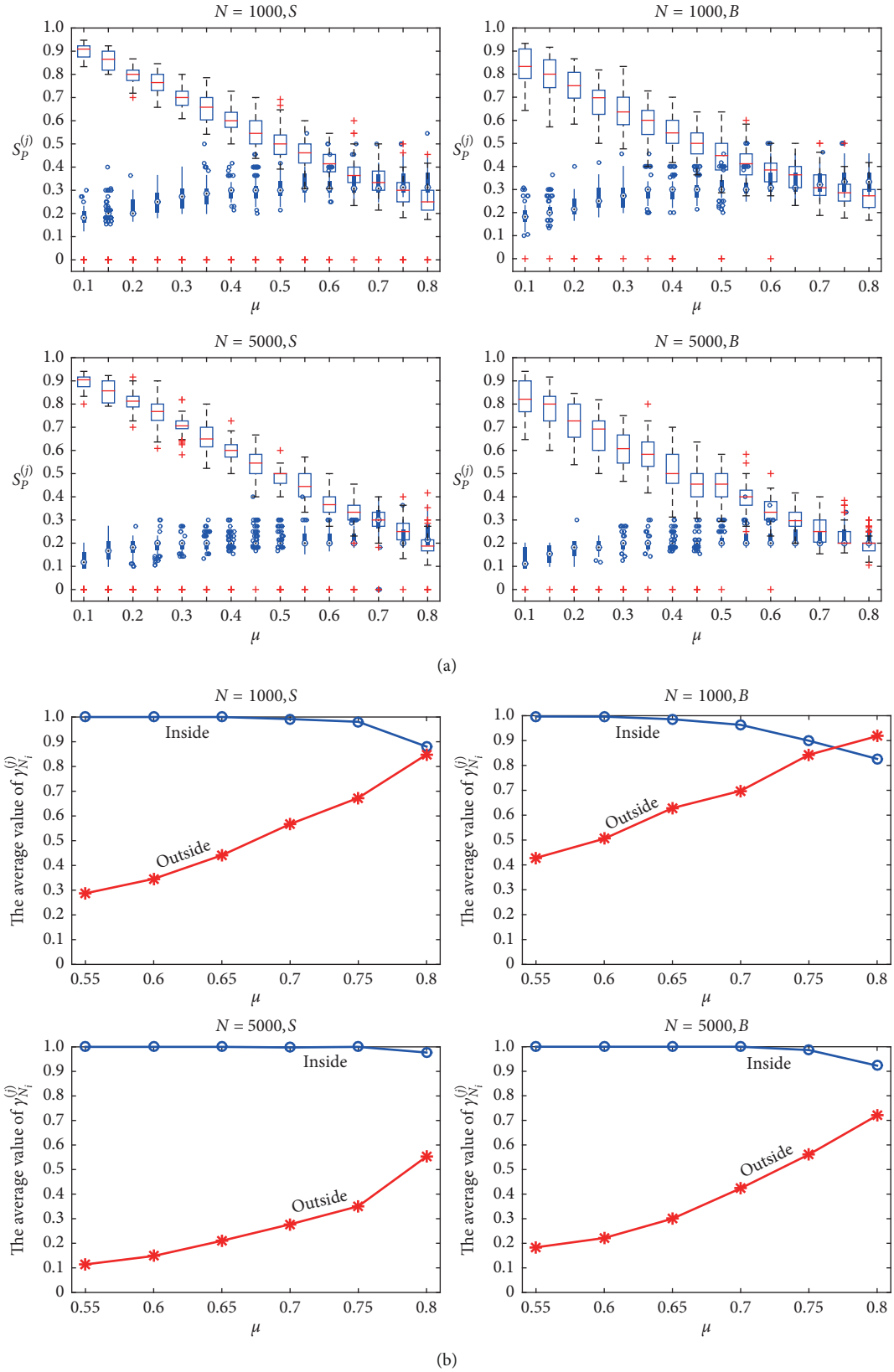


FIGURE 2: (a) We present the statistical results of $S_P^{(j)}$ for different values of μ from 0.1 to 0.8. The detailed information about the size and scale of network structure is shown in Section 3. The hollow box plot is calculated from the *hub* nodes' neighbors in the same community, and the solid blue plot is calculated from the *hub* nodes' neighbors in other communities. Each test is conducted by 100 times. (b) We statistically compute the normalization form $\gamma_{N_i}^{(j)}$ for the neighbors inside and those outside the communities, respectively. Then, we get the mean values of $\gamma_{N_i}^{(j)}$ for each *hub* node i, where the blue lines are derived from the inside neighbors of the *hub* node and the red lines are from outside. In both (a) and (b), the letter S denotes communities of size between 10 and 50 while the letter B denotes communities of size between 20 and 100.

Algorithm community detection with self-adapting switching.
(1) **input** Adjacency matrix of the network.
(2) **while** $|\cup_{m=1}^{n_c} C_m^*|/n_c \le \beta$.
(3) Select a node as the *hub* node such that its degree is maximal in the current network.
(4) Gradually search the suitable nodes by $S_{C_m(t-1)}^{(j)} \ge 0.5$.
(5) Repeat step 4 until no nodes satisfy the condition in step 4, and then update the network and start the next detection.
(6) **if** $|\cup_{m=1}^{n_c} C_m^*|/n_c \le \beta$ fails.
(7) Select a node as the hub node such that its degree is maximal in the network.
(8) Gradually search the suitable nodes by the double affinities condition: $S_{P_2}^{(j)} = \max\{S_{P_2}^{(j)}\}$, $s_{P_2}^{(j)} = \max\{s_{P_2}^{(j)}\}$.
(9) Repeat step 8 until no nodes satisfy $S_{P_2}^{(j)} \le \rho \cdot \max\{S_{P_2}^{(j')}\}$, then start the next detection.
(10) Adjust the results based on the community definition.

Algorithm 1: The pseudocodes of the SAS algorithm.

$$S_{C_m(t-1)}^{(j)} \ge 0.5, \tag{8}$$

and the community $C_m(t)$ is updated.

The detection process of the m^{th} community finishes until there are no nodes satisfying condition (8).

2.2.2. The Weak Community Method SAS-2. From the results in Figure 2, we can infer the method SAS-1 may detect many communities that are composed of several nodes or even a single node in *weak* communities. Hence, the algorithm needs a self-adapting switching condition to reflect this phenomenon and make it to switch from SAS-1 to SAS-2. Our method is to calculate the average scale of communities having been detected and the *switching condition* between the two methods is given by

$$\frac{\left|\cup_{m=1}^{n_c} C_m^*\right|}{n_c} \le \beta, \tag{9}$$

where $\beta = O(p\langle k\rangle)$, in which $p = 0.05$ and $\langle k\rangle$ is the average degree, and $n_c(\ge 1)$ is the current number of communities having been detected. Actually, few neighbors of *hub* node in *weak* community can satisfy condition (7), so the average scale of communities detected by SAS-1 is the same order with $p\langle k\rangle$, and the parameter p derived from hypothesis test is a small incidence rate.

Once the SAS-1 triggers the switching condition, it will switch to SAS-2 and redetects the network. Different from the first method SAS-1, this method does not delete any nodes or edges from the network because the recognition of the *weak* communities depends on the whole construction of the network. In the following, we will also introduce this new method by taking the detection progress of the m^{th} community as an example.

The first step: at step $t = 1$, the method selects the node i with the maximal degree as the starting node, which does not belong to other communities $C_1^*, \ldots, C_{m-1}^*$. Obviously, we have $C_m(1) = \{i\}$ after confirming the starting node.

The second step: at step $t = 2$, the method chooses the *hub* node's neighbor j not belonging to other communities $C_1^*, \ldots, C_{m-1}^*$, as the member of the m^{th} community if and only if it satisfies the following condition:

$$s_{N_i}^{(j)} \ge \gamma \cdot \max_{j \in N_i}\{s_{N_i}^{(j)}\}, \tag{10}$$

where γ is a threshold based on the average value of $\gamma_{N_i}^{(j)}$ in Figure 2, and then $C_m(2)$ is updated.

The t^{th} step: when $t \ge 3$, similar to the method SAS-1, this method searches the nodes in $C_m^{\text{new}}(t)$ and only detects these nodes' undetected neighbors. Then neighbor j is substituted into $C_m(t)$ if and only if it satisfies the condition

$$\begin{aligned} S_{C_m(t-1)}^{(j)} &= \max\left\{S_{C_m(t-1)}^{(j)}\right\}, \\ s_{C_m(t-1)}^{(j)} &= \max\left\{s_{C_m(t-1)}^{(j)}\right\}. \end{aligned} \tag{11}$$

The termination condition of the m^{th} community is to separate the undetected nodes with lower affinity from the nodes having been detected, which have higher affinity each other. We assume that the detection of the m^{th} community stops when there is no node j satisfying the following condition at step $t = t_0$:

$$S_{C_m(t_0-1)}^{(j)} \ge \rho \cdot \max\left\{S_{C_m(t_0-1)}^{(j')}\right\}, \tag{12}$$

where node j is one of the undetected neighbors of nodes, which belong to $C_m^{\text{new}}(t_0 - 1)$, and node j' belong to $C_m(t_0 - 1)$, and the parameter $\rho \in (0, 1)$ is used to cut the community in the network.

The algorithm pseudocodes are shown in Algorithm 1 and its process structure is shown in Figure 3. Last, we analyze the algorithm complexity. In the method SAS-1, the detection process is conducted in every communities, so we consider that the average step number for each community is t_a. The complexity in searching and filtering for each node by the condition (8) scale is $O(\langle k\rangle^2)$. With the detection of communities, the number of nodes is reduced, so the extreme complexity is about $O(\langle k\rangle^2 t_a n_c)$, where n_c is the number of communities having been detected and $\langle k\rangle$ is the average degree. In the method SAS-

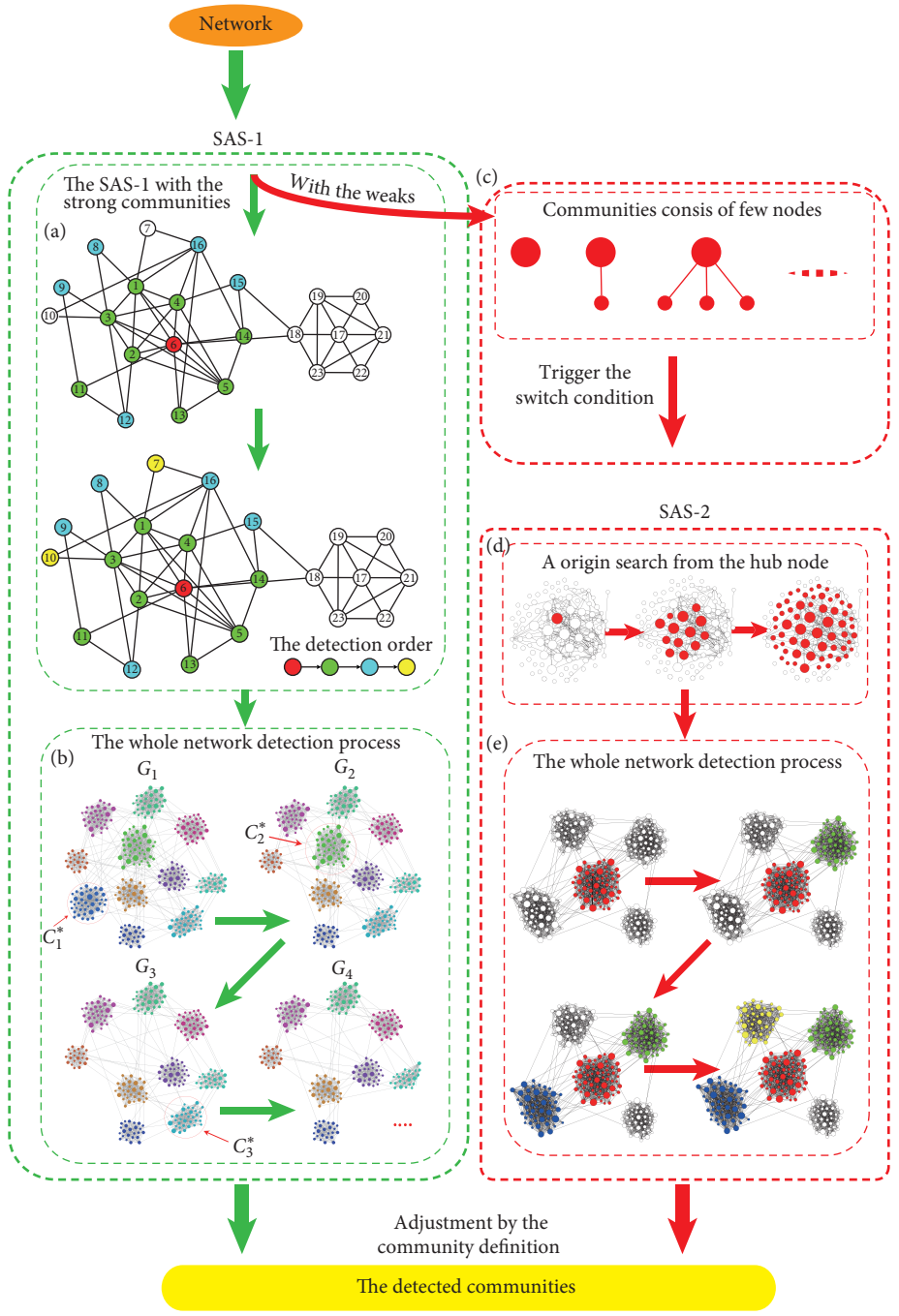
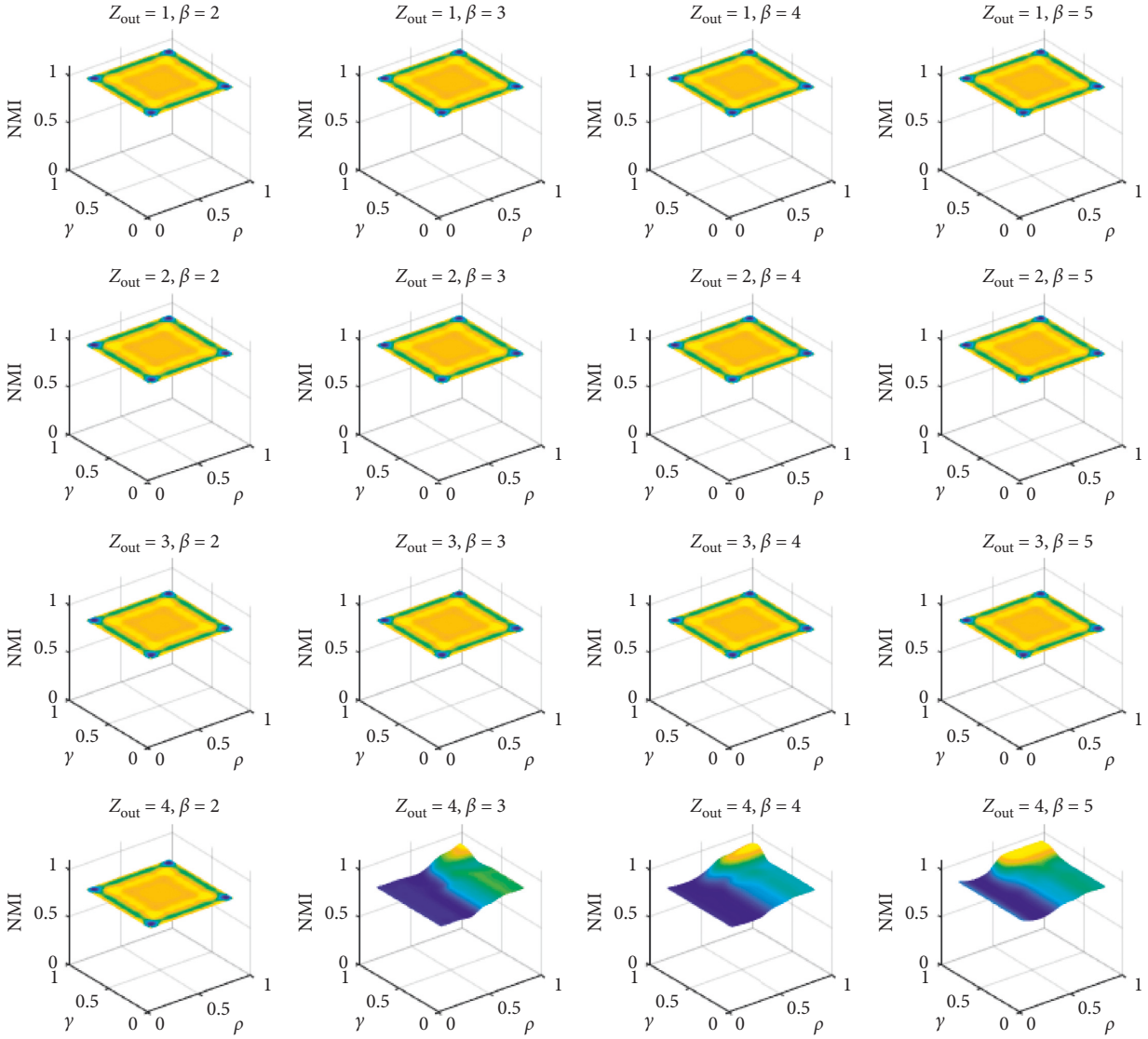


FIGURE 3: Flowchart of the algorithm with self-adapting switching (i.e., the SAS algorithm) based on affinity. The green area in the left is part of the algorithm SAS-1. (a) The SAS-1 starts from hub nodes and gradually searches their nodes in the same community. (b) After the end of one community detection, the nodes and edges of this community will be deleted in the network, and new network data will be provided for the following detection. The red area in the right is part of the algorithm SAS-2. (c-d) Once the algorithm satisfies the switch condition, the SAS-2 will use the combination of two affinities to detect communities one by one. (e) Different from SAS-1, SAS-2 does not delete communities having been detected.

2, the detection process is different only in the detected condition (11), so the complexity is also about $O(\langle k\rangle^2 t_a n_c)$. In summary, the complexity of the SAS algorithm is $O(\langle k\rangle^2 t_a n_c)$. Evidently, t_a is the same order as the average size of community N_c, namely, $O(t_a/N_c) = 1$. Since $O(n_c) = O(n/N_c)$, thus the complexity can be derived to $O(\langle k\rangle^2 n) = O(\langle k\rangle m)$, where m is the number of network edges.

3. Results

In this section, some experiments are performed on both real-world networks (the karate club network, the dolphin network, the football team network, and the political books network) and synthetic networks (LFR, GN, and nPSO benchmark graphs). First, we use the GN benchmark to



(a)

Figure 4: Continued.

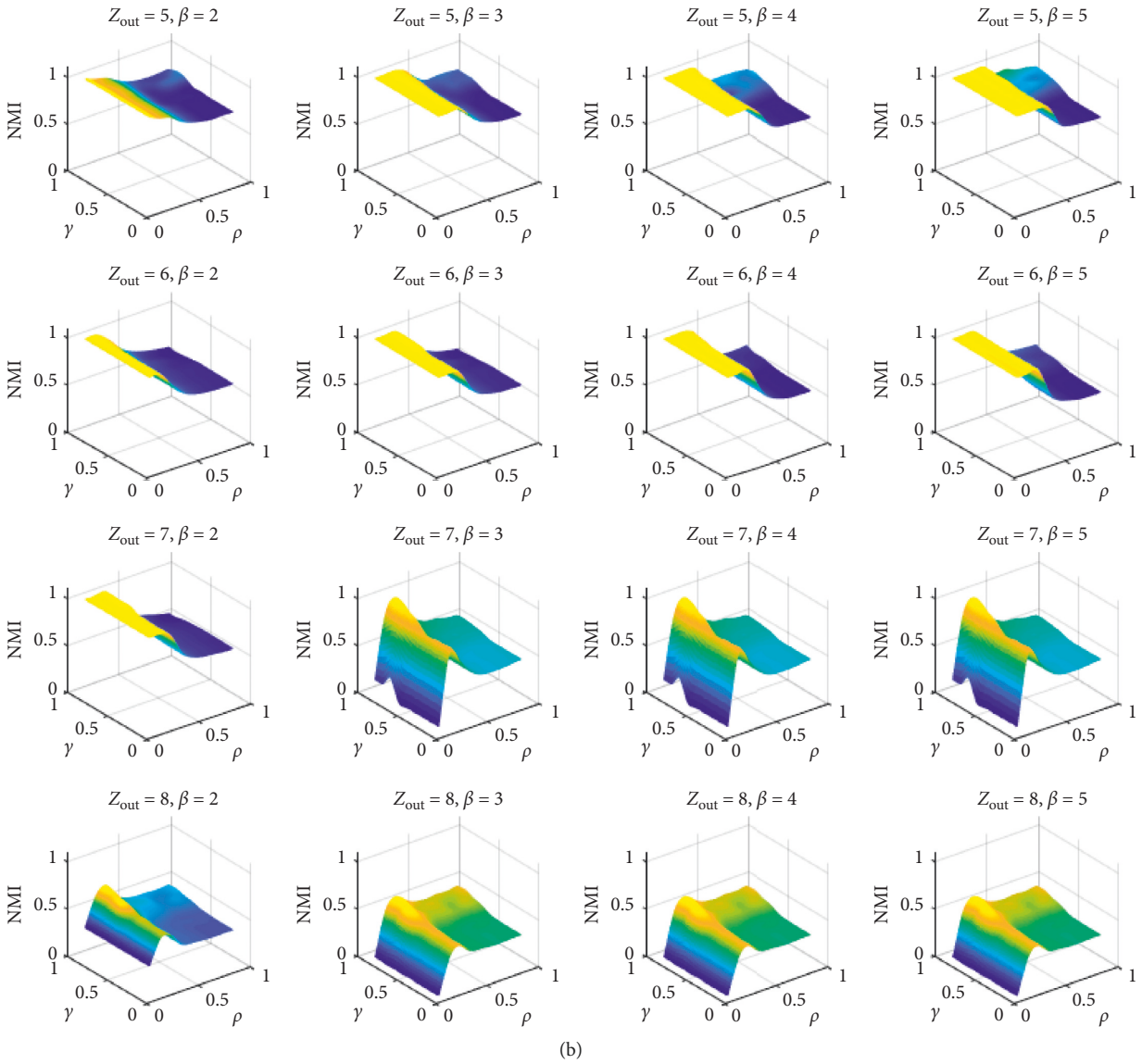


Figure 4: The sensitivity analysis of SAS algorithm detection results with parameters β, γ, and ρ in GN benchmark. For each combination of parameters, 100 networks have been generated.

estimate suitable parameters range and analyze parameter sensitivity. The SAS algorithm relies on three parameters β, γ, and ρ, where the choice of parameter β is related to the average degree $\langle k \rangle$. The parameters ρ and γ can be freely selected in the range $(0, 1)$. In the GN benchmark, its scale is 128 and degree distribution is relatively concentrated, so it is suitable for parameter sensitivity analysis. Because its average degree $\langle k \rangle$ is 16, so we default the $\beta = 2, 3, 4, 5$ and mainly study the sensitivity of parameters ρ and γ. Based on the results in Figure 4, we find that the results are insensitive to parameters β and γ. However, the changes of parameter ρ have obvious influence on the results when $Z_{out} > 4$. Fortunately, when the parameter $0.3 < \rho < 1$, all the detected results are stable and do not have wide-range fluctuations.

In practice, the parameter γ should be close to 1 to ensure the accuracy of initial detected nodes. The parameter $0.3 < \rho < 1$ should be appropriately increased with the increase of clustering coefficient. Then, we evaluate the advantages and disadvantages of our algorithm compared with other state-of-the-art algorithms: Infomap, LPA, Louvain, Walktrap, Fast greedy, EM, and Blondel. The performance comparison in real-world networks confirms its potential capability shown in Table 2 and Figure 5. It is worth mentioning that some community divisions are slightly different from the ground truth. The possible reason is that the detailed division of communities leads to an increase in the number of community, but its results at least satisfy the quantitative definition of our article and have a good accuracy rate.

In order to discuss our algorithm accuracy deeply, we use three benchmark networks: LFR, GN, and nPSO to study how the algorithm performance, NMI [26], changes with the weakening of the community structure.

Table 2: Performance comparison in real-world networks, where n and m are the number of nodes and edges and <k> is the average degree of the network.

Real-world networks	Dolphin social network	American college football	Zachary's karate club	Political books	Complexity
SAS (NMI)	0.9071	0.9114	1	0.5861	$O(\langle k\rangle m)$
Infomap (NMI) [34]	0.3892	0.9241	0.6994	0.4934	$O(n(n+m))$
LPA (NMI) [35]	0.5267	0.9094	0.7403	0.5744	$O(n+m)$
Louvain (NMI) [36]	0.4483	0.8903	0.5866	0.5368	$O(n)$ or $O(m)$
Walktrap (NMI) [37]	0.4740	0.8873	0.5041	0.5427	$O(n^2\log(n))$
Fast greedy (NMI) [16]	0.4482	0.6977	0.6924	0.5308	$O(n\log^2 n)$
EM (NMI) [38]	0.4428	0.6986	0.6771	0.5201	$O(n^2)$
Blondel (NMI) [39]	0.4143	0.8903	0.5866	0.5121	$O(m)$

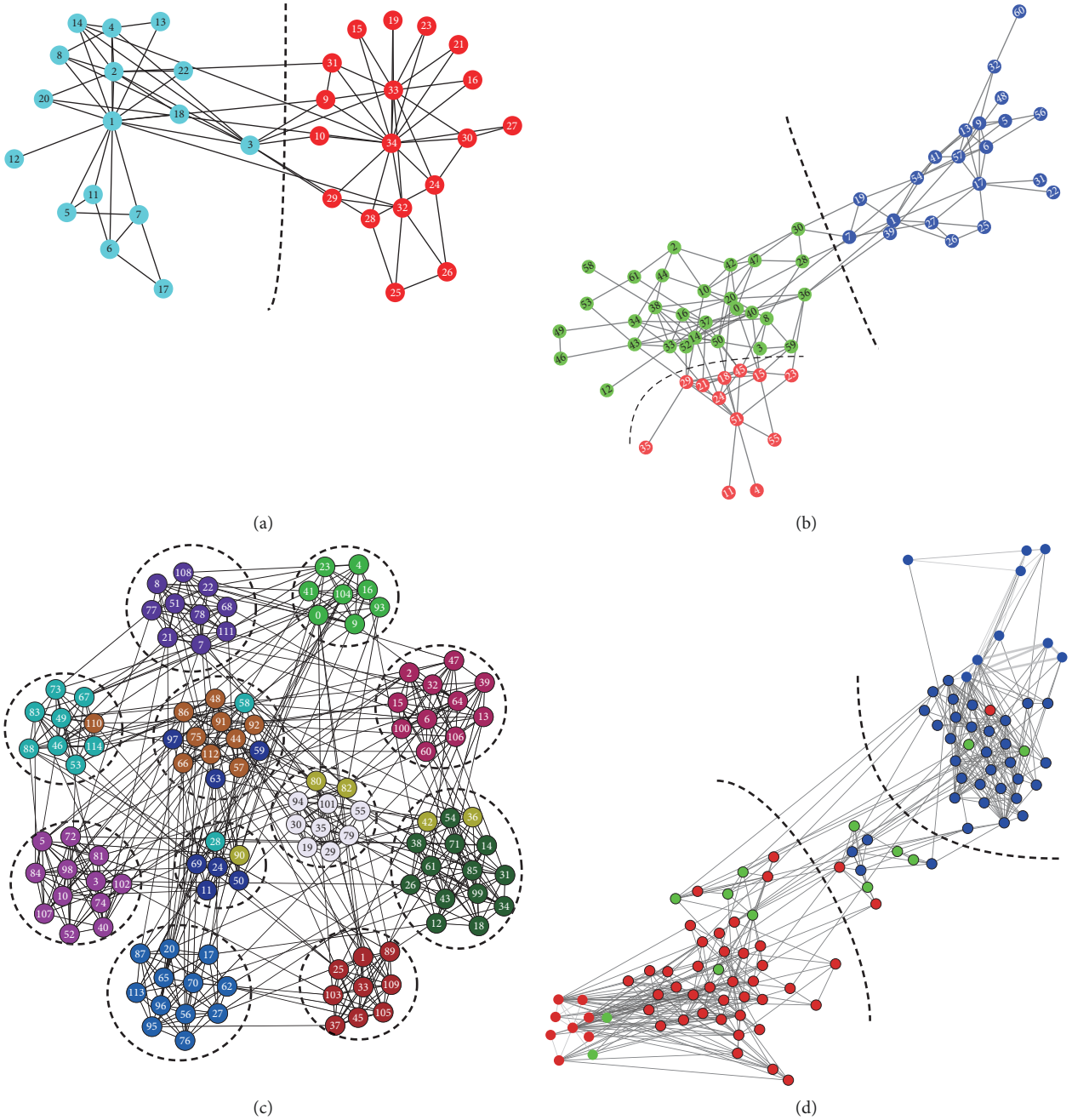


Figure 5: The detection results for various real-world networks. For the karate network (a), the result is identical with the original. But for the dolphin network (b), the left community in the original has been divided into two communities, which are marked in red and green, respectively. In the football team network (c) and the political books network (d), the detected communities are illustrated by dashed lines and the original partitions are distinguished by different colors. The parameters $\beta = 3$, $\gamma = 0.9$, and $\rho = 0.8$.

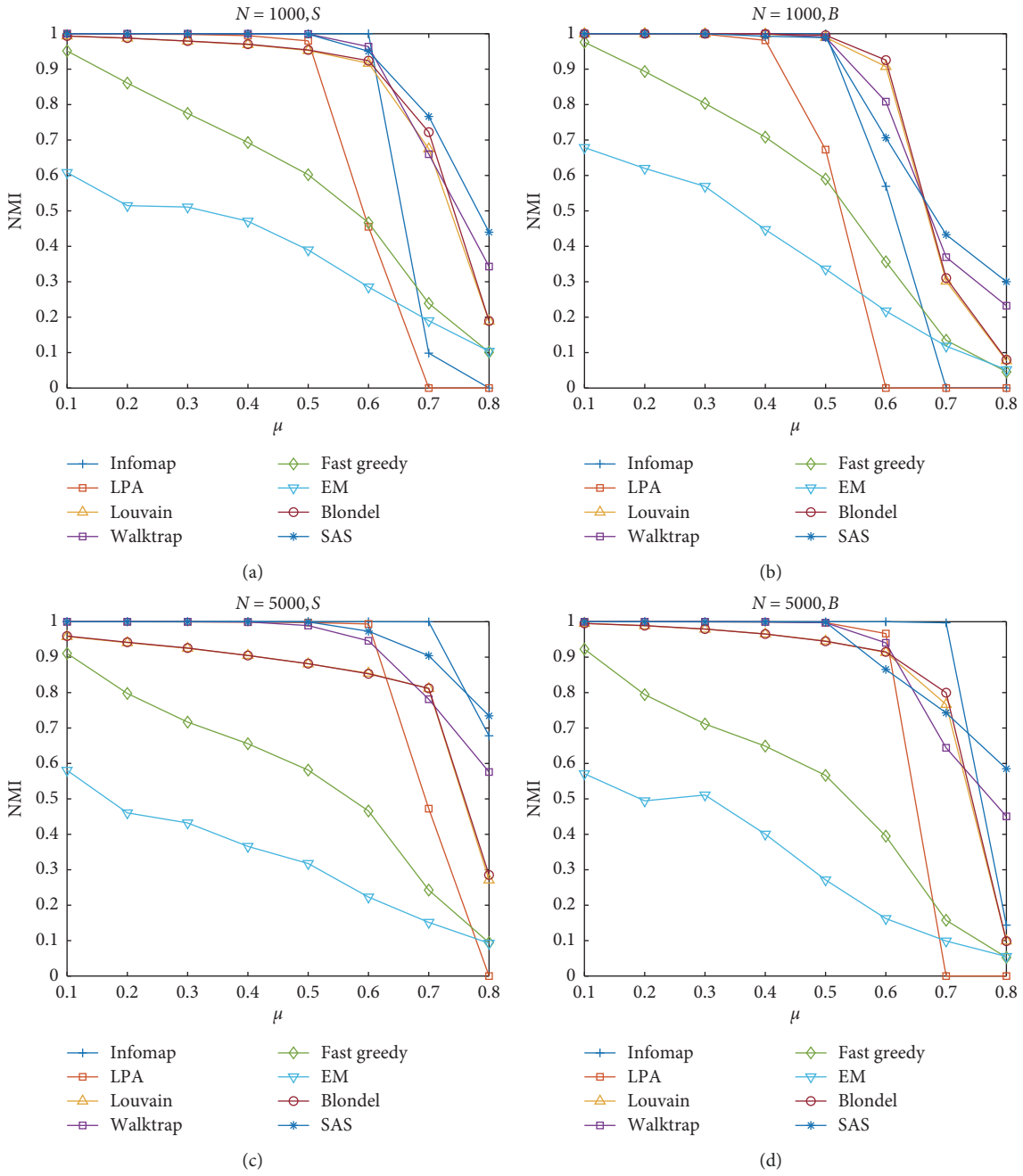


Figure 6: Community detection on LFR networks. For each combination of parameters, 100 networks have been generated. And for each network, the community detection methods Infomap, LPA, Louvain, Walktrap, Fast greedy, EM, and Blondel have been executed. The communities detected have been compared by computing the normalized mutual information (NMI). The parameters $\beta = 3$, $\gamma = 0.9$, and $\rho = 0.8$.

3.1. The LFR Benchmark. In this part, LFR networks have two different scales: 1000 and 5000, as presented in Figure 6. For each kind of network, we consider two different community sizes, indicated by the letters S and B, where S stands for "small" communities that have about 10 to 50 nodes and B stands for "big" communities that have about 20 to 100 nodes [15]. In Figure 6, our algorithm tests four types of networks by NMI with $\mu \in [0.1, 0.8]$. For the *strong* and *weak* community, the performance of our algorithm is better than some algorithms in Table 2.

3.2. The GN Benchmark. Beyond that, we test the SAS algorithm in the GN benchmark network with the results shown in Figure 7, where each point is also tested on 100 same kind networks. The performance of SAS algorithm is as good as other algorithms in Table 2. It is well known that the LFR benchmark is a kind of heterogeneous networks, whose degree distribution follows the power-law distribution. However, for the GN benchmark, its degree distribution follows the normal distribution and the role of *hub* nodes is weakened. Maybe the heterogeneity of network structure

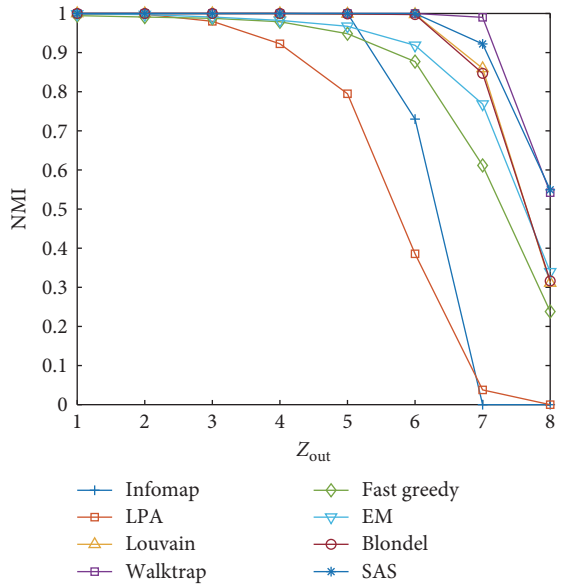
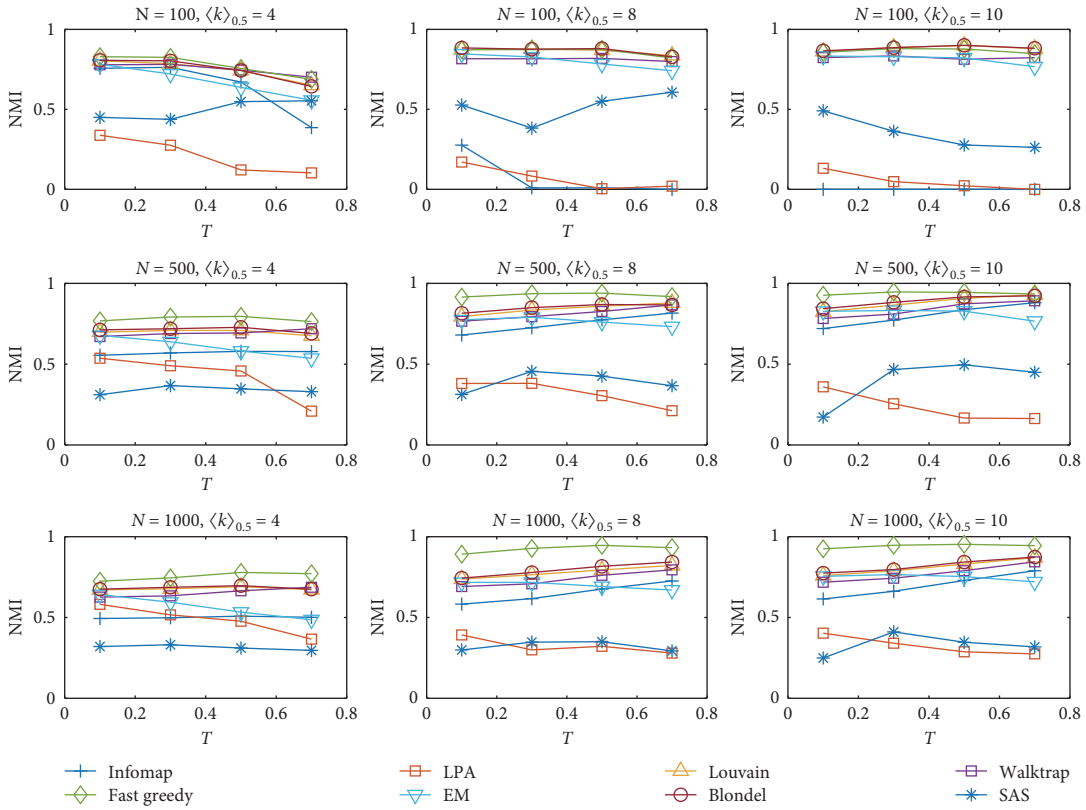
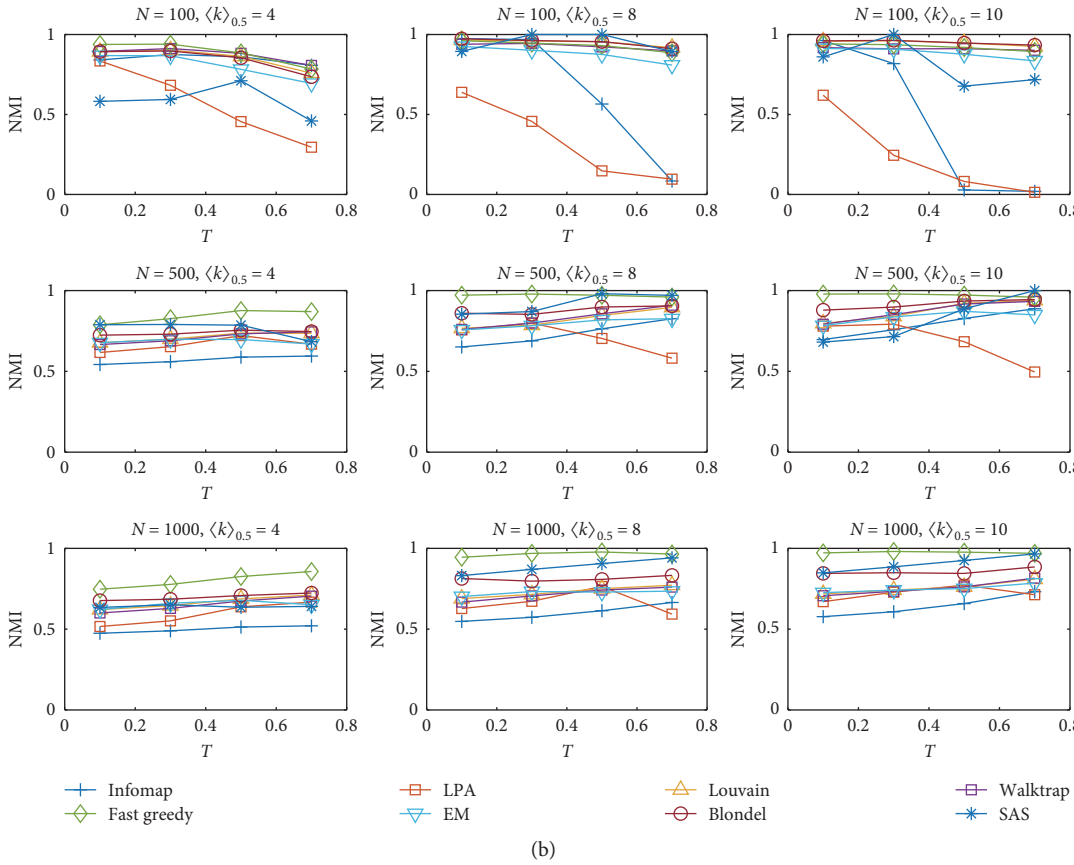


Figure 7: Community detection on GN networks. For each combination of parameters, 100 networks have been generated. And for each network, the community detection methods Infomap, LPA, Louvain, Walktrap, Fast greedy, EM, and Blondel have been executed. The communities detected have been compared by computing the normalized mutual information (NMI). The parameters $\beta = 3$, $\gamma = 0.9$, and $\rho = 0.3$.



(a)

Figure 8: Continued.



(b)

Figure 8: Community detection on nPSO networks. For each combination of parameters, 100 networks have been generated. And for each network, the community detection methods Infomap, LPA, Louvain, Walktrap, Fast greedy, EM, and Blondel have been executed. The communities detected have been compared by computing the normalized mutual information (NMI). The number of communities n_c is 3, and the parameter γ_{nPSO} is 2 and 3 in (a) and (b), respectively. The parameters $\beta = 3$, $\gamma = 0.9$, and $\rho = 0.7$.

will affect the accuracy of our algorithm. Next, we will use the nPSO benchmark to conduct the further analysis of the performance of our algorithm.

3.3. The nPSO Benchmark. Recently, there is a new network generative model named nonuniform popularity similarity optimization (nPSO) for evaluation of community detection and link prediction that can create synthetic networks with controlled parameters: network scale, average degree, community number, power-law exponent, and temperature. It allows one to tune the mixing property of networks by temperature. In particular, this model simulates how random geometric graphs grow in the hyperbolic space, generating realistic networks with clustering, small-worldness, scale-freeness, and rich-clubness.

In this part, we generate the nPSO hyperbolic networks with community with these parameters: $N = [100, 500, 1000]$ (network size), $\langle k \rangle_{0.5} = [4, 8, 10]$ (half of average degree), $T = [0.1, 0.3, 0.5, 0.7]$ (temperature, inversely related to the clustering coefficient), $n_c = [3, 6, 9]$ (number of communities), and $\gamma_{nPSO} = [2, 3]$ (power-law degree distribution exponent). We also compare the SAS algorithm with state-of-the-art community detection algorithms. From the results in Figures 8–10, we find that the performance of SAS algorithm is not sensitive to the change of parameters N, $\langle k \rangle_{0.5}$, and n_c. However, it performs well in the heterogeneous network with $\gamma_{nPSO} = 3$ and generally with $\gamma_{nPSO} = 2$. This indicates that our algorithm may be more suitable for heterogeneous networks with larger power-law exponent. Combining all the detection results, we can see that the SAS algorithm has some advantages over other state-of-the-art algorithms, and its accuracy ranks high among those algorithms in some benchmarks. The near linear time complexity is also an advantage of our algorithm.

4. Conclusions

In this paper, the performance of SAS algorithm is evaluated with some state-of-the-art algorithms in real-world networks as well as three benchmark graphs, traditionally used in the existing literatures. First, experimental results show that it is feasible to use different affinities for *strong* and *weak* communities. Our algorithm improves the accuracy of *weak* communities, compared with some algorithms based on single affinity, and has the same reliability as some state-of-the-art algorithms. Second, some heuristic algorithms based on *hub* node may need to analyze the network degree distribution or clustering coefficient in advance to improve the accuracy of the algorithm. The weakening of the role of

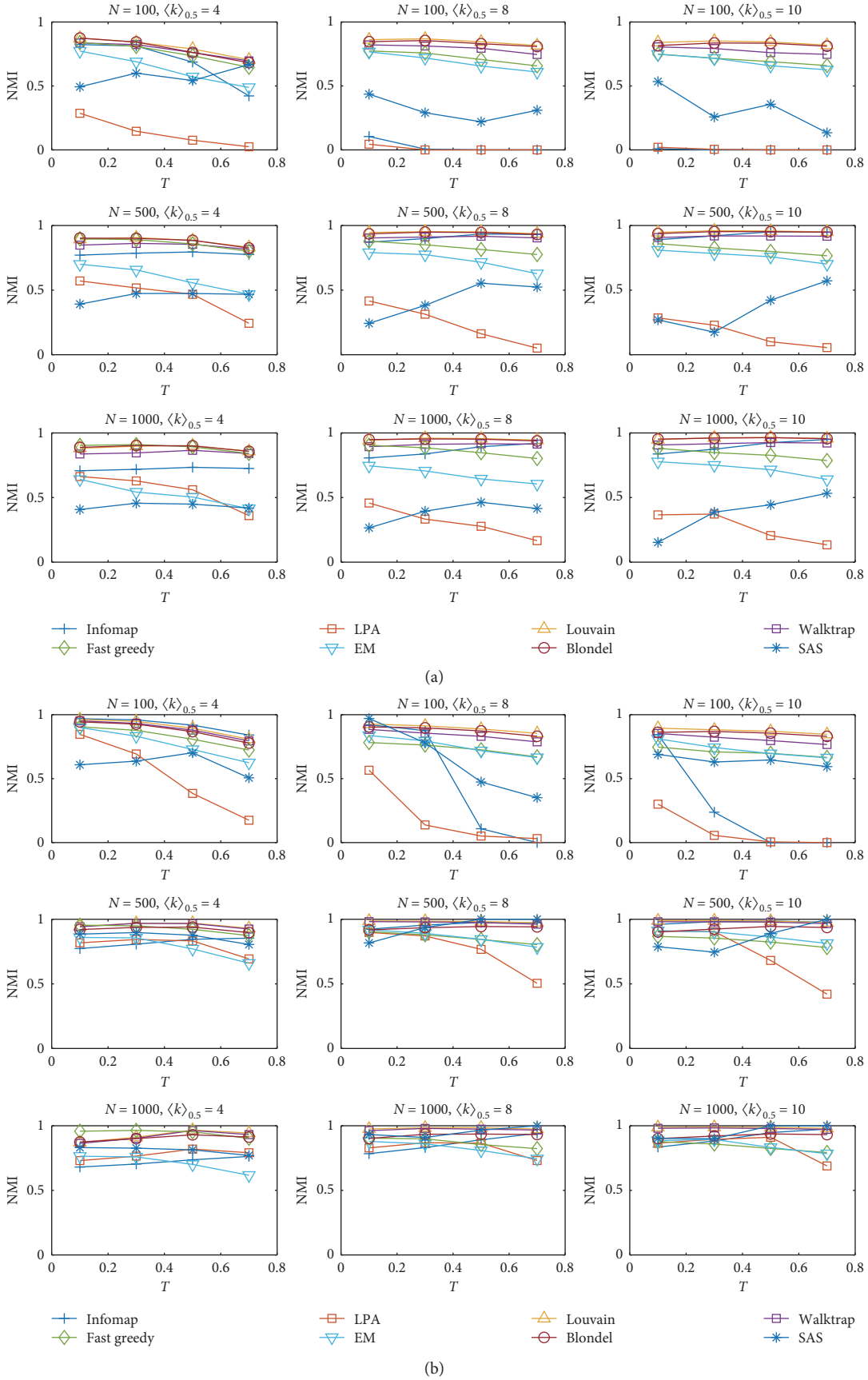


Figure 9: Community detection on nPSO networks. The number of communities n_c is 6, and other parameters are all the same with Figure 8.

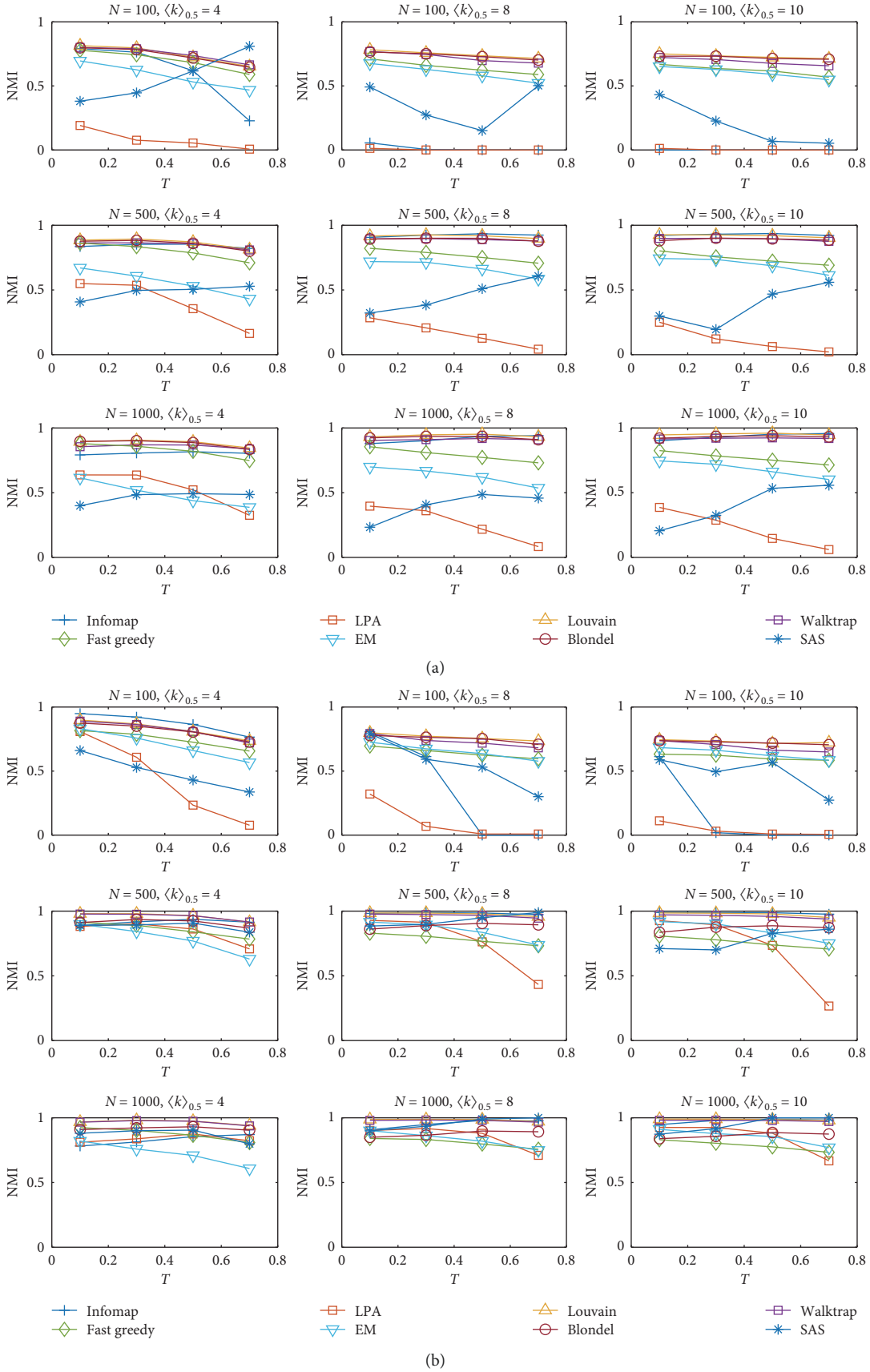


Figure 10: Community detection on nPSO networks. The number of communities n_c is 9, and other parameters are all the same with Figure 8.

hub nodes may be the reason why our algorithm performs bad in nPSO benchmark with power-law exponent 2, but performs well in LFR benchmark and nPSO benchmark with power-law exponent 3. This is also an important direction of algorithm improvement in the future. Last, our definitions of affinity are based on the concept of common neighbours. Recently, there is a new paradigm to define affinities that not only uses the information associated with the number of common neighbours but also considers (and integrates) the information associated with the links that occurs between the common neighbours. The union of common neighbours and their cross-links is named as local community, and the redefinition of affinities based on common neighbours in function of local communities has demonstrated to significantly boost link prediction in both monopartite and bipartite networks. If the SAS algorithm adopts affinities based on the local community paradigm, instead of the simple common neighbours' paradigm, we guess that this possible innovation may make our algorithm more suitable for heterogeneous networks with smaller power-law exponent.

Data Availability

Previously reported data were used to support this study and are available at Mark Newman's network data (see http://www-personal.umich.edu/~mejn/netdata/) and the algorithm LFR procedure is available at https://github.com/eXascaleInfolab/LFR-Benchmark_UndirWeightOvp#changelog. The original authors have already made the data freely available. These prior studies (and datasets) are cited at relevant places within the text as references [4, 15].

Conflicts of Interest

The authors declare that no conflicts of interest exist in the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 61873154, 11331009, and 11601294 and BNU Interdisciplinary Research Foundation for the First-Year Doctoral Candidates (no. BNUXKJC1806).

References

[1] J. Gao, B. Barzel, and A.-L. Barabási, "Universal resilience patterns in complex networks," *Nature*, vol. 530, no. 7590, pp. 307–312, 2016.

[2] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[3] Y. Yu, G. Xiao, J. Zhou et al., "System crash as dynamics of complex networks," *Proceedings of the National Academy of Sciences*, vol. 113, pp. 11726–11731, 2016.

[4] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[5] M. A. Porter, J. P. Onnela, and P. J. Mucha, "Communities in networks," *Notices of the American Mathematical Society*, vol. 56, no. 9, pp. 1082–1097, 2009.

[6] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

[7] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[8] A. Zeng, Z. S. Shen, J. L. Zhou et al., "Increasing trend of scientists to switch between topics," *Nature Communications*, vol. 10, no. 1, p. 3439, 2019.

[9] J. P. Zhang and Z. Jin, "Epidemic spreading on complex networks with community structure," *Applied Mathematics and Computation*, vol. 219, no. 6, pp. 2829–2839, 2012.

[10] X.-L. Peng, M. Small, X.-J. Xu, and X. Fu, "Temporal prediction of epidemic patterns in community networks," *New Journal of Physics*, vol. 15, no. 11, Article ID 113033, 2013.

[11] Y. Q. Hu, H. B. Chen, P. Zhang, M. H. Li, Z. R. Di, and Y. Fan, "Comparative definition of community and corresponding identifying algorithm," *Physical Review E*, vol. 78, no. 2, Article ID 026121, 2008.

[12] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences*, vol. 101, no. 9, pp. 2658–2663, 2004.

[13] D. Zhan, F. Xi, Y. Zhan, F. Don, and K. Hirota, "Fuzzy analysis of community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 22, pp. 5319–5327, 2010.

[14] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, Article ID 046110, 2008.

[15] S. Fortunato and A. Lancichinetti, "Community detection algorithms: a comparative analysis," *Physical Review E*, vol. 80, no. 5, Article ID 056117, 2009.

[16] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Article ID 066111, 2004.

[17] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 2004.

[18] A. Arenas and A. Díaz-Guilera, "Synchronization and modularity in complex networks," *The European Physical Journal Special Topics*, vol. 143, no. 1, pp. 19–25, 2007.

[19] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, "Detecting complex network modularity by dynamical clustering," *Physical Review E*, vol. 75, no. 4, Article ID 045102, 2007.

[20] J. S. Wu, L. C. Jiao, C. Jin et al., "Overlapping community detection via network dynamics," *Physical Review E*, vol. 85, no. 1, Article ID 016115, 2012.

[21] M. Rosvall and C. T. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks," *Proceedings of the National Academy of Sciences*, vol. 104, no. 18, pp. 7327–7331, 2007.

[22] E. Massaro, F. Bagnoli, A. Guazzini, and P. Lió, "Information dynamics algorithm for detecting communities in networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 11, pp. 4294–4303, 2012.

[23] D. M. N. Maia, J. E. M. de Oliveira, M. G. Quiles, and E. E. N. Macau, "Community detection in complex networks via adapted Kuramoto dynamics," *Communications in*

Nonlinear Science and Numerical Simulation, vol. 53, no. 11, pp. 130–141, 2017.

[24] T. Wang, L. Yin, and X. Wang, "A community detection method based on local similarity and degree clustering information," *Physica A: Statistical Mechanics and Its Applications*, vol. 490, pp. 1344–1354, 2018.

[25] F. D. Zarandi and M. K. Rafsanjani, "Community detection in complex networks using structural similarity," *Physica A: Statistical Mechanics and its Applications*, vol. 503, pp. 882–891, 2018.

[26] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 9, Article ID P, 2005.

[27] A. Muscoloni and C. V. Cannistraci, "A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities," *New Journal of Physics*, vol. 20, no. 5, Article ID 052002, 2018.

[28] R. Albert, H. Jeong, and A. L. Barabasi, "Internet: diameter of the world-wide web," *Nature*, vol. 401, no. 6479, pp. 130-131, 1999.

[29] M. E. J. Newman, *Networks: An Introduction*, Oxford University Press, New York, NY, USA, 2010.

[30] A. D. Broido and A. Clauset, "Scale-free networks are rare," *Nature Communications*, vol. 10, no. 1, p. 1017, 2019.

[31] X. Zhang, T. Martin, and M. E. J. Newman, "Identification of core-periphery structure in networks," *Physical Review E*, vol. 91, no. 3, Article ID 032803, 2015.

[32] Y. Chen, P. Zhao, P. Li, K. Zhang, and J. Zhang, "Finding communities by their centers," *Scientific Reports*, vol. 6, no. 1, p. 24017, 2016.

[33] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[34] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[35] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, Article ID 036106, 2007.

[36] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2018, no. 10, Article ID P10008, 2018.

[37] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Proceedings of the International Symposium on Computer and Information Sciences*, pp. 285–293, Springer, Istanbul, Turkey, October 2005.

[38] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phyical Review E*, vol. 74, no. 3, Article ID 036104, 2006.

[39] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of community hierarchies in large networks," 2008, https://arxiv-web.arxiv.org/abs/0803.0476v1.