Local Community Detection Algorithm Based on Alternating Strategy of Strong Fusion and Weak Fusion

Ronghua Shang, *Member, IEEE*, Weitong Zhang, *Member, IEEE*, Jingwen Zhang, Licheng Jiao, *Fellow, IEEE*, Yangyang Li, *Senior Member, IEEE*, and Rustam Stolkin

***Abstract*—Existing fusion-based local community detection algorithms have achieved good results. However, when assigning a node to a community, similarity functions are sometimes used, which only use node information, while ignoring connection information within the community. These algorithms sometimes fail to find influential nodes, which eventually leads to the failure to find a complete local community. To address these problems, a new local community detection algorithm is proposed in this article. Two strategies, of strong fusion followed by weak fusion, are used alternately to fuse nodes. Compared with using two fusion strategies alone, the alternating loop method can improve the solution of the algorithm in each stage. In strong fusion, we propose a new membership function that considers both node information and connection information in the local community. This improves the quality of the fused node while preserving the structure of the current community. In weak fusion, we propose a parameter-based similarity measure, which can detect influential nodes for a local community. We also propose a local community evaluation metric, which does not require true division to determine the optimal local community under different parameters. Experiments, compared to six state-of-the-art algorithms, show that the proposed algorithm improves accuracy and stability, and also demonstrate the effectiveness of the new local community evaluation metrics in parameter selection.**

***Index Terms*—Evaluation metrics, local community detection, membership function, strong fusion, weak fusion.**

I. Introduction

COMPLEX networks are a kind of network that exhibits self-similarity, small world, scale-free, and other useful properties. They can be used to describe a variety of difficult systems in real-world applications [1]. The entities in a system can be abstracted into network nodes, and the relationships between entities can transform into edges [2]. Therefore, the composition and function of such systems can be analyzed by studying complex networks [3]. For example, in social networks, each individual is a node, and there are edges between connected nodes. Finally, by performing community detection on the constructed social network, different types of groups in the network can be obtained, and more active and influential individuals in each group can be identified [4]. The community structure in a complex network can be intuitively expressed as a node set composed of multiple nodes, and a complex network can be divided into multiple communities [5]. Studying the community structure of a complex network, that is, discovering the smaller substructures in the network, helps to further visualize and analyze complex systems and is the key to understanding and revealing the organizational principles in such networks [6], [7]. Community detection has applications in many fields, ranging from biology, power systems, physics, and statistics to sociology. It is of great significance for the research of applications, such as trajectory clustering, recommendation system, anti-terrorism, virus propagation, and routing protocols [8].

Manuscript received 11 May 2021; revised 9 October 2021, 26 December 2021, and 16 February 2022; accepted 6 March 2022. Date of publication 25 March 2022; date of current version 13 January 2023. This work was supported in part by the National Key R&D Program of China; in part by the National Natural Science Foundation of China under Grant 62176200 and Grant 61871306; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2022JC-45; in part by the Open Research Projects of Zhejiang Lab under Grant 2021KG0AB03; and in part by the Guangdong Provincial Key Laboratory under Grant 2020B121201001. This article was recommended by Associate Editor X. Wang. *(Corresponding author: Weitong Zhang.)*

Ronghua Shang, Weitong Zhang, Jingwen Zhang, Licheng Jiao, and Yangyang Li are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University, Xi'an 710071, Shaanxi, China (e-mail: rhshang@mail.xidian.edu.cn; wtzhang_1@xidian.edu.cn; jwenzhang@stu.xidian.edu.cn).

Rustam Stolkin is with the School of Mechanical Engineering, University of Birmingham, Birmingham B15 2TT, U.K.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3159584.

Digital Object Identifier 10.1109/TCYB.2022.3159584

A. Related Work on Community Detection Algorithms

Global community detection attempts to divide the overall network into multiple communities. The main detection methods include label propagation algorithm, non-negative matrix decomposition, deep learning, and evolutionary clustering [9]. Local and global structure information are vital in community detection [10]. However, with the increase of data scale, the running time and memory required for global community detection increase [11], especially the evolutionary algorithms [12]. Many times, the overall community structure may often not be necessary [13]. With the increase of the network scale, the network's community structure becomes unclear. It is challenging to perform global detection on a large-scale network, which requires extensive computing resources. Sometimes, when a single node or a single community structure is focused on, using local community detection methods is most efficient [14]. Local community detection only finds the community where the initial node is located [15]. This initial node can be any point on the network, also known as a seed. Compared with global community detection, local community detection only focuses on the network part of the interest [16]. It uses the local information of the initial node to generate the entire local community. It is faster and more effective for some practical applications, avoiding computational expense [17]. However, due to the lack of neighborhood community and other information and the influence of noise and redundant information in the network topology, it cannot be easy to find local communities. Such algorithms are prone to convergence on local optima.

The fundamental challenge in local community detection is how to expand the community from the initial node. The modularity function and the relation strength function determine the effectiveness of these methods. The modularity function is a measure of the community structure. The nodes within a community are closely connected, while links from within the community to outside the community are relatively sparse [18]. Newman proposed the modularity function Q [19] to evaluate the quality of global community detection. The larger the value, the better the algorithm has performed. There are many variations of the modularity function, and some multiobjective optimization algorithms [20] have been used to optimize multiple modularity functions. For local community detection, Tabarzad and Hamzeh [21] proposed a local modularity R. Luo *et al.* [22] proposed another local modularity M. Both R and M are used as maximization optimization functions to solve local community detection. When there is no core part in a local community, the values of R and M are equal [23]. During community detection, the discovered communities require their internal connections to be tight and require that the nodes in the community are similar. Therefore, many similarity functions based on node neighbor information have been proposed [24]. The relationship strength function is used to measure the similarity between nodes or the membership between the community and its neighboring nodes. However, these functions only use node information and ignore the connection information within local communities. While we observe that this topology information can help to improve the accuracy of community expansion.

Local community detection algorithms based on fusion strategy and membership function have achieved useful results. M-based method and R-based method are two greedy local community detection algorithms by maximizing the local modularity M and R. They are sensitive to the position of the seed nodes and this can lead to obtaining low detection accuracy. Some other local community extension methods have been proposed in the past few years. These methods have enabled mining more connection information by improving modularity and similarity functions, or designing new algorithms according to the local community structure. Chen and Wu [25] proposed replacing the seed with a local central node to expand the community, which overcomes the sensitivity of seed position. Jiang *et al.* [26] proposed complete local influence to detect nuclear nodes as seed points. However, this method can only obtain the local community of specific nodes in the network. Chen *et al.* [27] proposed local community detection based on local centrality nodes. They use the node with the largest centrality in the first multilayer neighbor of the seed node to replace it as the initial node of local community expansion. However, this method does not consider the similarity between the core node and the original seed node. Ding *et al.* [28] further improved the algorithm by using core nodes to replace seeds and advanced two rules for expanding local community. A neighboring node can be integrated into the community as long as it meets one of the two rules, which improves the use of node information. This algorithm only uses one fusion strategy in each stage of the algorithm, which makes it easy to obtain a very closely connected local community in the initial stage of the algorithm. Although the modularity is high, it is impossible to expand nodes on a large scale in the subsequent stage of the algorithm, and finally leads to the local optimization of the entire algorithm. In [29], the local community detection method based on nearest nodes with greater centrality used the nearest node with greater centrality (NGC) for local community detection. It calculated the fuzzy relation between the node and its NGC node. Luo *et al.* [30] designed two methods: local community detection with the dynamic membership function R and M (DMF_R and DMF_M). Both methods include three stages. At each stage, the author considered the characteristics of the local community, and then proposed three corresponding dynamic membership functions based on these characteristics. Liakos *et al.* [31] proposed an algorithm uncovering local hierarchical overlapping communities at scale (LDLC). It used a new similarity function S improved by function *Jaccard* to calculate the similarity of two pairs of links. LDLC focused on the similarity of two links that shared a common node and merged links gradually in the ranking order according to S. These methods do not specifically find some influential nodes for the local community, resulting in their failure to obtain a complete community. The influential nodes refer to some central nodes in a local community, which ensure that the community can continue to expand. Without the nodes with rich connection information and influence, algorithms are prone to stopping early after only identifying a subset of the true local community.

B. Motivation and Innovation

To address the above challenges, this article proposes a local community detection algorithm with an alternating cyclic strong fusion and weak fusion strategy (ASFWF). ASFWF uses strong fusion and weak fusion strategies to alternately integrate nodes into local communities. Its characteristic is that after each strong fusion, a weak fusion is executed immediately. The alternating cycle is carried out until no node satisfies the strong fusion condition. The strong fusion strategy aims at the structure of the local community, while the weak fusion strategy is to find influential nodes.

Strong integration is a fusion method based on optimizing two modularity functions. Each integration node must increase the modularity value of local communities to ensure the structure of communities. In the strong fusion, we propose a new membership function. This function not only uses the node information but also considers the connection information within the local community. This function can better judge the complex relationship between nodes and communities, so as to improve the accuracy of local community detection. The weak fusion method proposed in this article is a parameter-based fusion strategy. The ability of the strategy expanding the community can be improved by adjusting the parameters. In weak fusion, we propose another membership function, which is designed to ensure that those nodes with strong relationships with local communities are finally integrated into the community, and the point with the largest node degree becomes the influential node. In order to set the parameters correctly, we also propose a local community evaluation index without real label which we incorporate into this algorithm. Parameter-based weak fusion helps to integrate nodes with rich information into local communities, so as to increase local available information and increase the detection width of local communities. ASFWF, which alternately circulates the strong and weak fusion strategies, not only integrates the nodes with high accuracy but also expands the influential nodes in the process of community detection. So, the algorithm can incorporate different types of nodes in each stage, and improve the accuracy of final local community detection.

The main contributions of this article are as follows.

1) Two fusion strategies are used, called: a) strong fusion and b) weak fusion. In contrast to previous methods, the two strategies are used alternately in an iterative cyclic process. This alternating cycle mechanism enables the algorithm to integrate different types of nodes in each stage.
2) In strong fusion, a new membership function is defined. The function uses not only the neighbor information of the node but also the connection information in the local community, which can better discriminate the membership relationship between the nodes and the communities.
3) Another new membership function is designed for use in the weak fusion process, and the ability of the fusion strategy to expand the community can be controlled by adjusting a parameter. In order to select the parameter correctly, a local community evaluation index without real community division is proposed.

The remainder of this article is arranged as follows. Section II introduces the design and overall flow of ASFWF. Section III details the experiments and the simulation results. Section IV gives the analysis of parameters and fusion iteration mode. Section V provides concluding remarks.

II. Proposed Algorithm

The strong fusion and weak fusion strategies alternate in the process of community discovery, so that the algorithm can integrate different types of nodes in each stage. The block diagram of the proposed algorithm is shown in Fig. 1.

This section first introduces the two fusion strategies used in our method. Then, the complete process and analysis of ASFWF are presented. Finally, the new evaluation metrics are given.

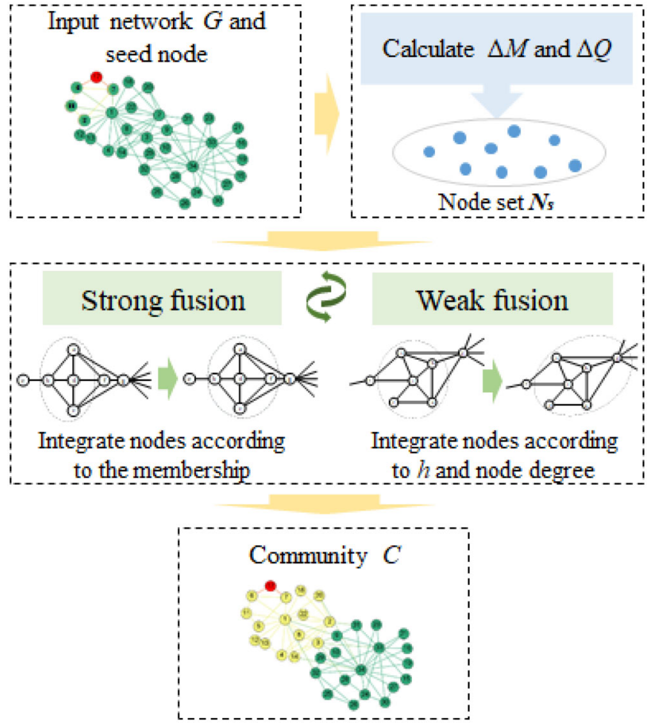


Fig. 1. Block diagram of the proposed algorithm.

A. Strong Fusion

Strong fusion is a strategy for merging nodes based on the modularity and membership functions. In networks without real labels, the modularity function Q is often used to evaluate community detection algorithms. The focus of local modularity R is to divide the local community into a boundary part and a core part. The boundary part refers to the community's nodes that are still connected with the outside. The core part refers to the nodes that are not connected with the outside of the community.

The local modularity R is the ratio of the number of community edges connected with the boundary node to the number of edges connected with the boundary node by the entire network. The local modularity M is the ratio of the total number of edges in the local community to the number of edges from nodes in the community to nodes outside the community. When there is no core part in the local community, the values of R and M are equal. For the similarity function based on node neighbor information, the similarity function *CN* only uses the number of common neighbors between two nodes to determine the similarity value, ignoring the characteristics of the nodes themselves. The theoretical basis of the similarity function *Jaccard* is that the greater the ratio of the common neighbors of two nodes, the better. The similarity functions *HP* and *HD* are the common neighbors to the minimum node degree and maximum node degree, respectively. In contrast to other local community detection algorithms, ASFWF uses two modularity functions: 1) modularity M and 2) modularity Q_{lcd} adjusted by modularity Q, to optimize the community jointly. The definition of M is

$$M = \frac{e_{\text{in}}}{e_{\text{out}}} \tag{1}$$

where e_{in} is the number of edges in the local community, and e_{out} is the number of edges connecting points within the local community to points outside the community.

The Q_{lcd} is defined as follows:

$$Q_{\text{lcd}} = \frac{e_{\text{in}}}{S} - \left(\frac{d_{\text{in}}}{2S}\right)^2 = \frac{e_{\text{in}}}{S} - \left(\frac{2e_{\text{in}} + e_{\text{out}}}{2S}\right)^2 \tag{2}$$

where d_{in} is the sum of the degree of all nodes in the local community and satisfies $d_{\text{in}} = 2e_{\text{in}} + e_{\text{out}}$. S is the total number of edges in the entire network.

The greater the values of modularity M and Q_{lcd}, the more the local communities meet the requirement of tight internal connections and sparse external connections. Therefore, in strong fusion, the node to be merged in the local community must satisfy $\Delta M \geq 0$ and $\Delta Q_{\text{lcd}} \geq 0$ at the same time. Define the nodes that meet this requirement as the set N_s, and the final fusion node is selected from it. ΔM and ΔQ_{lcd} can be calculated as follows:

$$\Delta M = \frac{e_{\text{in}} + x}{e_{\text{out}} - x + y} - \frac{e_{\text{in}}}{e_{\text{out}}} \tag{3}$$

$$\Delta Q_{\text{lcd}} = \frac{e_{\text{in}} + x}{S} - \left(\frac{2e_{\text{in}} + e_{\text{out}} + x + y}{2S}\right)^2 \tag{4}$$

where x represents the number of neighbors of node v_i to be fused in local community C, and y indicates the number of neighboring nodes not in the local community C. So, the sum of x and y is equal to the degree of the node v_i.

Formula (1) shows that when a community includes the entire network, M will be infinite. Although it satisfies $\Delta M \geq 0$, it is obviously not wanted. For avoiding this problem in some small networks, this article uses two optimization goals M and Q_{lcd}. Q_{lcd} introduces global information, that is, the total number of network edges. From (2), Q_{lcd} is nonzero, so if the entire network is integrated into a local community, the value of Q_{lcd} will be 0, which does not meet the condition of $\Delta Q_{\text{lcd}} \geq 0$. Limit the situation of excessive fusion.

After obtaining the node set N_s from the two modularity functions, the node from N_s that maximizes the membership function $f_{m1}(v_i)$ is finally selected as the fusion node in strong fusion. The definition of the formula $f_{m11}(v_i, v_j)$ is as follows:

$$f_{m11}(v_i, v_j) = \frac{\left|N(v_i) \bigcap N(v_j)\right| + 1}{\left|N(v_j)\right|}, v_j \in N(v_i) \bigcap C \tag{5}$$

where $N(v_i)$ indicates the neighboring nodes set of node v_i, and v_j belongs to the intersection of the neighboring nodes of v_i and the local community C. $|N(v_j)|$ represents the number of neighboring nodes of v_j. Analogy $f_{m11}(v_i, v_j)$ gives the definition of $f_{m12}(v_i, v_j)$

$$f_{m12}(v_i, v_j) = \frac{\left|NN(v_i) \bigcap NN(v_j)\right| + 1}{\left|NN(v_j)\right|}, v_j \in N(v_i) \bigcap C \tag{6}$$

where $NN(v_i)$ represents the second-layer neighboring nodes set of v_i, which is defined as $NN(v_i) = \{v_k | v_k \in N(N(v_i))\}$. $f_{m12}(v_i, v_j)$ is the number of second-layer neighbors of nodes v_i and v_j compared to the number of second-layer neighbors of v_i. Equations (5) and (6) calculate the similarity between two nodes. Based on these two formulas, the membership function $f_{m1}(v_i)$ used in strong fusion is given as

---

Algorithm 1 Strong Fusion

Input: network G, local community C, neighboring node set N;
Output: local community C;
initialize $N_s = \varnothing$;
for each $v_n \in N$:
  if $\Delta M \geq 0$ & $\Delta Q_{lcd} \geq 0$:
    put v_n to N_s;
  end if
end for
for each $v_i \in N_s$:
  Calculate $f_{m1}(v_i)$, save max $f_{m1}(v_i)$ as v_{best};
end for
Add v_{best} in C;
Return C.

$$f_{m1}(v_i) = \max_{v_j \in N(v_i) \bigcap C} d_{\text{in}}(v_j)\left(f_{m11}(v_i, v_j) + 0.1 * f_{m12}(v_i, v_j)\right) \tag{7}$$

where $d_{\text{in}}(v_j)$ calculates the number of connected edges of v_j in community C, which is called the node degree of v_j in community C. For constant 0.1 of $f_{m12}(v_i, v_j)$, function (7) uses the first two layers of neighbor information of the node. Compared to the similarity function using only one layer of neighbors, it can better utilize the connection relationship between nodes.

As the local community expands, the distribution and connections within the local community change, dramatically impacting the membership strength of nodes and the community. Therefore, this article introduces $d_{\text{in}}(v_j)$, which takes the connection information of node v_j in the local community as its influence in the local community. The greater this influence, the greater the membership relationship between nodes similar to v_j and local communities. In particular, when (7) is applied to the initial local community, there is only a seed node in the community and there is no connection information, that is $d_{\text{in}}(v_j) = 0$ and $f_{m1}(v_i) = 0$, it cannot judge the membership strength. So, $f_{m1}(v_i)$ is finally defined as

$$f_{m1}(v_i) = \begin{cases} f_{m11}(v_i, v_{\text{seed}}) + 0.1 f_{m12}(v_i, v_{\text{seed}}), & |C| = 1 \\ \max_{v_j \in N(v_i) \bigcap C} d_{\text{in}}(v_j)\left(f_{m11}(v_i, v_j) + 0.1 f_{m12}(v_i, v_j)\right), & \text{else} \end{cases} \tag{8}$$

where v_{seed} is the seed node. The membership function $f_{m1}(v_i)$ is used in strong fusion to calculate the intimacy relationship between node v_i and the local community C. The larger the value, the more v_i belongs to community C. So, the node with the largest f_{m1} in N_s is chosen to be integrated. Finally, strong fusion is summarized as Algorithm 1.

B. Weak Fusion

In strong fusion, the nodes of each fusion can increase the modularity of local community. This means that the new fusion nodes must make the internal connection of the local community is close, and the external connection is sparse. Unlike this effect of strong fusion, the nodes integrated by the second fusion strategy do not necessarily increase the modularity of the local community. It only selects the fusion node based on the membership strength and the maximum node

TABLE I
VALUE RANGE OF h AND THE CORRESPONDING FUSION STRENGTH

h	0.5	0.6	0.7	0.8	0.9	1	1.1
Fusion strength	Strong	→	Weak	→	No weak fusion		

degree, which we call weak fusion. Also, for this reason, the weak fusion strategy can bring more influential nodes to local community than the strong fusion strategy.

A new membership function is proposed in weak fusion, which defines the closeness between neighboring node v_i and local community C. First, the membership functions $f_{m21}(v_i)$ and $f_{m22}(v_i)$ are defined as follows:

$$f_{m21}(v_i) = \max_{v_j \in N(v_i) \bigcap C} \frac{\left|N(v_i) \bigcap N(v_j)\right| + 1}{|N(v_i)|} \tag{9}$$

$$f_{m22}(v_i) = \frac{\left|N(v_i) \bigcap C\right|}{|C|}. \tag{10}$$

Formula (9) is different from (5) in the strong fusion strategy. $f_{m11}(v_i, v_j)$ defines the similarity of node v_i to v_j, that is, the ratio of common neighbors of the two nodes to the neighbors of v_j, while $f_{m21}(v_i)$ defines the similarity of node v_j to v_i. In (9), v_j belongs to the neighboring nodes of v_i and is also in the local community. The value that maximizes the similarity between v_j and v_i is selected as the membership between v_i and local community C. $|C|$ indicates the number of nodes in the local community. So, $f_{m22}(v_i)$ calculates the ratio of the number of neighboring nodes of v_i in local community C to the number of all nodes in local community C.

The larger the value of the above two membership functions, the closer the node is to the local community, and the values of the two membership functions are both in the range (0, 1]. Finally, membership function $f_{m2}(v_i)$ in weak fusion is defined as the maximum value in $f_{m21}(v_i)$ and $f_{m22}(v_i)$

$$f_{m2}(v_i) = \max\{f_{m21}(v_i), f_{m22}(v_i)\}. \tag{11}$$

Membership function $f_{m2}(v_i)$ defines the close relationship between node and local community from two aspects. $f_{m21}(v_i)$ is an evaluation index based on the maximum similarity between two nodes. As long as a node is sufficiently similar to a point in the local community, it is identical to the local community. $f_{m22}(v_i)$ defines the strength of intimacy based on the number of connections between v_i and local community. When more nodes within local community C are connected to v_i, the closer the node is to the community.

As mentioned earlier, weak fusion is a parameter-based fusion strategy. The specific method is to add the nodes satisfying $f_{m2}(v_i) \geq h$ to N_s. Then, select the node with the largest degree from N_s as the fusion node in weak fusion. These make the node fused by weak fusion have both high membership and great influence, where h is a variable parameter, its value determines the strength of the weak fusion merging node. The range of h and the corresponding fusion strength are shown in Table I.

Table I shows that when h takes the minimum value of 0.5, there are more nodes satisfying $f_{m2}(v_i) \geq h$, so the strength of the weak fusion merging nodes is greatest. With the increase

Algorithm 2 Weak Fusion

Input: network G, local community C, neighboring node set N;
Output: local community C;
initialize $N_s = \varnothing$;
for each $v_n \in N$:
  if $f_{m2}(v_n) \geq h$:
    put v_n to N_s;
  end if
end for
for each $v_i \in N_s$:
  save max $d(v_i)$ as v_{best};
end for
  Add v_{best} in C;
Return C.

Algorithm 3 ASFWF

Input: network G, seed node v_{seed};
Output: local community C;
put v_{seed} in to local community C;
get neighbor set N;
$M = 0$, $Q_{lcd} = 0$;
set num_1, $num_2 = 0$;
While (True)
  Expand C by Strong fusion;
  $num_1 = |C|$;
  Update N, M and Q_{lcd};
  if $num_1 == num_2$:
    break
  end if
  Expand C by Weak fusion;
  $num_2 = |C|$;
  Update N, M and Q_{lcd};
end while
Return C.

of h, the fusion strength gradually decreases until the value of 1.1, where the weak fusion function is no longer able to add nodes to the local community.

The reason why parameter h starts from 0.5 and discards the relationship strength between 0.1 and 0.4 is determined by the definition of $f_{m22}(v_i)$. From (9) and (10), it can be seen that the value of $f_{m22}(v_i)$ is determined by the maximum relationship strength between node v_i and a certain node in the local community, or by the participation of the neighbors of v_i in the local community. The strength of the two memberships is in the range (0, 1], but when the value is less than 0.5, it means that the relationship between the node v_i and the local community is weak, which is not enough to join the local community. In other words, when the value of $f_{m22}(v_i)$ is less than 0.5, the node v_i will most likely not belong to the local community. So, when the value of h is between 0.1 and 0.4, weak fusion will integrate a large number of error nodes and reduce the accuracy of local community detection. Algorithm 2 gives the weak fusion.

To sum up, Algorithm 3 is the overall pseudocode procedure of ASFWF.

C. New Metrics

As mentioned above, the value of h can be used to control the intensity of weak fusion. So, the value of h that can make the local community closest to the true division should be chosen. However, the common evaluation metrics *f-score* and *NMI* for local community detection are based on real division, while h cannot be chosen based on these results. Therefore, this article proposes an evaluation metric M_C based on modularity M that does not require the real division to help the algorithm choose the appropriate h. The higher the value of M_C, the better the quality of local community C. Making the M_C reach the maximum value of h is the h that makes the ASFWF reach the optimal performance. The definition of M_C is as follows:

$$M_C = \frac{M}{\sqrt{|C|}} \tag{12}$$

where $|C|$ is the number of nodes in local community C.

The modularity M cannot be used as an evaluation index, because the value of M pursues a large number of connected edges in the local community, and a small number of connected edges from the inside to the outside. It does not consider the validity of internal edges. Here, the erroneous fused nodes are known as invalid nodes, and the edges connected by invalid nodes in the local community are called invalid edges. When the local community is overly integrated, a large number of invalid nodes will be integrated, and the number of internal edges will increase. Finally, a local community with a large M value, but inconsistent with the real partition, will be obtained.

Therefore, M_C with M compared with the square root of the number of nodes in community C, is proposed to correct the problem that M incorrectly evaluates the quality of community C due to the increase of a large number of invalid nodes. M_C is effective for selecting h value because the size of h controls the ability of ASFWF to expand the local community. The smaller the h is, the more nodes the local community expands, and the more likely it is to over fuse. Therefore, M_C makes a new evaluation of the local community in terms of quality and number of nodes. Its value can judge whether ASFWF over-extends the local community, and can help to choose the best h for the algorithm on different networks.

D. Time Complexity Analysis

Assume that the average node degree in the local community is d. For each node in the local community, the complexity of using strong fusion is $d^2\log d$, and the complexity of using weak fusion is d^2. The initial community size is 1, let n be the number of nodes in the final local community, and each time the community expands one node. So, the final time complexity of ASFWF is $O(\sum_{i=1}^{n} i*\max(d^2 \log d, d^2)) = O(n^2 d^2 \log d)$.

III. Experimental and Analysis

In order to ensure the fairness of the comparative experiments, all the experiments are implemented using the same computer configuration. All algorithms are compiled and run using the python language. The computer processor used in these experiments is Intel Core i7-8750H CPU @ 2.20 GHz, the memory is 24 GB, and the operating system is Windows10.

TABLE II
Information of Six Comparison Algorithms

Algorithms	Underlying idea	Year	Ref.
M-based	Local community extension algorithms by optimizing modularity functions M	2006	[28]
R-based	Local community extension algorithms by optimizing modularity functions R	2017	[21]
DMF_M	Three stages with three corresponding dynamic membership functions according to M	2018	[27]
DMF_R	Three stages with three corresponding dynamic membership functions according to R	2018	[27]
RTLCD	A robust two-stage local community detection algorithm, including core detecting stage and community extension stage	2018	[26]
LCDNN	Local community detection using NGC node	2020	[29]

A. Evaluation Metrics

The universal evaluation metric *fscore* is used to evaluate local community detection results. The definition is as follows:

$$\text{fscore} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{13}$$

$$\text{recall} = \frac{\left|C_{\text{Found}} \bigcap C_{\text{Ture}}\right|}{|C_{\text{Ture}}|} \tag{14}$$

$$\text{precision} = \frac{\left|C_{\text{Found}} \bigcap C_{\text{Ture}}\right|}{|C_{\text{Found}}|} \tag{15}$$

where C_{Found} is the local community detected by the algorithm, and C_{Ture} is the community where the seed is located. *Recall* and *precision* are the number of correctly classified nodes divided by the number of the nodes in C_{Ture} and C_{Found}, respectively. The *fscore*, which combines *recall* and *precision*, is more persuasive and represents the real performance of local community detection results. Its value ranges from 0 to 1. Only when the detected local community is completely correct, its value is 1.

B. Comparison Algorithms and Their Complexity

In order to compare the performance of our algorithm, six algorithms are run for verification tests. They are M-based method, R-based method, DMF_M, DMF_R, RTLCD, and LCDNN. The specific information of six comparison algorithms is shown in Table II.

C. Datasets

The experiments are carried out on two kind of datasets: 1) synthetic networks and 2) real networks.

1) Synthetic Networks: The synthetic network used in this experiment is the LFR network [32]. There are ten kinds of parameters to control LFR network generation. Among them, N is used to set the number of nodes in the network, k is the average node degree, max$k$ is the maximum node degree, and the mixed parameter μ is used to control the fuzzy degree of community structure. The larger the value, the fuzzier the community. t_1 and t_2 are used to control the degree distribution and community size distribution, respectively, min$c$ and max$c$ refer to the minimum and maximum number of nodes in the community, and *on* and *om* are parameters used to control the overlapping nodes. Six LFR networks are tested. The specific parameter values are shown in Table III.

TABLE III
PARAMETER SETTINGS FOR SYNTHETIC NETWORKS

Parameters	LFR1	LFR2	LFR3	LFR4	LFR5	LFR6
N	100	1000	1000	10000	10000	20000
k	5	15	7	7	10	15
maxk	15	50	25	20	50	50
μ	0.1	0.4	0.1	0.1	0.4	0.4
t_1	2	2	2	2	2	2
t_2	1	1	1	1	1	1
minc	10	10	10	10	10	10
maxc	50	50	300	100	200	50
on	0	0	0	0	0	0
om	0	0	0	0	0	0

TABLE IV
ALL NETWORKS INFORMATION

Networks	Type	Node	Edge	$\tilde{d}$	$\lvert\tilde{C}\rvert$
LFR1	Synthetic	100	264	5.28	50.00
LFR2	Synthetic	1000	7895	15.79	27.03
LFR3	Synthetic	1000	3269	6.54	90.91
LFR4	Synthetic	10000	37637	7.53	39.37
LFR5	Synthetic	10000	50033	10.01	61.73
LFR6	Synthetic	20000	151704	15.17	24.21
Dolphins	Social	62	159	5.13	31.00
Football	Football games	115	441	7.67	9.58
Political Books	Co-purchasing	105	615	11.71	35.00
Amazon	Co-purchasing	334863	925872	5.53	13.49
DBLP	Co-authorship	317080	1049866	6.62	22.45

2) Real Networks: Five real networks are used in the experiment. Among them, the Dolphins network [33], the Football network [18] and the Political Books network [34] are three nonoverlapping networks. The Dolphins network is a dolphin social network. An edge represents contact between two dolphins. The Football network is based on college football matches. Different communities represent different leagues, and edge means there is a confrontation between two teams. The Political Books network is a book co-purchasing network. The communities of all nodes in these three networks will be detected. The Amazon network and DBLP network are two large-scale and overlapping networks [35]. The Amazon network is a product co-purchasing network. Nodes in the network represent products and the edges connect that two products have been purchased together. The DBLP network is a coauthorship network where nodes represent authors and edges connect two authors who coauthored a paper. The details of six synthetic networks and five real networks are shown in Table IV. In Table IV, $\tilde{d}$ is the average degree of the network, and $|\tilde{C}|$ is the average size of ground-truth communities.

D. Experimental Results of Synthetic Networks

The local community detection results of the algorithm on six synthetic networks are tested. M-based method, R-based method, DMF_M, DMF_R, RTLCD, and LCDNN are used as comparison algorithms. In order to ensure the fairness of the comparative experiment, the seed point selection and evaluation index calculation methods of all algorithms are the same. For all algorithms, the same test method is used: select each node in the network as an initial node, detect the local community of the initial node by algorithm, and calculate their average *fscore*, which represents the overall performance. The standard deviation of *fscore* of local communities from all initial nodes is also given. For ASFWF, the same h value for different seed nodes on the same network is set. The best h value for each network makes the evaluation metrics M_C reaches the maximum. The experimental results are shown in Table V. The last column represents the average result of all networks.

From Table V, ASFWF improves *fscore* on both small and large networks. The average result is also better than all comparison algorithms, which means that the results of ASFWF are closer to the real local community. Among the six networks, ASFWF increases the most on LFR1, LFR3, and LFR5 networks, with a maximum increase of 0.1474. The M-based method and R-based method are two greedy local community detection algorithms. They maximize local modularity M and R, so they are very sensitive to the selection of seed nodes. RTLCD algorithm depends on the core node label information of neighbor nodes. The LCDNN algorithm is based on NGC nodes and has high dependence on node centrality. The ASFWF algorithm uses two fusion strategies alternately and circularly, so that the algorithm can integrate different types of nodes in each stage. For most of the comparison algorithms, the standard deviation of ASFWF is also reduced, that is, it can stably obtain a better result for different seed nodes.

E. Experimental Results of Real Networks

In order to test the effectiveness of ASFWF, the above six methods are also compared on Dolphins, Football, Political Books, Amazon, and DBLP networks. For the first three networks, the same test method is used for synthetic networks. For the Amazon and DBLP, 500 different communities are considered, and we randomly select a node from each community as seed. When an overlapped node is selected as seed, all the communities it belongs to will be integrated as a real local community. The results are also shown in Table VI. The last column represents the average result of all networks.

It can be seen from Table VI that ASFWF achieves the highest *fscore* on most real networks and the highest average *fscore* of all real networks. The average result is improved by at least 6%. Among them, the maximum increment is 0.1727 on the Dolphins network. Only on the Football network, ASFWF obtains a little worse result, which is only 0.0081 lower than the result of DMF_R. This indicates that ASFWF can improve the accuracy of local community detection in real networks, even in some large networks. While the membership function used by DMF_M and DMF_R mainly defines the community quality from the point of view of node edge, and does not pay enough attention to the connection information within the local community.

F. Simulation Results on Single Node

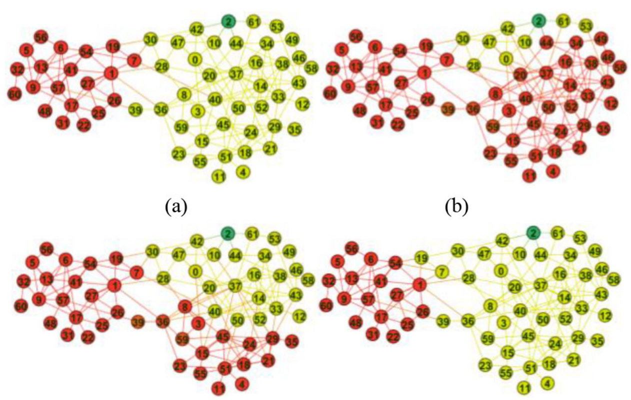
A single node is used to show how ASFWF improves the local community detection algorithm compared with DMF_M, RTLCD, and LCDNN. These algorithms obtain the top three results on the Dolphins network. The real local community

TABLE V
Fscore COMPARISON OF DIFFERENT ALGORITHMS ON SYNTHETIC NETWORKS

Algorithms	LFR1	LFR2	LFR3	LFR4	LFR5	LFR6	Average
M-based	0.2894±0.2273	0.4722±0.4560	0.1859±0.2718	0.5880±0.4117	0.1186±0.2117	0.7073±0.4378	0.3936±0.3361
R-based	0.2781±0.2165	0.4722±0.4560	0.1841±0.2684	0.5866±0.4111	0.1180±0.2105	0.7073±0.4378	0.3911±0.3334
DMF_M	0.8595±0.2845	0.9960±0.0446	0.7089±0.4116	0.9673±0.1592	0.6782±0.3893	0.9993±0.0211	0.8682±0.2184
DMF_R	0.8651±0.2823	0.9960±0.0446	0.7240±0.4075	0.9671±0.1593	0.6771±0.3893	0.9993±0.0211	0.8714±0.2174
RTLCD	0.6224±0.2215	0.4256±0.2661	0.7060±0.3372	0.7724±0.2739	0.3218±0.2682	0.4662±0.2570	0.5524±0.2707
LCDNN	0.6222±0.3296	0.9804±0.0839	0.4915±0.3671	0.8094±0.2871	0.6068±0.3380	0.7456±0.2549	0.7093±0.2768
ASFWF	**0.9245**±0.2128	**0.9974**±0.0438	**0.8767**±0.3003	**0.9698**±0.1554	**0.7708**±0.3814	**0.9996**±0.0173	**0.9231**±0.1582

TABLE VI
Fscore COMPARISON OF DIFFERENT ALGORITHMS ON REAL NETWORKS

Algorithms	Dolphins	Football	Political Books	Amazon	DBLP	Average
M-based	0.4685±0.2404	0.7146±0.3656	0.5257±0.3384	0.6938±0.2968	0.4131±0.3002	0.5631±0.3083
R-based	0.4505±0.2191	0.7101±0.3703	0.5030±0.3256	0.6566±0.2989	0.3956±0.2852	0.5432±0.2998
DMF_M	0.7255±0.2569	0.8881±0.2358	0.7295±0.3009	0.7862±0.2357	0.4205±0.2896	0.7100±0.2638
DMF_R	0.7242±0.2691	**0.8893±0.2388**	0.7286±0.3008	0.7825±0.2409	0.4223±0.2904	0.7094±0.2680
RTLCD	0.7376±0.2152	0.6639±0.2558	0.7888±0.2623	0.7766±0.2506	0.1628±0.2856	0.6259±0.2539
LCDNN	0.7558±0.1935	0.8498±0.2138	0.7463±0.3033	0.7978±0.2295	0.3700±0.2933	0.7039±0.2467
ASFWF	**0.9285**±0.1616	0.8812±0.2364	**0.7972**±0.2444	**0.8163**±0.2139	**0.4232**±0.3033	**0.7699**±0.2319



(a) (b) (c) (d)

Fig. 2. (a) Real local community. Local communities detected by (b) DMF_M, (c) LCDNN, and (d) ASFWF on Dolphins network (the seed is node 2).

division of the Dolphins network is shown in Fig. 2(a). The green node 2 is the seed node, and these yellow nodes are local community members. The detection results of DMF_M and LCDNN are shown in Fig. 2(b) and (c). Fig. 2(d) is the detection result of our algorithm. The result of RTLCD is only one more node labeled 36 than the result of LCDNN, so the result of RTLCD is not given.

From Fig. 2, it can be seen that the proposed algorithm achieves the best results. DMF_M is difficult to find the complete community of this seed node. LCDNN and RTLCD only find most members in the communities. The proposed ASFWF achieves the best result because it can continuously integrate influential nodes during the expansion of the community, such as the node 45.

The *fscore* of the local community detected by ASFWF for this seed is 0.9767. Table VII shows the number of seed nodes that exceed two different levels of *fscore* on this network.

It can be seen from Table VII that for most nodes in the network, the detection results of ASFWF exceed 0.9700.

TABLE VII
NUMBER OF SEED NODES THAT EXCEED TWO LEVELS OF *fscore*

fscore	>0.9700	>0.5200
DMF_M	19	51
RTLCD	18	59
LCDNN	20	62
ASFWF	52	61

However, other algorithms can only achieve this result on a small number of nodes.

G. Wilcoxon Signed Rank Test

In order to further investigate the effectiveness of the proposed algorithm, the results obtained by ASFWF and the comparison algorithm are statistically analyzed. The specific method is Wilcoxon signed rank test. The Wilcoxon signed rank test results of this algorithm and the comparison algorithm are shown in Table VIII. p represents the probability that the median values of two samples are equal. When p is close to 0, the null hypothesis should be questioned. h is the test result. $h = 0$ means that the median difference between the two samples is not significant, while $h = 1$ means that the median difference between the two samples is significant. It can be seen from Tables V and VI, ASFWF achieves the highest *fscore* on most networks and the highest average *fscore* of all networks. The results in Table VIII show that all comparison results are significantly improved. This further illustrates that the ASFWF algorithm has stable results and a significant improvement compared with the six comparison algorithms. This is because in the process of community fusion, the ASFWF deeply excavates the first two layers of neighbor information of nodes, which improves the algorithm's stability.

H. Running Time

The ASFWF algorithm and six comparison algorithms are run on 11 datasets for ten times, and the average run time is shown in Table IX. As can be seen from Table IX, the *M*-based

TABLE VIII
WILCOXON SIGNED RANK TEST RESULTS OF ASFWF ALGORITHM AND COMPARISON ALGORITHMS

Networks	M-based/ASWFS		R-based/ASWFS		DMF_M/ASWFS		DMF_R/ASWFS		RTLCD/ASWFS		LCDNN/ASWFS	
	p	h	p	h	p	h	p	h	p	h	p	h
LFR1	0	1	0	1	0	1	0	1	0	1	0	1
LFR2	0	1	0	1	0	1	0	1	0	1	0	1
LFR3	0	1	0	1	0	1	0	1	0	1	0	1
LFR4	0	1	0	1	0	1	0	1	0	1	0	1
LFR5	0	1	0	1	0	1	0	1	0	1	0	1
LFR6	0	1	0	1	0	1	0	1	0	1	0	1
Dolphins	0	1	0	1	0	1	0	1	0	1	0	1
Football	0	1	0	1	0	1	0	1	0	1	0	1
Political Books	0	1	0	1	0	1	0	1	0	1	0	1
Amazon	0	1	0	1	0	1	0	1	0	1	0	1
DBLP	0	1	0	1	0	1	0	1	0	1	0	1

TABLE IX
RUNNING TIME OF ASFWF ALGORITHM AND COMPARISON ALGORITHM (S)

Networks	M-based	R-based	DMF_M	DMF_R	RTLCD	LCDNN	ASFWF
LFR1	**0.0648**	0.5774	1.5568	7.1042	0.3661	0.1656	2.5646
LFR2	8.6538	241.54	234.34	848.8	156.11	**6.3192**	273.6
LFR3	**1.2506**	18.347	313.9	5183.5	23.864	11.1593	608.59
LFR4	**46.374**	906.82	610.04	3609.8	192.29	142.04	747.94
LFR5	**52.776**	439.39	23743	346806	1993.6	552.19	13996
LFR6	**305.58**	7531.4	1782.9	9110.3	1666.5	539.41	1753.9
Dolphins	0.1306	0.4478	0.2962	1.0572	0.0838	**0.0413**	0.3008
Football	0.4598	2.7316	0.8936	2.6179	0.3772	**0.1051**	0.8038
Political Books	0.4009	3.5803	4.8167	12.636	0.2125	**0.1179**	4.4839
Amazon	293.02	352.85	426.12	542.44	11.191	**9.2655**	39.168
DBLP	3977615	459183	428175	1192830	465.61	**427.14**	372713

algorithm and LCDNN algorithm have obvious advantages in running time. The LCDNN algorithm expands the community by calculating the NGC node of each neighbor point and propagating the label information of the NGC node already in the local community, so the computational complexity is low. DMF_M and DMF_R algorithms carry out three-stage local community detection, so the algorithm complexity is high. Although ASFWF's running time is not the fastest, it demonstrates large advantages in accuracy, while the running time is of comparable magnitude, and competitive with other algorithms in many cases. The ASFWF algorithm is significantly faster than the DMF_R algorithm. It also can be seen that, with the increase of data size, the growth rate of the running time of the ASFWF algorithm decreases compared with other algorithms.

I. Analysis of the Results

From the results given in Sections III-D and III-E, it can be seen that, compared with the six comparison algorithms, the results of the proposed algorithm on 11 datasets show significant advantages. Section III-F, the single point experiment shows that for a single node in dolphin network, the ASFWF algorithm demonstrates stronger local community discovery ability and can find the overall community structure of the node more completely. The test method used in this article is to test all nodes as seed points and calculate the average accuracy. The M-based method and R-based method are two greedy local community detection algorithms. They maximize local modularity M and R, so they are very sensitive to the selection of seed nodes. The membership function used by the DMF_M and DMF_R algorithms mainly defines the community quality from the point of view of node edge, and does not pay enough attention to the connection information within the local community. The RTLCD algorithm depends on the core node label information of neighbor nodes. The LCDNN algorithm is based on NGC nodes and has high dependence on node centrality. The ASFWF algorithm uses two fusion strategies alternately and iteratively, so that the algorithm can integrate different types of nodes in each stage. The membership function used in the ASFWF algorithm reflects the neighbor information of the node and the connection information in the local community obtained in the detection process.

The statistical experiments in Section III-G show that the local community discovery results obtained by the ASFWF algorithm are relatively stable. In the process of community fusion, the ASFWF algorithm deeply excavates the first two layers of neighbor information of nodes, makes the membership value between two nodes as unique as possible, and improves the algorithm's stability.

The running time experimental results of Section III-H show that M-based and LCDNN algorithms have obvious advantages in running time. First, the M-based method and R-based method have low overall complexity by maximizing local modularity M and R. However, because the modularity R divides the local community into two parts: 1) boundary and 2) core. The M-based algorithm only calculates the total number of edges in the local community to the number of

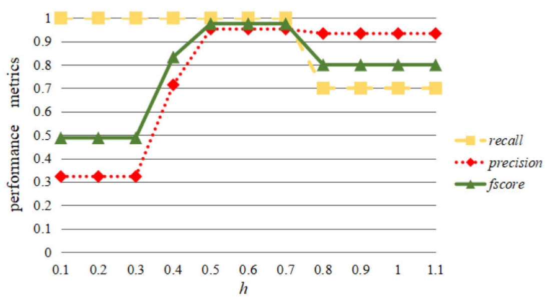


Fig. 3. Local community detection results with node 9 as the seed point under different h values.

edges outside the community. So, the algorithm's computational complexity based on function M is lower than that based on function R. DMF_M and DMF_R algorithms carry out three-stage local community detection, and the algorithm complexity is high. RTLCD algorithm is a two-stage local community detection method, which has no advantage in running time. LCDNN algorithm expands the community by calculating the NGC node of each neighbor point and propagating the label information of the NGC node already in the local community, so the computational complexity is low.

To sum up, the ASFWF algorithm is not always the fastest of the compared algorithms due to the alternating cycle's strong and weak fusion mode but it also makes the ASFWF algorithm more accurate. For different nodes in the network, the algorithm is more stable, less affected by the selection of seed points, and can better find the complete community where the seed points are located, which is of more practical significance.

IV. Analysis of Parameters and Fusion Iteration Mode

The analysis of parameters and fusion iteration mode will be given in this section.

A. Experiments on Parameters

The discussion of parameter h and the effectiveness of evaluation metrics Mc are given in this section.

1) Discussion of Parameter h: The Dolphins network is used as the test network, with node 9 as the seed, and the *recall*, *precision*, and *fscore* of the local community obtained by ASFWF are calculated at all h values. Fig. 3 shows the detection results of all h values starting from one seed. When h is between 0.1 and 0.4, it means that ASFWF is allowed to integrate nodes with lower membership of local community, which increases the probability of the algorithm integrating into the wrong nodes.

From Fig. 3, when h is less than 0.5, the lower precision indicates that the algorithm expands a large number of error nodes for the local community as h decreases. This is consistent with our theoretical statement. So, it is not the case that the lower the value of h, the better the effect of ASFWF. Sometimes it will expand the local community excessively. For example, on the Football network, ASFWF obtains the best result when h is 1.1. Actually, the value of h is set between 0.5 and 1.1, with an interval of 0.1.

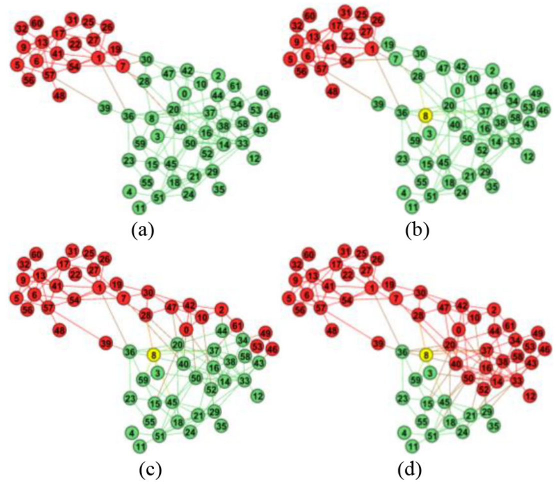


Fig. 4. Local community detected by ASFWF under different h. (a) Real community. Local community when h is (b) 0.5, (c) 0.7, and (d) 1.1.

In order to show how the parameter h affects the algorithm intuitively. Fig. 4 plots the local communities of node 8 of Dolphins network obtained by ASFWF under different h values.

In Fig. 4, the yellow point is the seed node, the green points represent the local community nodes, and the red points are nodes outside the local community. When h is 1.1, it means that only strong fusion is used in ASFWF. Fig. 4(d) shows that ASFWF with only strong fusion cannot expand the local community well. When h varied from 0.6 to 1, weak fusion plays a role. As shown in Fig. 4(c), by using weak fusion, influential nodes, such as nodes 14, 20, and 40 can be found, thereby increasing the width of community detection. When h is 0.5, ASFWF obtains the best effect at this seed point. This means that weak fusion does have a great effect on improving the algorithm results, and the strength of weak fusion can be controlled by adjusting h. Therefore, the parameter-based community weak fusion strategy has more advantages for networks of different sizes and structures. A single parameter value will not accurately divide the community structure for networks with different structures. Therefore, the selection of parameter h value in this article relies on a new evaluation index M_C. Mc will evaluate and select the local community detection results corresponding to all h values on different datasets. When the real community structure information of the network cannot be obtained, the community detection result corresponding to the parameter h value selected by the evaluation index M_C will be more accurate.

2) Effectiveness of Evaluation Metrics M_C: To prove the validity of the evaluation index M_C, 6 synthetic networks and five real networks are tested. For h from 0.5 to 1.1, we calculate the *fscore* and M_C of each node as seed, and then compute the average *fscore* and the average M_C of all seeds. Clearly, suppose M_C's evaluation of the local community is consistent with that of *fscore*. In that case M_C will be effective and can be used to evaluate and choose the local community detected

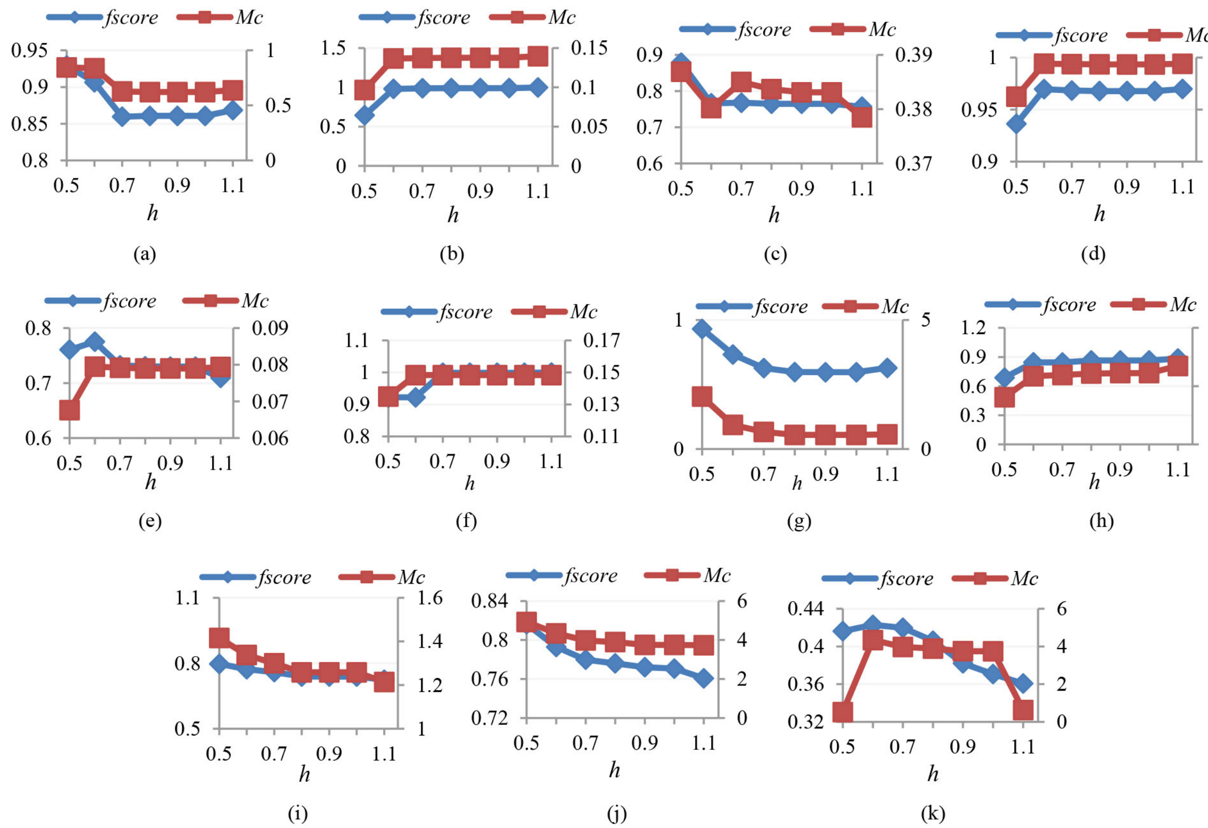


Fig. 5. Evaluation of local community detected by ASFWF under different h values. (a) LFR1 network. (b) LFR2 network. (c) LFR3 network. (d) LFR4 network. (e) LFR5 network. (f) LFR6 network. (g) Dolphin network. (h) Football network. (i) Polbooks network. (j) Amazon network. (k) DBLP network.

by ASFWF under different h values. The experimental results are shown in Fig. 5.

It can be seen from Fig. 5 that the evaluation levels of M_C and *fscore* on the two test networks are almost the same. M_C may identify errors for results with small gaps, but it can help select the best results with large gaps. For different h, the best results are much higher than other results. So, M_C is very effective for ASFWF to choose the best result and the right h. When the ASFWF performs local community detection, the parameter h that can make M_C reach the maximum value is the best parameter setting on the network.

As can be seen from Fig. 5, the trends of M_C and *fscore* with h value on all datasets are similar. On the Dolphin, Polbooks, Amazon, LFR1, and LFR3 networks, the corresponding h value of the highest *fscore* value and the highest M_C value is 0.5. This shows that strong integration alone is not able to obtain more accurate community division results on these five networks. After adding high strength weak fusion, the local community is better able to overcome the optimal local solution and find more influential nodes to integrate into the community. On the DBLP and LFR5 networks, the corresponding h value of the highest *fscore* value and the highest M_C value is 0.6. When the weak fusion is slightly weakened, more accurate community detection results can be obtained. On all datasets, the change trends of M_C and *fscore* are consistent, indicating that M_C is very effective in selecting the best parameters and results of ASFWF. When ASFWF performs local detection, the parameter h that can make M_C reach the maximum value is the best parameter setting on the network. The more consistent the change of M_C value with the change of *fscore*, the more effective M_C is. In conclusion, the single intensity fusion iteration cannot obtain the optimal detection results for networks with different sizes and structures. Therefore, by adjusting the value of parameter h, it is more advantageous to use M_C to judge and select the local community structure of the network that cannot be truly divided.

B. Discussion on the Fusion Iteration Mode

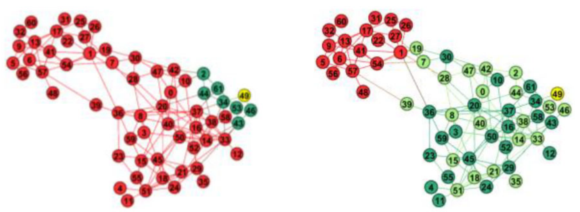
There are two strategies to fuse nodes in ASFWF. In order to prove that alternate use of the two methods can play a more effective role of the two strategies, another algorithm TSLCD is first defined.

TSLCD is designed as a two-step local community detection algorithm with strong fusion and weak fusion. First, strong fusion is performed on the seed points until no neighboring nodes meet the requirement. Second, the local community is expanded with weak fusion until no neighbors can be merged. Table X shows the detection results of the two algorithms on the Dolphins network, where *recall*, *precision*, and *fscore* are the average results of all seed points and M_C is the sum of the results of all seed points.

Table X shows that both algorithms achieve optimal results when h is 0.5, and the evaluation index M_C also correctly evaluates the optimal local community. Comparing the optimal

TABLE X
COMPARISON RESULTS OF ASFWF AND TSLCD ON DOLPHINS NETWORK

Algorithms	h	0.5	0.6	0.7	0.8	0.9	1.0	1.1
TSLCD	*recall*	0.7491	0.5957	0.5067	0.4962	0.4962	0.4962	0.4962
	precision	0.9386	0.9727	0.9702	0.9694	0.9694	0.9694	0.9694
	fscore	**0.7905**	0.7003	0.6360	0.6291	0.6291	0.6291	0.6291
	M_C	**79.1784**	35.8023	32.7482	35.3518	35.3518	35.3518	35.3518
ASFWF	*recall*	0.9428	0.6514	0.5104	0.4584	0.4584	0.4584	0.4962
	precision	0.9363	0.9700	0.9745	0.9716	0.9716	0.9716	0.9694
	fscore	**0.9285**	0.7333	0.6275	0.5966	0.5958	0.5958	0.6291
	M_C	**124.9275**	58.0385	41.2146	33.9178	33.9117	33.9117	35.3518



(a) (b)

Fig. 6. Local community detected by (a) TSLCD and (b) ASFWF.

TABLE XI
TSLCD AND ASFWF LOCAL COMMUNITY NODE FUSION ORDER

TSLCD	49, 34, 46, 44, 2, 61, 53, 43
ASFWF	49, 34, 14, 37, 33, 16, 38, 50, 40, 43, 0, 20, 21, 52, 18, 45, 51, 29, 15, 24, 8, 59, 42, 23, 47, 10, 28, 30, 2, 36, 44, 3, 19, 12, 7, 55, 39, 4, 46, 11, 53, 61, 35, 58

results of the two algorithms, their precision values are very close, but ASFWF can better expand the local community, greatly improve *recall,* and obtain better results in overall performance. It is because the mechanism of ASFWF makes the two strategies play a role in various stages of community expansion.

Then, a node on the Dolphins network is used as a seed to demonstrate the advantages of ASFWF's alternate use of the two strategies. The local communities in Fig. 6 are obtained by TSLCD and ASFWF. The yellow points indicate the seed, the dark green nodes are fused by strong fusion, and the light green points are fused by weak fusion.

From Fig. 6, it can be seen that after TSLCD runs strong fusion, no nodes can meet the requirements of weak fusion. So, it only obtains a small local community. However, due to the alternative use of these two strategies, ASFWF enables them to play a role in every stage of the algorithm.

Table XI shows the sequence of the two community expansion nodes in Fig. 6.

From Table XI, the influential node 14 is integrated at the first operation of weak fusion in ASFWF, which greatly improves the scope of community expansion. Therefore, it is very important to find influential nodes through weak fusion in the early stage of community expansion. By comparing method TSLCD, the results are shown in Table VIII and Fig. 6. It is proven that ASFWF alternately uses two fusion strategies can make them play a better role.

In conclusion, in Section III, six methods on six synthetic networks and six real networks are compared. Among them, a better *fscore* in most networks is achieved by the proposed method, which means ASFWF is more effective in local community detection. ASFWF mainly improves results by expanding the width of community detection. Furthermore, M_C can help our method choose the appropriate parameters. Finally, alternately using two fusion strategies can make these two strategies play a better role is proved.

V. CONCLUSION

A local community detection algorithm ASFWF was proposed in this article, which uses two fusion strategies alternately for local community detection. The difference between the two fusion strategies is called strong fusion and weak fusion, respectively. The strong fusion can increase the modularity of the local community and make the local community expand to the direction of close internal connection and sparse external connection. In the strong fusion, a new membership function is designed, which makes full use of the node information and the internal topology information of the current local community. Weak fusion is a parameter-based fusion method. Only nodes with membership values greater than the parameter can join the local community. These two fusion strategies will fuse nodes of different natures due to their different qualities. The strong fusion method is used to fuse the nodes with a small degree, while the accuracy for the local community is high. The weak fusion method can help local communities integrate with influential nodes, thereby expanding the width of local community detection. However, an appropriate parameter h needs to be set. In order to prove the effectiveness of ASFWF, comparative experiments on both synthetic networks and real networks were performed. The results demonstrated that ASFWF can increase the detection width of the algorithm to make its overall performance better than previous algorithms. Results also showed that the alternative use of the two fusion strategies can effectively exploit the strengths of both strategies to improve the overall performance of the algorithm. However, this article has only studied unweighted and undirected networks. The weighted network, directed network, and attribute network are not considered in this article. Future work will further improve and adjust the situation of the weighted network, directed network and attribute network, to make the algorithm generalizable to a wider range of applications.

ACKNOWLEDGMENT

The authors would like to express their sincere appreciation to the editors and the anonymous reviewers for their insightful comments, which have greatly helped them in improving the quality of this article.

REFERENCES

[1] M. Ghazanfari, "An overlapping community detection algorithm based on local community and information flow expansion (LCFE) in weighted directed networks," *Decis. Sci. Lett.*, vol. 9, no. 4, pp. 547–558, 2020.

[2] X. Teng, J. Liu, and M. Li, "Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 138–150, Jan. 2021.

[3] R. Shang, W. Zhang, J. Zhang, J. Feng, and L. Jiao, "Local community detection based on higher-order structure and edge information," *Phys. A, Stat. Mech. Appl.*, vol. 587, Feb. 2022, Art. no. 126513.

[4] R. Shang, H. Liu, L. Jiao, and A. M. G. Esfahani, "Community mining using three closely joint techniques based on community mutual membership and refinement strategy," *Appl. Soft Comput.*, vol. 61, pp. 1060–1073, Dec. 2017.

[5] C. Liu, J. Liu, and Z. Jiang, "A multi-objective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2274–2287, Dec. 2014.

[6] A. N. Papadopoulos and G. Tzortzidis, "Distributed time-based local community detection," in *Proc. 24th Pan-Hellenic Conf. Informat.*, 2020, pp. 390–393.

[7] M. A. Midoun, X. Wang, and M. Z. Talhaoui, "A jungle community detection algorithm based on new weighted similarity," *Arabian J. Sci. Eng.*, vol. 46, pp. 8493–8507, Mar. 2021.

[8] R. Shang, W. Zhang, L. Jiao, X. Zhang, and R. Stolkin, "Dynamic immunization node model for complex networks based on community structure and threshold," *IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1539–1552, Mar. 2022, doi: 10.1109/TCYB.2020.2989427.

[9] X. You, Y. Ma, and Z. Liu, "A three-stage algorithm on community detection in social networks," *Knowl. based Syst.*, vol. 187, Jan. 2020, Art. no. 104822.

[10] Z. Bu, Z. Wu, J. Cao, and Y. Jiang, "Local community mining on distributed and dynamic networks from a multiagent perspective," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 986–999, Apr. 2016.

[11] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2703–2716, Sep. 2017.

[12] X. Zhang, K. Zhou, H. Pan, L. Zhang, X. Zeng, and Y. Jin, "A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 703–716, Feb. 2020.

[13] Y. Li, C. Jia, X. Kong, L. Yang, and J. Yu, "Locally weighted fusion of structural and attribute information in graph clustering," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 247–260, Jan. 2019.

[14] Y. Gao, X. Yu, and H. Zhang, "Overlapping community detection by constrained personalized PageRank," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114682.

[15] C. Lyu, Y. Shi, and L. Sun, "A novel local community detection method using evolutionary computation," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3348–3360, Jun. 2021, doi: 10.1109/TCYB.2019.2933041.

[16] K. Guo, L. He, Y. Chen, W. Guo, and J. Zheng, "A local community detection algorithm based on internal force between nodes," *Appl. Intell.*, vol. 50, no. 2, pp. 328–340, 2020.

[17] D. He, Y. Wang, J. Cao, W. Ding, S. Chen, and Z. Feng, "A network embedding-enhanced Bayesian model for generalized community detection in complex networks," *Inf. Sci.*, vol. 575, pp. 306–322, Oct. 2021.

[18] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.

[19] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, 2006, Art. no. 36104.

[20] Y. Guo, H. Yang, M. Chen, J. Cheng, and G. Dong, "Ensemble prediction-based dynamic robust multi-objective optimization methods," *Swarm Evol. Comput.*, vol. 48, pp. 156–171, Aug. 2019.

[21] M. A. Tabarzad and A. Hamzeh, "A heuristic local community detection method (HLCD)," *Appl. Intell.*, vol. 46, no. 1, pp. 62–78, 2017.

[22] F. Luo, J. Z. Wang, and E. Promislow, "Exploring local community structures in large networks," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2006, pp. 233–239.

[23] J. Chen, O. Zaïane, and R. Goebel, "Local community identification in social networks," in *Proc. Int. Conf. Adv. Social Netw. Anal. Min.*, 2009, pp. 237–242.

[24] R. Shang, W. Zhang, L. Jiao, R. Stolkin, and Y. Xue, "A community integration strategy based on an improved modularity density increment for large-scale networks," *Phys. A, Stat. Mech. Appl.*, vol. 469, pp. 471–485, Mar. 2017.

[25] Q. Chen and T. T. Wu, "A method for local community detection by finding maximal-degree nodes," in *Proc. IEEE Int. Conf. Mach. Learn. Cybern.*, vol. 1, 2010, pp. 8–13.

[26] F. Jiang, S. Y. Jin, Y. L. Wu, and J. Xu, "A uniform framework for community detection via influence maximization in social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Min.*, 2014, pp. 27–32.

[27] Q. Chen, T.-T. Wu, and F. Ming, "Detecting local community structures in complex networks based on local degree central nodes," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 3, pp. 529–537, 2013.

[28] X. Ding, J. Zhang, and J. Yang, "A robust two-stage algorithm for local community detection," *Knowl. Based Syst.*, vol. 152, pp. 188–199, Jul. 2018.

[29] W. Luo, N. Lu, L. Ni, W. Zhu, and W. Ding, "Local community detection by the nearest nodes with greater centrality," *Inf. Sci.*, vol. 517, pp. 377–392, May 2020.

[30] W. Luo, D. Zhang, H. Jiang, L. Ni, and Y. Hu, "Local community detection with the dynamic membership function," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3136–3150, Oct. 2018.

[31] P. Liakos, A. Ntoulas, and A. Delis, "Uncovering local hierarchical overlapping communities at scale," *IEEE Trans. Big Data*, vol. 8, no. 2, pp. 432–445, Apr. 2022.

[32] R. Shang, H. Liu, and L. Jiao, "Multi-objective clustering technique based on K-nodes update policy and similarity matrix for mining communities in social networks," *Phys. A, Stat. Mech. Appl.*, vol. 486, no. 15, pp. 1–24, 2017.

[33] D. Lusseau, "The emergent properties of a dolphin social network," *Proc. Royal Soc. B*, vol. 270, no. S2, pp. 186–188, 2003.

[34] V. Krebs. "Political Books Network." 2004. [Online]. Available: http://www.orgnet.com

[35] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowl. Inf. Syst.*, vol. 12, pp. 181–213, Jan. 2015.

Ronghua Shang (Member, IEEE) received the B.S. degree in information and computation science and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2003 and 2008, respectively.

She is currently a Professor with Xidian University. Her current research interests include optimization problems, evolutionary computation, image processing, and data mining.

Weitong Zhang (Member, IEEE) received the B.E. degree in electronic and information engineering from the Changchun University of Science and Technology, Changchun, China, in 2013, and the M.S. degree in electronics and communication engineering and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2021.

She is currently a Lecturer with Xidian University. Her current research interests include complex networks and machine learning.

Jingwen Zhang received the B.E. degree from the School of Information Science and Engineering, Henan University of Technology, Zhengzhou, China, in 2018. She is currently pursuing the M.S. degree with the School of Artificial Intelligence, Xidian University, Xi'an, China.

Her current research interests include complex networks analysis and data mining.

Licheng Jiao (Fellow, IEEE) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

From 1990 to 1991, he was a Postdoctoral Fellow with the National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, where he has been a Professor with the School of Electronic Engineering since 1992, and is currently the Director of the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education of China. His research interests include image processing, natural computation, machine learning, and intelligent information processing.

Dr. Jiao is a Member of IEEE Xian Section Executive Committee, the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, the Councilor of the Chinese Institute of Electronics, the Committee Member of the Chinese Committee of Neural Networks, and an Expert of the Academic Degrees Committee of the State Council.

Yangyang Li (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and technology and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2001, 2004, and 2007, respectively.

She is currently a Professor with the School of Artificial Intelligence, Xidian University. Her research interests include quantumin-inspired evolutionary computation, artificial immune systems, and data mining.

Rustam Stolkin received the M.Eng. degree in engineering science from the University of Oxford, Oxford, U.K. in 1998, and the Ph.D. degree in computer vision from the University College London, London, U.K., in 2004.

He is the Chair of Robotics with the University of Birmingham, Birmingham, U.K.; the Royal Society Industry Fellow; the Chair of the Expert Group on Robotic and Remote Systems for the OECD's Nuclear Energy Agency; and founded the U.K. National Centre for Nuclear Robotics in 2017. His research interests include computer vision and image processing, machine learning and AI, robotic grasping and manipulation, and human–robot interaction.