Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 406485, 10 pages
http://dx.doi.org/10.1155/2014/406485

Research Article

An Improved Local Community Detection Algorithm Using Selection Probability

Shixiong Xia, Ranran Zhou, Yong Zhou, and Mu Zhu

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

Correspondence should be addressed to Ranran Zhou; zhouranrann@163.com

Received 27 September 2013; Revised 18 December 2013; Accepted 19 December 2013; Published 12 January 2014

Academic Editor: Yuri Vladimirovich Mikhlin

In order to find the structure of local community more effectively, we propose an improved local community detection algorithm ILCDSP, which improves the node selection strategy, and sets selection probability value for every candidate node. ILCDSP assigns nodes with different selection probability values, which are equal to the degree of the nodes to be chosen. By this kind of strategy, the proposed algorithm can detect the local communities effectively, since it can ensure the best search direction and avoid the local optimal solution. Various experimental results on both synthetic and real networks demonstrate that the quality of the local communities detected by our algorithm is significantly superior to the state-of-the-art methods.

1. Introduction

In the real world, many complex systems can be described through various kinds of networks, such as interpersonal networks, biological networks, neural networks, social networks, and WWW. Commonly, in these networks, individuals represented by nodes are linked with some special relationships. A large amount of studies reveal that there exist underlying communities in most complex networks. Community detection, as a key technology for network analysis, can discover the hidden structures and functions in complex networks, which is attracting a considerable amount of attention from researchers in various domains.

In recent years, a large number of community detection algorithms have been proposed, such as module optimization algorithm [1–3], spectral clustering algorithm [4, 5], hierarchical clustering algorithm [6, 7], and label propagation algorithm [8, 9]. Most of them focus on finding the global community structure which contains all the possible communities. However, in real networks, such as WWW, they are large-scale and even dynamic. The information of the whole network is very difficult or even impossible to be obtained, so none of the algorithms is suitable to detect the community structures of large-scale networks with incomplete information. Because of these restrictions, the problem of local community detection has drawn the scholars' attention increasingly.

Some local community detection algorithms have been proposed based on the limited information, such as Clauset [10], LWP [11], and LS [12]. These algorithms start from a source node and then gradually join nodes to the local community. Eventually they will find the local community with the source node. Usually the nodes in a community can be divided into core nodes and edge nodes. Core node is the node which connects closely with the interior nodes of the community it is located in. Edge node is the node which connects sparsely with the interior nodes of the community it is located in. The effect of an algorithm depends on the choice of the source node. If the core node is the source node, the algorithm usually can get better structure of the local community. Most of the local community detection algorithms adopt strategy of climbing optimization, which will make the algorithm fall into local optimum, since they join the node with the optimal value of the objective function at each step. As shown in Figure 1, the source node is the edge node i of community C. Because the node i has closer connection with community D, the next step may join node j of community D to the local community. Then the following steps may continue to gather nodes of community D. Finally a local community composed of community D and node i of

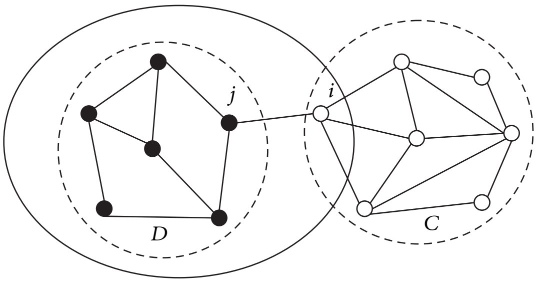


Figure 1: Problem of previous algorithms.

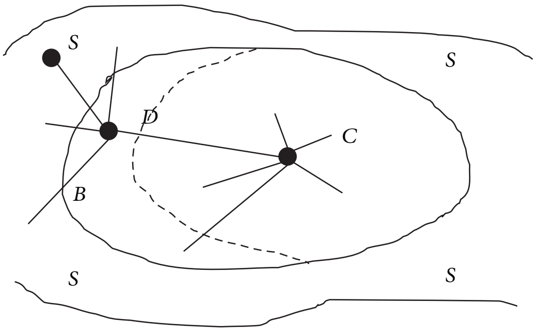


Figure 2: Definition of local community.

community C could be obtained. So the community C where the node i really locates could not be obtained.

To solve the above problem, we propose an improved local community detection algorithm using selection probability (ILCDSP). The main idea of the algorithm is to set selection probability for the candidate nodes at each step, making the nodes with high selection probability more probably be chosen. ILCDSP screens nodes randomly; it does not just only select the node with high modularity, and thus it leads the algorithm process to the best search direction. It can avoid such a vicious circle that one misstep will undo all our work. The performance of the algorithm is verified, respectively, on the real and simulated data sets. The evaluation method used is Precision, Recall, and F-Score. The experimental results show that this algorithm, compared with Clauset and LWP, could find a local community with higher quality.

The paper is organized as follows. Section 2 introduces concepts and algorithms related to local community detection; Section 3 introduces the algorithm ILCDSP proposed in this paper; Section 4 verifies the performance of the algorithm ILCDSP through the experiment on the real and simulated networks; and Section 5 gives the conclusion and discusses the future work which will be done.

2. Related Work

2.1. Problem Definition. The definition of local community detection was firstly proposed by Clauset [10]. Usually the following method is used to define the problem: there is an undirected graph $G = (V, E)$; V represents the node set of the graph and E represents the edge set of the graph. The connectivity information of part of the nodes in the graph is known or available. As shown in Figure 2, the known local community is defined as area D, and the set of nodes connected with nodes in D is defined as area S; the set of nodes in D which is connected with the nodes in S is defined as edge nodes area B, where any node in B has at least one neighbor node in S. The rest of D is the core nodes area C.

The task of local community detection is to constitute a local community D from a source node. During the process, the algorithms should merge the node which meets the conditions into D and remove the node in D which does not meet the conditions. Different algorithms have different conditions to select node, and these will be shown in the next section.

2.2. Related Algorithms. We have reviewed several effective approaches to explore local community structure. These algorithms are presented below in the sequence of publication date.

2.2.1. Clauset Algorithm. In order to solve the problem of local community detection, Clauset [10] defined local community modularity R and proposed a greedy algorithm of fast convergence to find the local community with maximum modularity.

Local community modularity R is defined as follows:

$$R = \frac{\sum_{ij} B_{ij}\delta(i, j)}{\sum_{ij} B_{ij}}, \tag{1}$$

where i and j represent node i and node j, and

$$B_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise,} \end{cases}$$
$$\delta(i, j) = \begin{cases} 1 & \text{both nodes } i \text{ and } j \text{ in } D \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The process of Clauset algorithm is similar to web crawler. First, it starts from a source node v, merges node v into the subgraph D, merges all neighbor nodes of node v into S, and then iterates the following steps.

Step 1. Calculate ΔR for all the nodes in S. ΔR is the increment of modularity R.

Step 2. Merge the node j with the largest modularity incremental ΔR into D.

Step 3. Merge all the neighbor nodes of node j into S and then update R and B.

The algorithm merges the neighbor nodes, which will be able to bring the biggest increment of R into the local community one by one until the size of the local community reaches the predetermined value. That is to say, this algorithm needs to set a parameter to decide the size of the local community in advance. The clustering result is highly affected by the source node.

2.2.2. LWP Algorithm. LWP [11] algorithm is an improved algorithm. Compared with Clauset, it has definite termination conditions. The algorithm defines the community modularity M as follows:

$$M = \frac{(1/2)\sum_{ij} A_{ij}\theta(i,j)}{\sum_{ij}\lambda(i,j)}, \tag{3}$$

where

$$A_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise,} \end{cases}$$
$$\theta(i,j) = \begin{cases} 1 & \text{both nodes } i \text{ and } j \text{ in } D \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$
$$\lambda(i,j) = \begin{cases} 1 & \text{only one of nodes } i \text{ or } j \text{ in } D \\ 0 & \text{otherwise.} \end{cases}$$

Given an undirected graph $G(V, E)$, LWP algorithm will start from a source node and then find a subgraph with maximum M. If the subgraph is a community ($M > 1$), the algorithm returns the community consisting of the subgraph. Otherwise, we hold the opinion that it cannot find any community starting from the source node. For one source node v, LWP algorithm finds the subgraph with maximum M through two main steps. Firstly, a subgraph which has only one source node v is constructed, and then the neighbor nodes of v are merged into S. The following is the incremental step and pruning step of the algorithm.

In the incremental step, the algorithm merges the nodes of S into D iteratively until D is extended to a certain degree. Each time the algorithm will select the node which can bring the biggest increment of M; in the pruning step, the iteration algorithm will remove the nodes of D which will make the local community modularity of D increase until no nodes can be removed. This algorithm turns out to result in high recall but low accuracy.

2.2.3. Bagrow Algorithm. Bagrow and Bollt proposed a local community detection algorithm [13]. The algorithm from a source node, according to the boundary, gradually merges node into the community; here the boundary is the set of nodes whose distance to the source node is a fixed value. The effect of the algorithm depends on the choice of the source node. If the source node is a boundary node rather than a core node, the final clustering results will be very different. In order to overcome this problem, the authors suggest making each node as a source node at a time, repeating the calculation to find the optimal result; however, the speed of the algorithm is very slow.

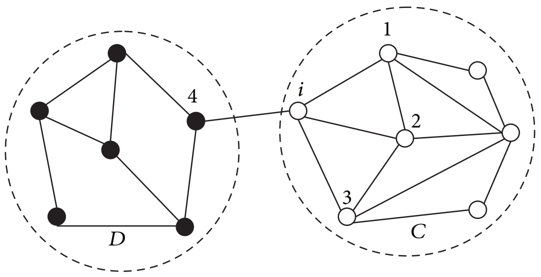


Figure 3: An example of application of ILCDSP.

3. Improved Local Community Detection Algorithm Using Selection Probability

Most of the existing algorithms utilize the greedy strategy to select the present optimal node to join the local community. It can easily make the algorithm traps in local optimal solution. In order to avoid the occurrence of this phenomenon, this paper on the basis of the local community modularity, will give another selection criteria of nodes—selection probability.

3.1. Selection Probability. LWP algorithm needs to select the subsequent node according to the largest increment of M. The nodes which make the increment of M greater than zero are regarded as candidate nodes. Here, we set the selection probability P for the candidate nodes according to ΔM which is the increment of M. Consider

$$p_i = \frac{\Delta M(i)}{\sum_{k=1}^{n}\Delta M(k)} \quad \text{s.t. } M(k) > 0, \tag{5}$$

where p_i is the selection probability of candidate node i, and n is the number of candidate nodes.

The selection probability is determined by ΔM. Usually the lager ΔM is, the lager the selection probability is; that is to say, the greater the increment of M is, the greater likelihood the node will be added to the local community.

3.2. Steps of the Algorithm. ILCDSP algorithm generates random number r between 0 and 1; if $P_1 + P_2 + \cdots + P_{i-1} < r < P_1 + P_2 + \cdots + P_i$, node i will be merged into community D. It is clear that the node with a greater selection probability will have more chance to be selected.

Steps of the algorithm are shown in Algorithm 1.

ILCDSP algorithm firstly merges the source node v_0 into the local community D and then merges all the neighbor nodes into S (Step 1). It initializes the community modularity as zero (Step 2), goes through each node in N, then chooses the nodes which make ΔM lager than zero as candidate nodes, and, at the same time, calculates their selection probability (Step 3). ILCDSP algorithm generates random number r between 0 and 1, then selects node v according to the node selection probability, and merges it into D. It updates S, B, and M and repeats the above steps until no node can be merged into D (Steps 4–9). Finally the algorithm will return the local community D containing the source node (Step 10).

Here, we give an example to illustrate the application of the algorithm, as show in Figure 3.

```
**Input**: A source node $v_0$ and a network
**Output**: A local community containing source node $v_0$
Merge the source node $v_0$ into $D$ and merge the
    neighbor nodes of $v_0$ into $S$
Set local modularity $M = 0$
Calculate $P_i$ for each node in $N$ which makes $\Delta M > 0$
**Do**
**While** $N$ is not empty **Do**
Generate $r \in [0, 1]$ randomly, find the candidate node $v$
Merge node $v$ into $D$, and remove it from $S$
Update $S, B, M$
**Until** no new node is merged into $D$
**Return** $D$
```

Algorithm 1: ILCDSP algorithm.

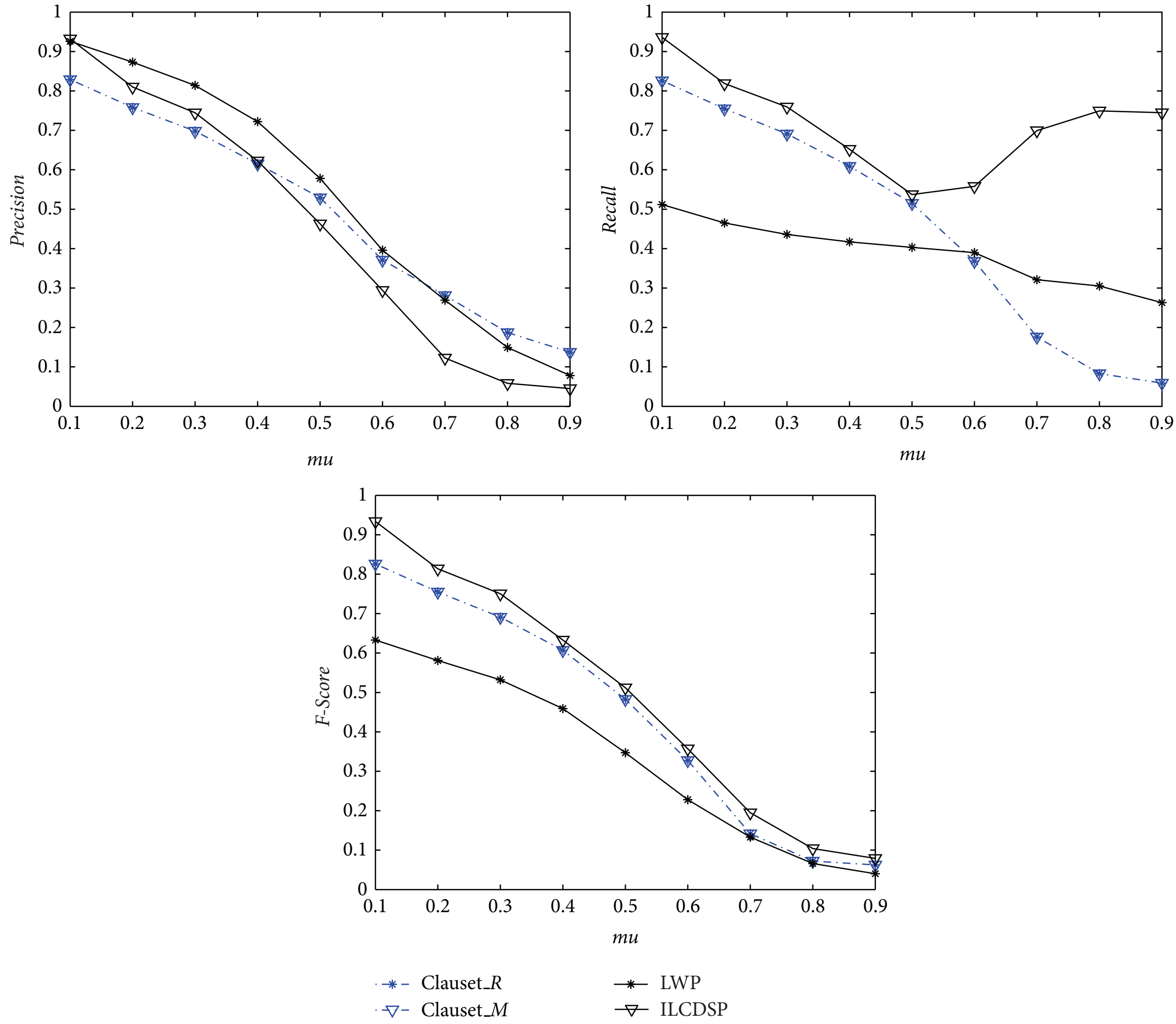


Figure 4: Comparison of B1.

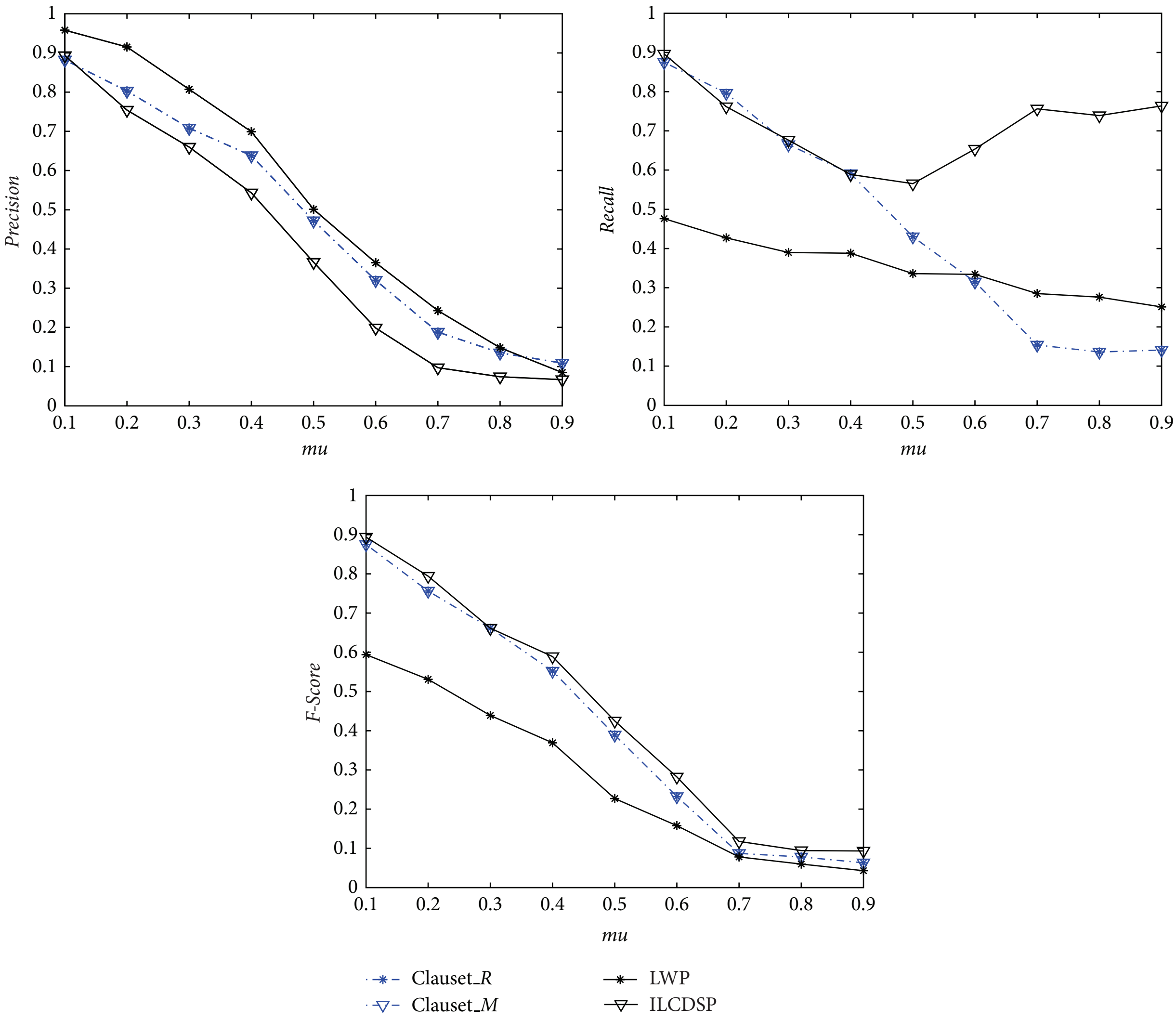


Figure 5: Comparison of B2.

The source node is the edge node i of community C. In order to find the local community C that node i locates, the next step should select one node from the candidate nodes 1, 2, 3, and 4 to be joined into the local community. Their increment of M is $\Delta M_1 = \Delta M_2 = \Delta M_3 = 1/6$, $\Delta M_4 = 1/5$. According to LWP, node 4 with the largest increment of M should be joined into the local community. It is clear that node 4 does not belong to community C. If node 4 is joined into the local community, the following steps may continue to gather nodes of community D. Finally, we will get a local community composed of community D and node i of community C. So we cannot get the community C where the node i locates. But in our algorithm, we calculate the selection probability for nodes 1, 2, 3, and 4. Considering $p_1 = p_2 = p_3 = 5/21$, $p_4 = 2/7$, one node will be randomly selected from them according to our algorithm. Although node 4 cannot avoid being selected completely, the probability of each candidate nodes has to be considered; our algorithm reduces the probability of the situation like node 4. It can get a better result.

In order to improve the efficiency of the algorithm, for any node $v_j \in S$, the formula (6) [12] can be used to speed up the calculation of ΔM_j,

$$\Delta M_j = \frac{xk_{\text{out}} + (x-y)\,k_{\text{in}}}{(k_{\text{out}} - x + y)\,k_{\text{out}}}, \qquad (6)$$

where k_{in} denotes the number of edges inside D and k_{out} denotes the number of edges between D and S. x represents the number of edges that will be added into D because of the agglomeration of node v_j, and y is the number of edges which connect v_j and nodes outside D. It is easy to find out that the degree of v_j equals to $x + y$.

The original calculation of ΔM is to use the M after joining node v_j minus the M before joining v_j. It equals to formula (6), which can be proved as follows.

Firstly,

$$M = \frac{(1/2)\sum_{ij} A_{ij}\theta(i,j)}{\sum_{ij}\lambda(i,j)} = \frac{k_{\text{in}}}{k_{\text{out}}}. \qquad (7)$$

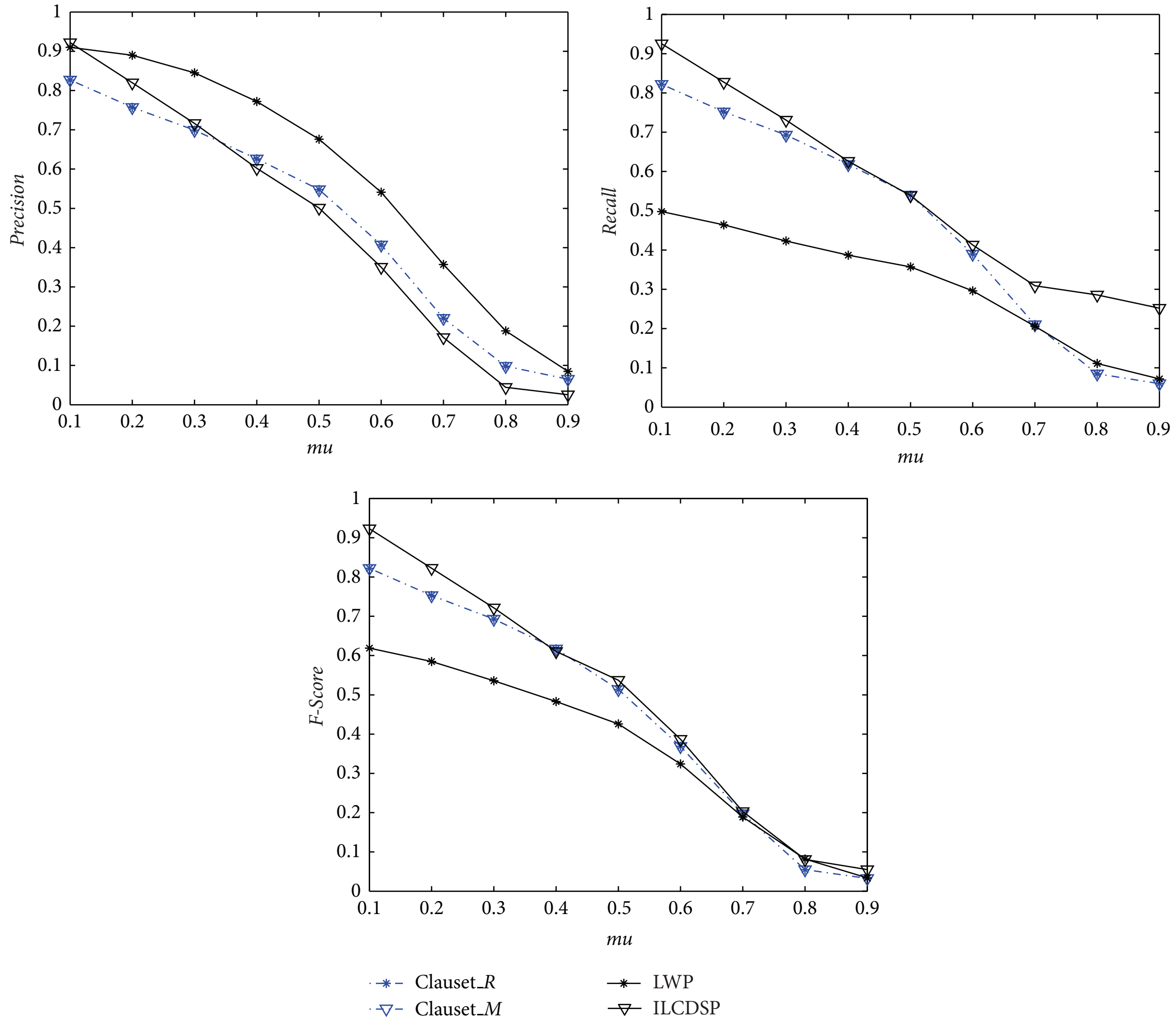


Figure 6: Comparison of B3.

And then

$$\Delta M_j = \frac{k_{\text{in}} + x}{k_{\text{out}} - x + y} - \frac{k_{\text{in}}}{k_{\text{out}}} = \frac{x k_{\text{out}} + (x - y) k_{\text{in}}}{(k_{\text{out}} - x + y) k_{\text{out}}}. \quad (8)$$

Compared to the original calculation of ΔM which needs to traverse all the nodes each time, the formula (6) only needs to know x and y of the node which will be joined into the local community.

4. Experiments

This section verifies the performance of the improved algorithm ILCDSP. We will compare ILCDSP algorithm with several typical algorithms of local community detection; respectively, they are Clauset_R, Clauset_M, and LWP algorithm. Clauset_R and Clauset_M are the algorithms taking R and M, respectively, as the objective functions. Experimental environment is the processor, Intel (R) Core (TM) i5-2400 @ 3.10 GHz CPU 3.10 GHz; the memory, 2 G; the operating system, Windows 7; and the programming language, C#.Net.

4.1. Datasets. We select three real networks and LFR benchmark network as experimental data.

(1) LFR benchmark network [14] is the most commonly used data sets in the current study of community detection. LFR benchmark networks mainly include the following parameters: N is the parameters of node degree distribution; K is the average degree of nodes in network; $\max k$ is the biggest degree of node; Min c is the number of nodes that the smallest community contains; $\max c$ is the number of nodes that the biggest community contains; and *mu* is a mixed parameter, which is the probability of nodes connected with nodes of external community. The bigger *mu* is, the more difficult the community detection is. We produce four groups of LFR benchmark network; respectively, two of the groups share these parameters $N = 1000$, $k = 20$, and $\max k = 50$, and the other two groups share parameters, $N = 5000$, $k = 20$, and $\max k = 50$; each group contains nine simulation networks, and the detailed parameters setting is shown in Table 1. The $\min c$ and $\max c$ of B1, B2, B3 and B4 were respectively set to {10, 50} and {20, 100}, that represents

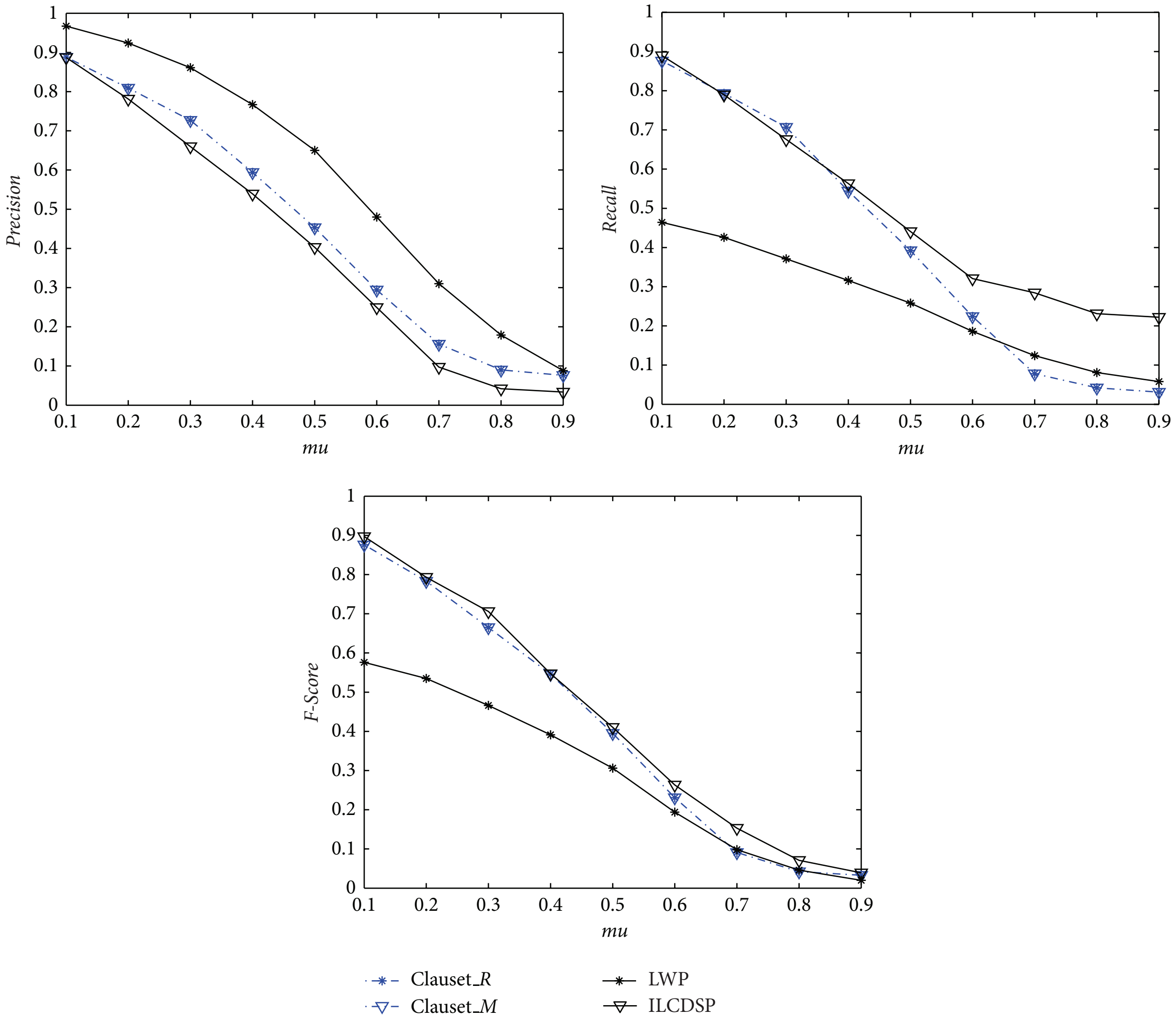


Figure 7: Comparison of B4.

Table 1: Information of LFR benchmark network.

Num	N	k	max k	min c	max c	mu
B1	1000	20	50	10	50	0.1~0.9
B2	1000	20	50	20	100	0.1~0.9
B3	5000	20	50	10	50	0.1~0.9
B4	5000	20	50	20	100	0.1 ~0.9

Table 2: Information of real networks.

Num	Name	Node number	Edge number	From
R1	Karate	34	78	[15]
R2	Football	115	613	[15]
R3	Polbooks	105	441	[15]

small community networks and big community networks. The value of *mu* in each group is set from 0.1 to 0.9, which generates nine simulation networks; these values represent that the networks change from low hybrid network to high hybrid network.

(2) The detailed information of real network data is shown in Table 2.

4.2. Evaluation. We utilize *Precision*, *Recall,* and *F-Score* as the evaluation criteria. They are used frequently in many areas such as statistics, information retrieval, and machine learning [16–18]. Some other articles about community detection also used these metrics criterions [19–23]. *Precision* is the fraction of the correct classification of nodes in the community; *Recall* is the ratio of the number of correctly classified nodes to the total number of nodes that should be agglomerated into the community; and *F-Score* is the harmonic value of *Precision* and *Recall*. The specific formulas are as follows:

$$\begin{aligned} Recall &= \frac{\left|C_R \cap C_F\right|}{C_R}, \\ Precision &= \frac{\left|C_R \cap C_F\right|}{C_F}, \\ F\text{-}Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall}, \end{aligned} \tag{9}$$

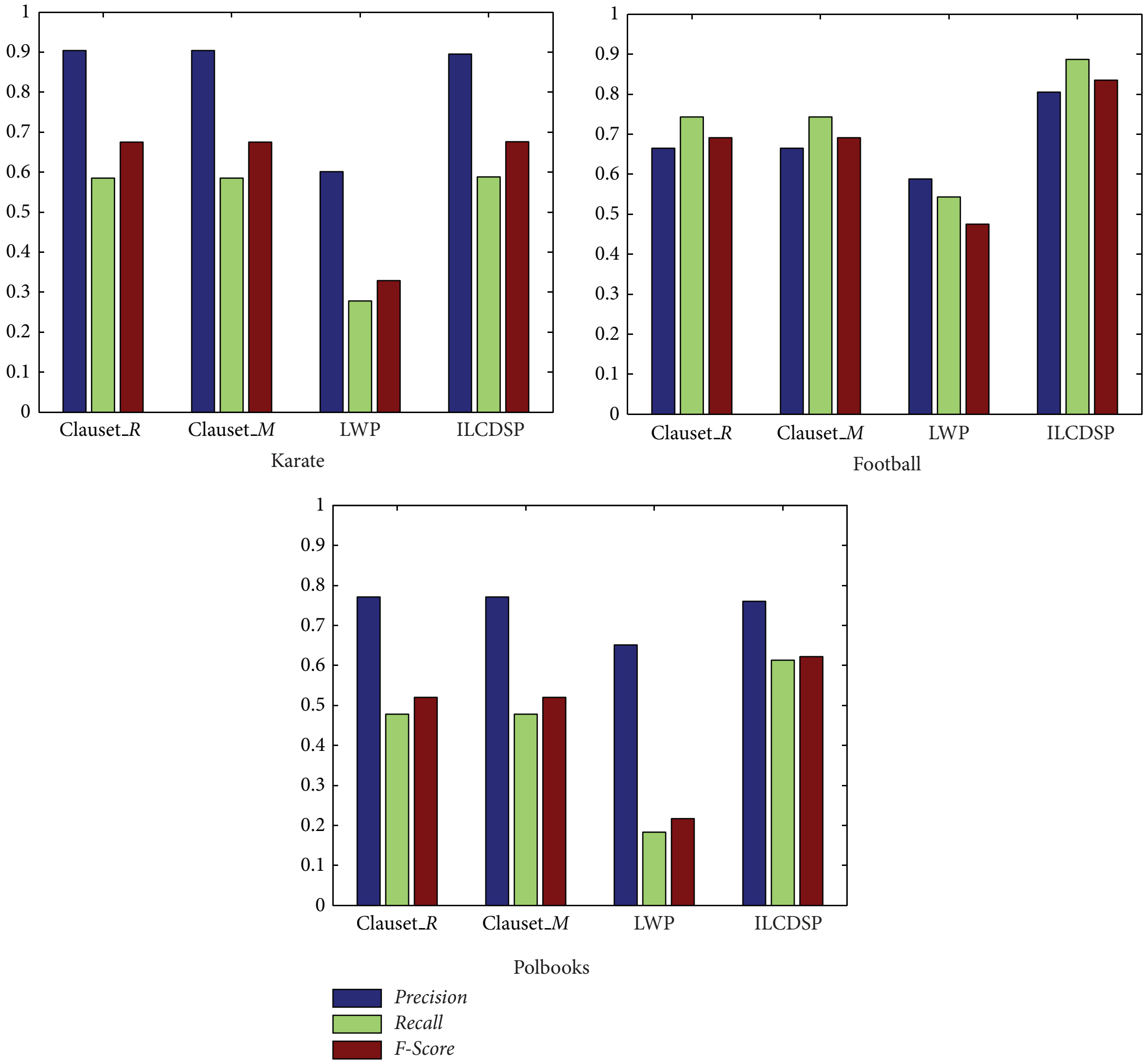


Figure 8: Comparison of different real networks.

where C_R is the local community where the source node belongs in the true partition, and C_F is the local community of the source node identified by the algorithm of local community detection. A well-performed algorithm should get high *Precision*, *Recall,* and *F-Score* at the same time. In the experiments, every node in the network is taken as a source node to discover its local community. We average the *Precision*, *Recall,* and *F-Score* for all nodes to evaluate and compare the accuracy of our algorithm against the others.

4.3. Analysis of Simulation Network. Figures 4, 5, 6, and 7 show the results of algorithms detecting local community in four groups of LFR benchmark networks (B1–B4), respectively. The horizontal is the value of *mu* which is between 0.1 and 0.9; the vertical is the three evaluations of local community detection. From these figures, it can be seen that the results of Clauset_*R* and Clauset_*M* are completely overlapped. Although the form of the community modularity *R* and *M* is different, the effect of the algorithm used them is the same. At the same time, it is obvious that the improved algorithm is better than the original algorithm LWP. Although ILCDSP does not improve the value of *Precision*, it greatly improves the value of *Recall*, thus making the value of *F-Score* greater than the LWP significantly. A well-performed algorithm should have a high *Precision*, *Recall*, and *F-Score* at the same time; thus the proposed algorithm in this paper is superior to LWP. That is to say, our algorithm can better detect the structure of local community compared with the original algorithm.

4.4. Analysis of Real Network. In our experiments, we apply each algorithm, respectively, in three real networks. Table 3 shows the results. The bold figures are the maximum values of each evaluation on each network. The results of Clauset_*R* and Clauset_*M* are the same. Most of the maximum values appear in ILCDSP. In spite of the fact that the *Precision* of

Table 3: Comparison of different real networks.

Datasets	Evaluation	Clauset_*R*	Clauset_*M*	LWP	ILCDSP
Karate	*Precision*	**0.904**	**0.904**	0.601	0.895
	Recall	0.585	0.585	0.278	**0.588**
	F-Score	0.675	0.675	0.329	**0.676**
Football	*Precision*	0.665	0.665	0.588	**0.805**
	Recall	0.743	0.743	0.543	**0.887**
	F-Score	0.691	0.691	0.475	**0.835**
Polbooks	*Precision*	**0.771**	**0.771**	0.651	0.760
	Recall	0.478	0.478	0.183	**0.613**
	F-Score	0.520	0.520	0.217	**0.622**

Clauset_*R* and Clauset_*M* in karate and polbooks is greater than ILCDSP, their *Recall* is lower than ILCDSP, which leads to the *F-Score* of the former two algorithms to be lower than ILCDSP. It is said that the overall effect of ILCDSP is the best. Through Figure 8, it can be seen intuitively that the result of our algorithm is better than other algorithms. It further shows that our algorithm cannot only detect local community of high quality in the simulated network but also could be applied to detect community structure effectively in real networks.

5. Conclusions

This paper proposed an improved local community detection algorithm—ILCDSP. The algorithm firstly sets selection probability for each candidate node, making the nodes with high selection probability more likely to be selected, and then it randomly screens nodes. The algorithm will process to the best direction, so as to improve the accuracy of local community detection. Experimental results show that ILCDSP could detect the structure of the local community more effectively than other algorithms both in the real and simulate networks.

Although this proposed algorithm improves the accuracy of community detection, it is not stable enough; at the same time, it remains to be further researched and improved in time.

Conflict of Interests

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work; there is no professional or other personal interest of any nature or kind in any product, service, or company that could be construed as influencing the position presented in or the review of the paper entitled.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities, the National High Technology Research and Development Program of China (863 Program) (no. 2012AA011004 and no. 2012AA0622022), the Fundamental Research Funds for the Central Universities under Grant (no. 2013XK10), and the Doctoral Fund of the Ministry of Education (no. 20100095110003 and no. 20110095110010).

References

[1] 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.

[2] J. Lee, S. P. Gross, and J. Lee, “Modularity optimization by conformational space annealing,” *Physical Review E*, vol. 85, no. 5, Article ID 056702, 2012.

[3] R. H. Shang, J. Bai, L. C. Jiao et al., “Community detection based on modularity and an improved genetic algorithm,” *Physica A*, vol. 392, no. 5, pp. 1215–1231, 2013.

[4] L. Huang, R. Li, H. Chen, X. Gu, K. Wen, and Y. Li, “Detecting network communities using regularized spectral clustering algorithm,” *Artificial Intelligence Review*, 2012.

[5] H.-W. Shen and X.-Q. Cheng, “Spectral methods for the detection of network community structure: a comparative analysis,” *Journal of Statistical Mechanics*, vol. 2010, no. 10, Article ID P10020, 2010.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics*, vol. 2008, no. 10, Article ID P10008, 2008.

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

[8] L. Šubelj and M. Bajec, “Unfolding communities in large complex networks: combining defensive and offensive label propagation for core extraction,” *Physical Review E*, vol. 83, no. 3, Article ID 036103, 2011.

[9] 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.

[10] A. Clauset, “Finding local community structure in networks,” *Physical Review E*, vol. 72, no. 2, Article ID 026132, 2005.

[11] F. Luo, J. Z. Wang, and E. Promislow, “Exploring local community structures in large networks,” *Web Intelligence and Agent Systems*, vol. 6, no. 4, pp. 387–400, 2008.

[12] Y. J. Wu, H. Huang, Z. F. Hao et al., “Local community detection using link similarity,” *Journal of Computer Science and Technology*, vol. 27, no. 6, pp. 1261–1268, 2012.

[13] J. P. Bagrow and E. M. Bollt, “Local method for detecting communities,” *Physical Review E*, vol. 72, no. 4, Article ID 046108, 2005.

[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] http://www-personal.umich.edu/~mejn/netdata/.

[16] Y. Lu, L. Zhang, J. Liu, and Q. Tian, “Constructing concept lexica with small semantic gaps,” *IEEE Transactions on Multimedia*, vol. 12, no. 4, pp. 288–299, 2010.

[17] J. Yang, R. Cai, Y. Wang et al., “Incorporating site-level knowledge to extract structured data from web forums,” in *Proceedings of the 18th International Conference on World Wide Web*, pp. 181–190, 2009.

[18] X. Li, Y.-Y. Wang, and A. Acero, “Learning query intent from regularized click graphs,” in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’08)*, pp. 339–346, July 2008.

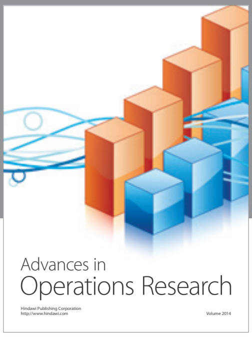
[19] J. Chen, O. R. Zaï, and R. Goebel, “Local community identification in social networks,” in *Proceedings of the International Conference on Advances in Social Network Analysis and Mining (ASONAM ’09)*, pp. 237–242, July 2009.

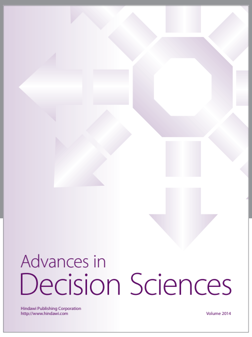
[20] X. Zhang, L. Wang, Y. Li et al., “Extracting local community structure from local cores,” in *Proceedings of the 16th International Conference Database Systems for Advanced Applications*, vol. 6637, pp. 287–298, 2011.

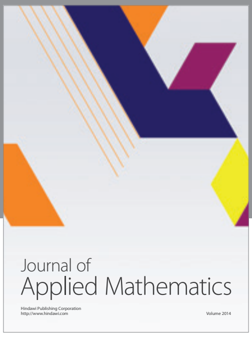
[21] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Physical Review E*, vol. 70, no. 6, Article ID 066111, 2004.

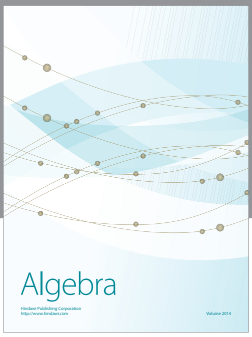
[22] J. P. Bagrow, “Evaluating local community methods in networks,” *Journal of Statistical Mechanics*, vol. 2008, no. 5, Article ID P05001, 2008.

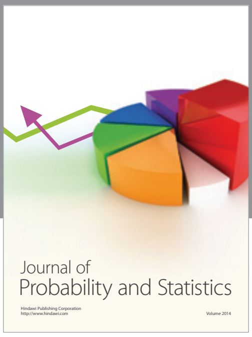
[23] C. Qiong, W. Tingting, and F. Ming, “Detecting local community structure in complex networks based on local degree central nodes,” *Physica A*, vol. 392, no. 3, pp. 529–537, 2013.


Advances in Operations Research

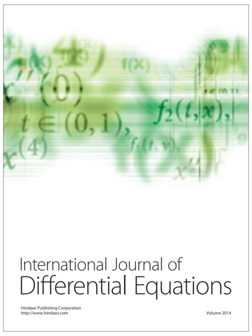

Advances in Decision Sciences


Journal of Applied Mathematics

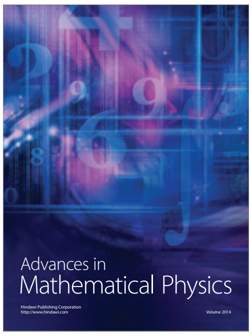

Algebra

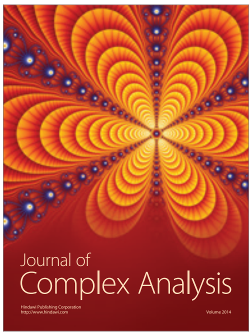

Journal of Probability and Statistics

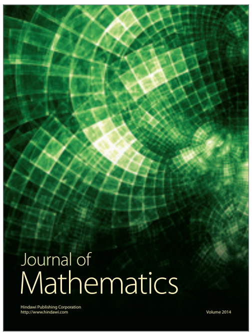
The Scientific World Journal

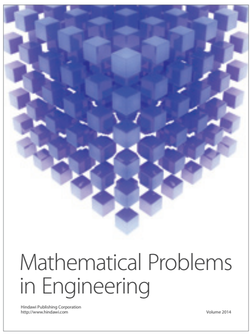

International Journal of Differential Equations

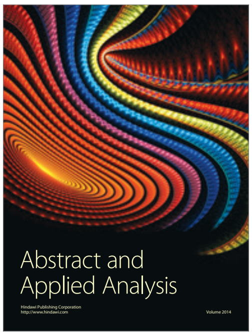
International Journal of Combinatorics

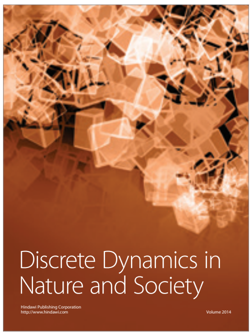

Advances in Mathematical Physics

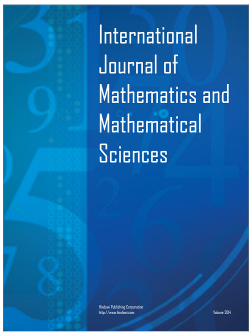

Journal of Complex Analysis


Journal of Mathematics


Mathematical Problems in Engineering

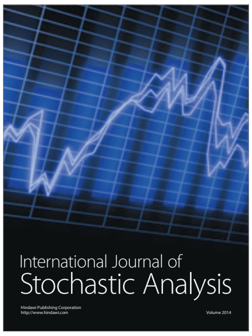

Abstract and Applied Analysis


Discrete Dynamics in Nature and Society


International Journal of Mathematics and Mathematical Sciences

Journal of Discrete Mathematics

Journal of Function Spaces


International Journal of Stochastic Analysis

Journal of Optimization