



## Original article

An efficient shortest path approach for social networks based on community structure<sup>☆</sup>Maoguo Gong<sup>a,\*</sup>, Guanjun Li<sup>a</sup>, Zhao Wang<sup>a</sup>, Lijia Ma<sup>a</sup>, Dayong Tian<sup>b</sup><sup>a</sup> Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China<sup>b</sup> Center for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Broadway, NSW 2007, Australia

Available online 2 June 2016

## Abstract

Finding the shortest path (SP) in a large-scale network analysis between any two nodes is a tough but very significant task. The SP can help us to analyze the information spreading performance and research the latent relationship in the weighted social network, and so on. As the size of the social network increases, the traditional SP algorithms have poor performance and there is not a suitable algorithm for weighted social network. Some features of the network analysis are beneficial to solve this problem, and community structure ignored by the traditional methods is one of the most important features. In this paper, we propose a shortest path algorithm based on community detection (SPCD) by integrating community detection algorithm with traditional search methods. SPCD constructs a community graph by using community structure to narrow the searching scope. The algorithm presented improves the time efficiency and maintains the accuracy scale of the SP. Experimental results on five real-world networks demonstrate the effectiveness of the proposed methods for the SP problem.

Copyright © 2016, Chongqing University of Technology. Production and hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

**Keywords:** Shortest path; Community structure; Weighted social network

## 1. Introduction

Recently, studies on social network have attracted a lot of attention from sociology, informatics and computer science [1,2]. A social network is a social structure made up of a set of social actors (such as individuals or organizations) and a set of the dyadic ties between these actors. A huge amount of data and resources make it critical to analyze social network related problems. A series of research on social network have been done in many aspects of sociology, such as social influence,

social groupings, inequality, disease propagation and communication of information [3]. The SP problem which calculates the distance or relationship between the nodes in the social network is an extremely important problem of social network analysis. It can be used to study the behavior of information spreading, especially fastest information spreading. The recommendation system is also based on the SP problem. For example, when we analyze the relationship between scientists, the scientists collaboration network is established and analyzed utilizing the SP problem. Analyzing the scientist collaboration network, the work and paper reviewing efficiency can be improved. The SP problem serves as an elementary aspect when analyze the network structure, for example betweenness and average distance. With the increase of network scale, the previous algorithms may not suitable for the existing network, so we need effective methods for large-scale network. In this paper, we focus on the SP problem between any two nodes in the weighted social network. The SP problem concerns with finding the shortest path from a

<sup>☆</sup> This work was supported in part by the National Natural Science Foundation of China under Grants 61273317, and 61422209, the National Program for Support of Top-notch Young Professionals of China, and the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20130203110011.

\* Corresponding author.

E-mail address: [gong@ieee.org](mailto:gong@ieee.org) (M. Gong).

Peer review under responsibility of Chongqing University of Technology.

specific origin to a specified destination in a given network while minimizing the total cost associated with the path. SP is a fundamental problem in social network. In a graph, finding the path with the minimum cost from a source node  $s$  to a destination node  $d$  is called the point-to-point (P2P) problem, but a common variant fixes a single node as the source node and finds shortest paths from the source to all other nodes in the graph. In addition to P2P problem, other shortest path problem, such as single-destination, all-pairs and so on, could be converted to single-source shortest path. The single-source problem with non-negative arc lengths has been studied most extensively [4]. For a general network, traditional dijkstra's algorithm could be used for solving all the non-negative, weighted or non-weighted networks without any domain-specific information. Though it is a single-source algorithm, we can transform it to a P2P algorithm by terminating at destination node. But it has high time complexity of  $O(n^2)$ , so it might not work well on the SP in social network, especially in large-scale social network. The time complexity of modified standard breadth-first is  $O(n)$ , but it is suited to non-weighted network [5]. With the increase of network scale, authors used domain-specific information about the network to deal with it. In traffic networks, researchers have adopted the natural hierarchies to speed up a shortest path algorithm significantly [6–10], and more algorithm are given in [11]. It is obvious that these algorithms could not be applied to social network. From the same point of view in traffic networks, the feature of social network can be utilized to the SP problem. Community feature is one of the most important characteristics of social network.

In recent years, community detection has gained a lot of attention in the field of network analysis. Community structure is similar to the small world effect and the right-skewed degree distributions which are an important distinctive property in a complex network [12]. Qualitatively, a community is defined as a subset of a graph in which the interconnections of nodes are denser than those observed with the rest of the network [13,14]. For the general case of a weighted graph, many approaches mainly focused on various criteria including modularity [15], structural density [16] and partition density [17]. Blondel et al. introduced a fast greedy approach (BGLL) to optimize the modularity [18]. Besides, parameter-free hierarchical network clustering algorithm (SHRINK) proposed by Huang et al. combines the advantages of density-based clustering and modularity optimization methods [19]. In the field of social networks, Xie et al. proposed a general speaker-listener algorithm named SLPA based on label propagation [2]. Evolutionary computation is also an important and influential method [20,20,21]. Community detection clusters the edges into two classes: the shorter ones and the longer ones. In that case, we can pay our attention to the large edges and find out the communities including the SP, and then searching the SP in these communities. As far as we know, there is no method using community information to solve P2P problem in weighted social network. In this paper, we propose a new shortest path algorithm based on community detection (SPCD). Using community information, SPCD can narrow

down the search space and decrease computing time for a path while sacrificing accuracy within a certain level. It can strike a balance between accuracy and time efficiency while being faced with different problems.

The paper is organized as follows. Section2 describes the proposed SPCD algorithm in detail. Section3 is experiment in five datasets. This subsection analyses accuracy and efficiency. Moreover the influence of community path number and community detection method to SPCD is discussed. We conclude the paper is Section4 and point out the future work.

## 2. Algorithm description

In this section, the proposed algorithm is described in detail. First, the community definition and detection is introduced, and then the method of constructing community graph is described. The  $k$  shortest community paths are given in subsection2.3. Finally, the shortest path in sub-graph is given.

### 2.1. Community definition and detection

Adopted symbols are listed in Table 1. Communities are groups of vertices which share common properties and/or play similar roles within the graph. Qualitatively, connections between the nodes in a community are denser and closer than connections with the rest of the network. For SP in weighted social network, we define the community in such a way that the connections in the community have a lower cost but connections between the communities have higher cost. In this paper, we use distance to measure the relationship of any two nodes in the network. The smaller the distance, the more highly related the any two nodes. Thus the target of the SP problem is to find the minimal distance between any two nodes in the network. As for the other applications which need to find the maximum the distance, they can be easily transformed into this minimization problem by reverting the evaluation function.

A demonstration of community in a network is shown in Fig. 1. The weight of edge between  $j$  and  $l$  is 20, which is

Table 1  
Notation.

Symbols	Definition
$G = (V, E)$	Target network, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices and $E$ is the set of edges
$e_{ij} \in E(i, j \in V)$	The non-negative edge between vertex $i$ and $j$
$s, d$	Source node and destination node
$C$	Community information of network
$c_i (i \in V)$	Vertex $i$ 's community
$\bar{c}$	The set of nodes in community $c$
$\overleftrightarrow{E}_{ij}$	The set of edges between community $i$ and $j$
$SG = (SV, SE)$	Sub-graph, $SV$ is the set of vertices and $SE$ is the set of edges
$P_{ij}$	The path from $i$ to $j$
$SP$	The shortest path
$ SP $	The weighted sum of $SP$
$KSCP$	The shortest path of community graph
$SV$	Resort the $SV$
$SP'$	The shortest path in sub-graph

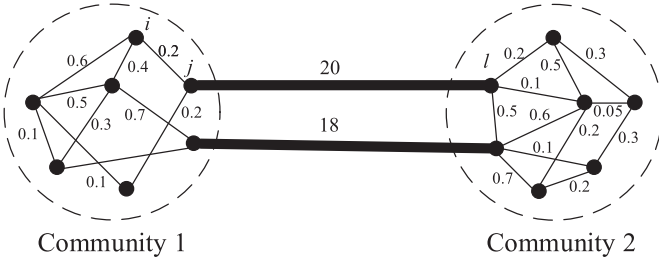


Fig. 1. Community detection by our definition.



Fig. 3. Edge between communities can replace the path.

much bigger than 0.2, the weight of edge between  $i$  and  $j$ , i.e.  $e_{ij} \ll e_{jl}$ . So according to our definition of community, nodes  $i$  and  $j$  are clustered into the same community ( $c_i = c_j$ ) while  $l$  is clustered to another community ( $c_l \neq c_i$ ).

Xie J. et al. [2] suggest a general framework that uses Speaker-listener Label Propagation Algorithm (SLPA) to detect communities and it clusters the bigger edges into a community for weighted network. In this case, we cannot use Xie's method directly. We reverse the target network before performing SLPA. The reversal network is obtained as follows. First, find the maximum value of the weights of the edges, namely the maximum edge cost, denoted by  $maxe$ . Second, generate a positive value 1 or any other positive value and added with  $maxe$ , denoted by  $maxe'$ , i.e.  $maxe' = maxe + 1$ . Creating a positive value is to prevent the maximum edge from becoming zero in the next step. Then the cost of the edges in the reversal network are  $maxe'$  minus the original corresponding non-negative cost. The cost of edges remains the same if the cost is zero and we can get the reversal network (A example in Fig. 2). Through the above steps, the bigger edge in the original network is reversed to the smaller edge in the reversal network, so that the community information that comes from performing the Xie's method on the reversal network is the community information that suits our community definition.

## 2.2. Constructing community graph

We construct the community graph to make full use of the network community information. According to our definition of community, these edges between communities have bigger cost, so we can use the edges between communities to represent these paths. In Fig. 3, we know  $e_{ij}$  is bigger than  $e_{si}$

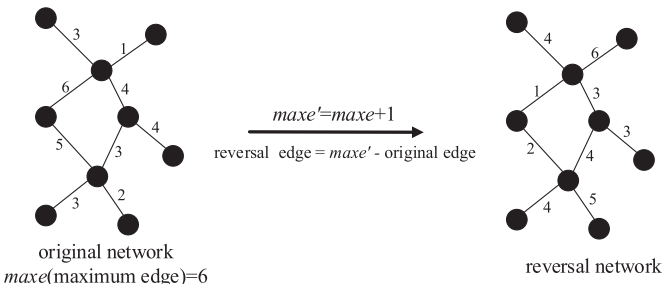


Fig. 2. Transforming network into the reversal network.

and  $e_{jd}$ , in other word,  $e_{ij} \approx e_{sd}$ , so  $P_{ij}$  can be used to replace  $P_{sd}$  and we can focus on the edges between community when searching the shortest path from node  $s$  to node  $d$ .

According to the above analysis, we can build the community graph  $CG = (CV, CE)$  based on the result of community detection, where each vertices in  $CV$  represents a community of the original network and edges  $CE$  are the minimum cost between each two communities  $\min |E|$ . Compared with the edges between communities, the cost of edges in the community is smaller and ignored temporarily, so we treat all the nodes in the community as one node, that is to say, the community these nodes located is the node of community graph. There might be more than one edge between communities and we choose the minimum cost edge as the connection of community graph between corresponding community nodes. The aim of SP problem is searching the smallest path, so we select the minimum cost edge in this step. After that we model the original network to a new weighted community graph.

See an example in Fig. 4. The original network is clustered into three communities, so the community graph has three nodes  $CV = \{cv_1, cv_2, cv_3\}$ . Edge  $ce_{1,2}$  in community graph is equivalent to the minimum cost edge between community 1 and 2  $\min |E_{1,2}| = 4.0$ , e.g.  $ce_{1,2} = 4.0$ . In a similar way we get edges  $ce_{1,3} = 3.3$  and  $ce_{2,3} = 8.0$ .

## 2.3. K shortest community paths

So as to find the SP in the original network, the shortest community path includes the SP should be found primarily.

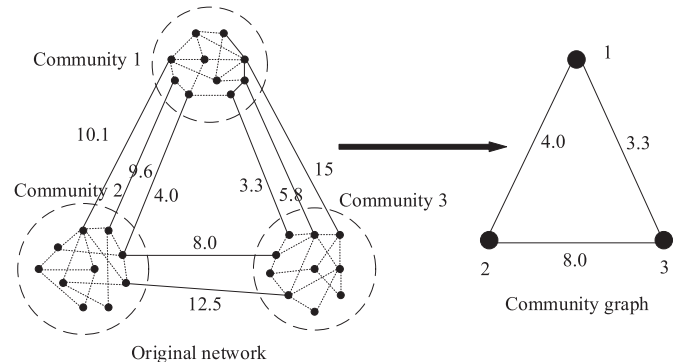
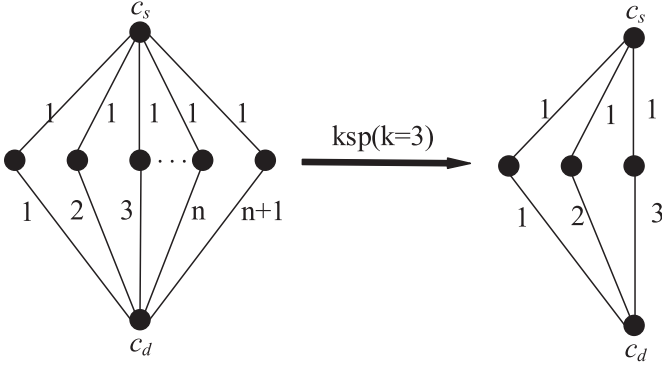


Fig. 4. Building community graph.

Fig. 5.  $k$  shortest path in the community graph.

There is a situation that the SP is not included in the shortest community path but in the second shortest or  $k$ -th shortest community path, so the next step is calculate the  $k$  shortest community paths.  $K$  shortest paths problem means finding the top  $k$  paths from source node to destination node in order of increasing cost. If the number of paths from source community to destination community is less than  $k$  in our community graph, all the paths are selected.

Original network was modeled to community graph by our method. The real shortest path in the original network is included in the community graph. Moreover, there is a high probability that the actual shortest path can be found in the shortest community path. However, we cannot ignore the situation that the real shortest path is not concluded in the shortest community path, but in another path among the  $k$  shortest community paths. Based on the above analysis, we calculate the  $k$  shortest community paths  $SP_k = \{sp_1, sp_2, \dots, sp_k\}$ , where  $|sp_1| < |sp_2| < \dots < |sp_k|$ . We use Yen's algorithm [22] to find the exact  $SP_k$  from the source community  $c_s$  to the destination community  $c_d$  (See example in Fig. 5). When the number of paths from  $s$  to  $d$  in the graph is less than  $k$ , we take all the existing paths as the  $SP_k$ . If  $s$  and  $d$  are in the same community, the shortest community path is the community, and the  $k$  shortest

community paths are the community with its neighbor communities. Intuitively, the scale of community graph is far smaller than that of the original graph, so the  $SP_k$  can be obtained efficiently.

#### 2.4. The shortest path in sub-graph

We utilize the  $k$  shortest community paths to establish the search sub-graph. There may be two community paths intersect, e.g.  $sp_1 = (c_1, c_2, c_3, c_4, c_5, c_6)$  and  $sp_2 = (c_{11}, c_{32}, c_3, c_4, c_5, c_{19})$  have same communities  $(c_3, c_4, c_5)$ , so we get the search communities  $C' = c_1, c_2, \dots, c_n$  by

---

##### Algorithm 1 Framework of the SPCD algorithm

---

**Input:**  $G, C, s, d$

**Step 1)** Build  $CG$ :

**Step 1.1)** Initialize  $CE = \infty$

**Step 1.2)** For every edge  $e_{ij} \in ce_{c_i c_j}$   
if  $c_i \neq c_j$  and  $e_{ij} < ce_{c_i c_j}$   
 $ce_{c_i c_j} = e_{ij}$

**Step 2)** Calculate the  $KSCP$

**Step 2.1)** Source point in  $CG$  is  $c_s$ , destination point:  $c_d$

**Step 2.2)** Get the  $KSCP$  by  $k$  shortest path algorithm

**Step 3)** Compute the  $SP$

**Step 3.1)** The search community:

$C' = \text{union}(KSCP)$

**Step 3.2)** The search nodes of sub-graph

$SV = C'$

**Step 3.3)** The search edges of sub-graph

$SE = E(C')$

**Step 3.3)** Get the  $SP$  by  $SP$  algorithm

**Output:**  $SP (\{v_1, v_2, \dots, v_{i-1}, v_i\} \text{ and } |SP|)$

---

striving for the intersection of the  $k$  shortest community paths. Next we get the nodes of sub-graph  $SV = \{c_1, c_2, \dots, c_n\}$ , and sort from  $SV$  to  $SV'$ . For example, we obtain a set of sub-node  $SV = 5, 3, 8, 22, 66, \dots, 88$  that from the original graph and sorts it as  $SV' = 1, 2, 3, \dots, n$ .  $SE$  is equal to the cost between the

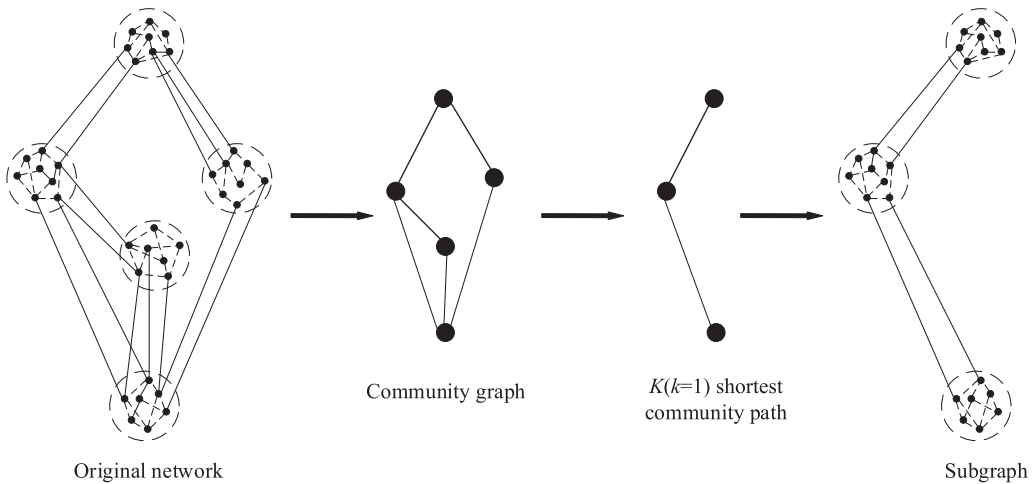


Fig. 6. Decrease the scale of graph by our method.

corresponding nodes in the original graph, e.g.  $se_{1,2} = e_{5,3}$  that sub-node 1 is sorted from the original graph node 5 and 2 from 3. The following step is calculating the SP' by SP algorithm in the sub-graph, but the node in SP' we get is from SV', so we need to get the original SP from the SP' according to the sorting. If  $SP' = 1, 3, \dots, n$ , the SP we wanted is 5, 8, ..., 88. The cost of the path  $|SP| = |SP'|$ . From above we know that the SV is a part of V, in other words, sub-graph is much smaller than the original network, so we can find the SP with less time.

Our algorithm utilizes community information to build a community network and find the community paths, then extracting sub-graph from the original network. Finally, obtaining the SP from the sub-graph. This community path can help us narrow down the search space size, so the searching time is saved (see a example in Fig. 6). A high precision requires a larger  $k$ , but the larger  $k$  leading to increase the time complexity, so we must set a proper  $k$ . The pseudo-codes for general algorithmic flow of this algorithm are listed in Algorithm.1.

### 3. Experiment and analysis

Our algorithm SPCD is implemented in Matlab 2012a. All the experiments were run on a Windows 7 server with 2 2.00 GHz Intel(R) Xeon(R) CPU E5-2620 and 64 GB of memory.

#### 3.1. Datasets

We ran experiments in five collaboration networks in which the nodes mean scientists and a link between two scientists is established by their coauthorship of one or more scientific papers. Collaboration network has a more promising source of data and this network is in some ways more truly a social network than many other networks [3].

- (1) Coauthorships in network science: coauthorship network of scientists (net-science) working on network theory and experiment including publications up until early 2006, as compiled by M. Newman in May 2006 [23]. The network has a total of 1589 scientists in it, from a broad variety of fields [23].
- (2) High-energy theory (hep-th) collaborations: weighted network of coauthorships between scientists posting preprints on the High-Energy Theory E-Print Archive [3] between Jan 1, 1995 and December 31, 1999 [24]. The network contains 8361 nodes and 15751 edges.
- (3) Astrophysics collaborations (astro-ph): weighted network of coauthorships between scientists posting preprints on the Astrophysics E-Print Archive [25] between Jan 1, 1995 and December 31, 1999 [24]. The network contains 16706 nodes and 121251 edges.
- (4) Condensed matter collaborations 1999 (con-mat): weighted network of coauthorships between scientists posting preprints on the Condensed Matter E-Print Archive [26] between Jan 1, 1995 and December 31, 1999 [24]. The network contains 16726 nodes and 47594 edges.

Table 2  
Some fundamental statistics for the datasets of nuclear experiment.

Nuclear experiment Los Alamos e-Print Archive																					
	Year	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Papers		46	87	125	184	300	328	416	503	561	700	724	673	810	787	876	935	1258	1380	1310	1306
Authors		413	488	825	966	1434	1803	1914	2101	2153	2876	2690	3026	3894	4008	3916	5643	6000	6823	7370	7035
Edges		5918	7669	14402	11282	16840	27169	23438	23361	29694	46722	29161	53265	67901	87006	96406	165598	170792	219627	244326	273598



- (5) Datasets of nuclear experiment (nucl-ex): published in Los Alamos e-Print Archive [27] from December 1994 to the present. We construct the network of every year from 1994 to 2014 and make some statistics. The authors (i.e. vertices) that identify each author by his or her surname and first initial only [3] and weight [5] are defined by M. Newman.

The scale of these collections is listed in Tables 2 and 6.

### 3.2. Performance evaluation

We evaluate the accuracy of the proposed algorithm in the search process for the SP between nodes. We choose 500 existing pairs of nodes randomly and run 10 times independently for each dataset to get average accuracy. The approximation error *aper* is:

$$aper = \sum_{i=1}^N \frac{(\hat{d}_i - d_i)}{d_i} / N$$

where  $d_i$  is the actual distance for  $i$ -th pair of nodes and  $\hat{d}_i$  is the approximation value. The actual distance was measured by dijkstra's algorithm [28] with time complexity  $O(n^2)$ . It can be optimized to  $O(|E| + |V|\log|V|)$  by a Fibonacci heap [29]. We must guarantee a minimum *aper* that is different in some cases. Set the minimum *aper* = 0.05 for all the datasets in this paper. The improve time:

$$eff = \sum_{i=1}^N \frac{t_i}{\tilde{t}_i} / N$$

where  $t_i$  is the run time by dijkstra's algorithm and  $\tilde{t}_i$  is that by our method. The time is the mean value of all the 500 random node pairs and minimum, maximum are also presented.

### 3.3. Experimental results

There are too many networks so that we show the experimental results in two tables (Tables 3 and 7). In Table 7, we can see that time efficiency increase by numerous times in efficiency by our method in the case of high precision that *aper*  $\leq$  0.05. The number is the *aper* in the parentheses. For net-science network, the *aper* is 0.0363 that approximately equals zero, while the *eff* is 387.58. In the other three networks, the greatest *aper* that is 0.0479 of con-mat less than 0.05, and the *eff* is 2740.48. For astro-ph network, its *eff* is nearly reach up to 3000 with that *aper* is 0.0473. It is shows that SPCD can find the SP in a very tiny sub-graph network.

In these networks (see Table 3), we set  $k$  to 1 in most cases and get the approximate shortest path with high time efficiency. The *eff* of 1998 is 104.7 and the *eff* from 1995 to 1997 is less than 100, but their *aper* are zero. It means that we can find the exact SP with improvement efficiency. Maximum *eff* is 701.9 in 2010 and the *eff* is greater than 300 since 2004,

Table 3  
Results of nuclear experiment in *aper* and *eff*.

year	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
<i>aper</i>	0	0	0	0	0.007	0.005	0.021	0.005	0.028	0.027	0.006	0.048	0.037	0.025	0.043	0.046	0.037	0.05	0.04	0.047
<i>eff</i>	26.79	21.66	62.61	104.7	120.3	157	192	256	173.5	340.2	306.8	305.1	290.9	250.4	316.7	701.9	540.8	94.57	394.1	201.7

Table 4  
The suitable  $k$  in small-scale network.

Nuclear experiment													net-science
	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	
$k$	1	1	1	1	2	2	2	1	2	2	2	3	3
$aper$	0	0	0	0	0	0.003	0.001	0.004	0.012	0.008	0.002	0.017	0.014

Table 5  
The suitable  $k$  in large-scale network.

Nuclear experiment									hep-th	astro-ph	con-mat
	2007	2008	2009	2010	2011	2012	2013	2014			
$k$	3	6	5	5	8	6	4	5	9	12	14
$aper$	0.009	0.011	0.01	0.004	0.011	0.049	0.039	0.047	0.047	0.047	0.048

Table 6  
Some fundamental statistics for the studied collaboration networks.

Network	net-science	hep-th	astro-ph	con-mat
Total number of nodes	1589	8361	16706	16726
Total number of vetices	2742	15751	121251	47594

Table 7  
Computational efficiency: the time of calculate SP and  $eff$ .

	Net-science	hep-th	astro-ph	com-mat
$aper$	0.0363	0.0472	0.0473	0.0479
$eff$	387.58	960.65	2915.97	2940.48

although 2012's  $eff$  is only 94.51 but it is obvious that our algorithm is effective.

### 3.4. Parameter $k$

The result will be more accurate with the increase of  $k$ , the number of shortest community path. Though  $k$  is set to 1 in

most cases in nuclear experimental datasets for small  $aper$ , large-scale network needs greater  $k$  if we require that  $aper \leq 0.05$ . In small-scale network nuclear experiment that from 1995 to 2006 and network science (see Table 4), we only need a small  $k$  to obtain a very high accuracy shortest path (above 99%), but we can find these paths much efficiently. Big  $k$  would be needful in large-scale network for similar or required accuracy (see Table 5), especially, in hep-th, con-mat and astro-ph. In this case, our algorithm still has a valuable effect in efficiency.

For small-scale network, a smaller  $k$  may result in better results, but for a large-scale network, we need to select a suitable  $k$  to obtain satisfying results. Fig. 7, the accuracy of the results greatly improved with the increasing of  $k$ . As shown in Fig. 7, the  $aper$  has a rapid decrease with increasing of  $k$ , which indicates that the proposed method could find the scope of SP efficiently. From the aspect of practical application, you can choose the appropriate  $k$  to satisfy the requirement for accuracy.

As we know,  $k$  is associated with the size of the sub-graph, so that the size of subgraph will increased and the efficiency of

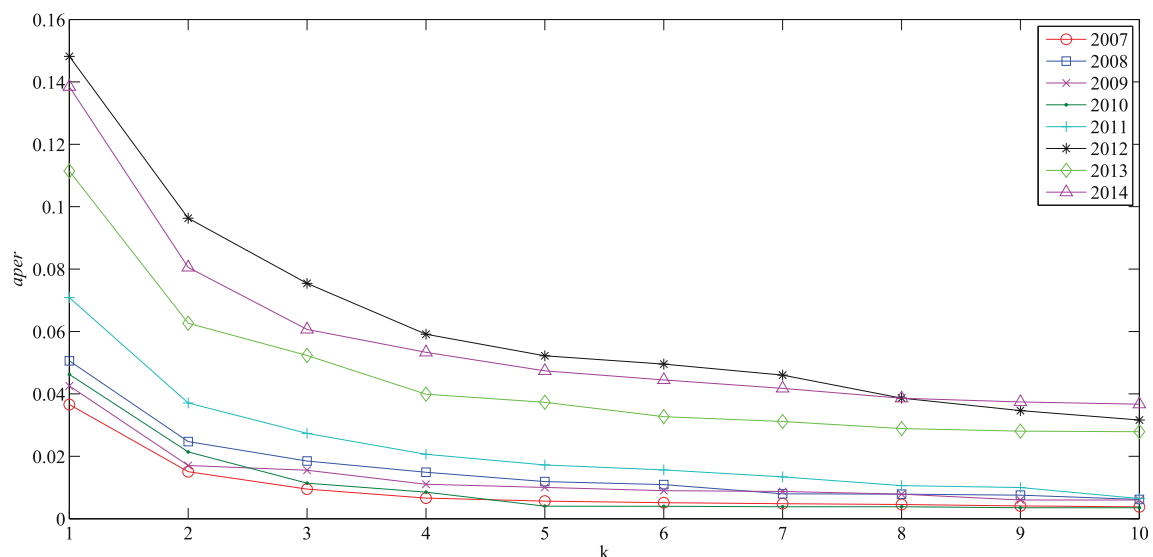


Fig. 7. The  $aper$  with  $k$  in nuclear experiment datasets.

search will reduced with the increasing of  $k$ . Fig. 8 confirm our analysis. In the nuclear experiment network datasets, the efficiency of searching the shortest path will be weakened with the increase of  $k$ . Taking this into account, we need a smaller  $k$  to protect the efficiency, i.e. a great  $eff$ . In the astro-ph, con-mat and hep-th,  $eff$  will also deteriorated with the increasing of  $k$ .

We can learn from the above experimental results that the accuracy and efficiency are the two objectives to be optimized in the proposed method. Increasing  $k$  can improve the result accuracy but damage the efficiency. It is desirable that we need great  $eff$  and  $(1-aper)$  (small  $aper$  is perfect), but we can't make both great from Fig. 9. Then we need a tradeoff of

accuracy and efficiency that  $aper$  is less than 0.05 in our experiment.

SPCD uses community information to search the SP, so we need an accurate community detection method. We choose two different approaches that BGLL and SHRINK for general network and one that SLPA for the social network.

Fig. 10 compares the  $aper$  returned by the three community detection methods. It is observed that SLPA and BGLL performed well while SHRINK can not be used in our method on account of the poor  $aper$ . Between SLPA and BGLL, SLPA is better from the network of nucl-ex2010, nucl-ex2012 to nucl-ex2014. In Fig. 11, it obvious that BGLL has the worst efficiency even though it may be better than SLPA in nucl-

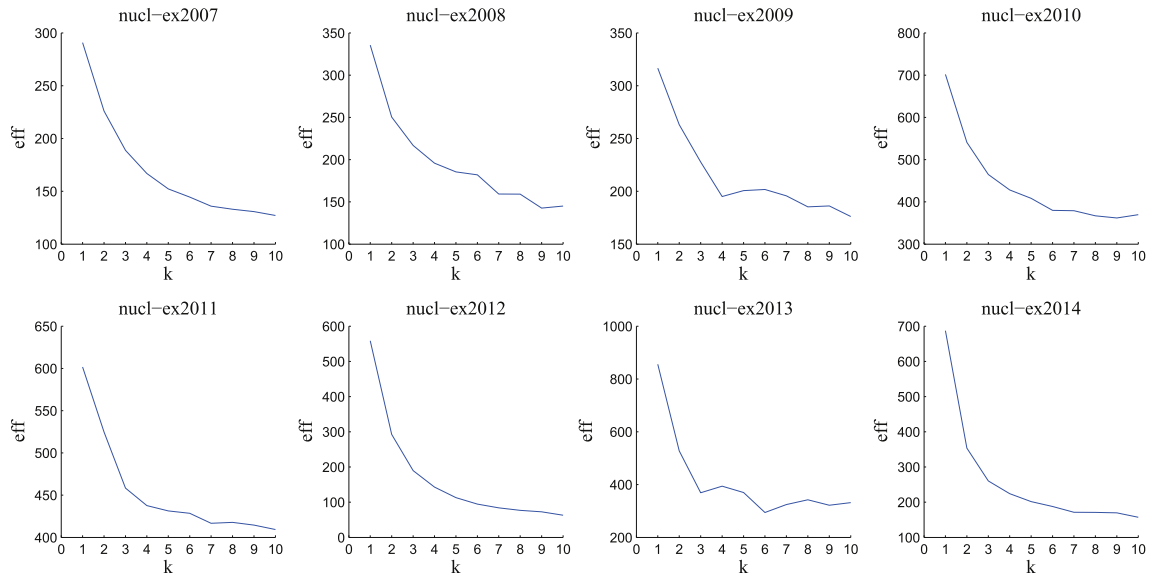


Fig. 8. The  $eff$  with  $k$  in nuclear experiment datasets.

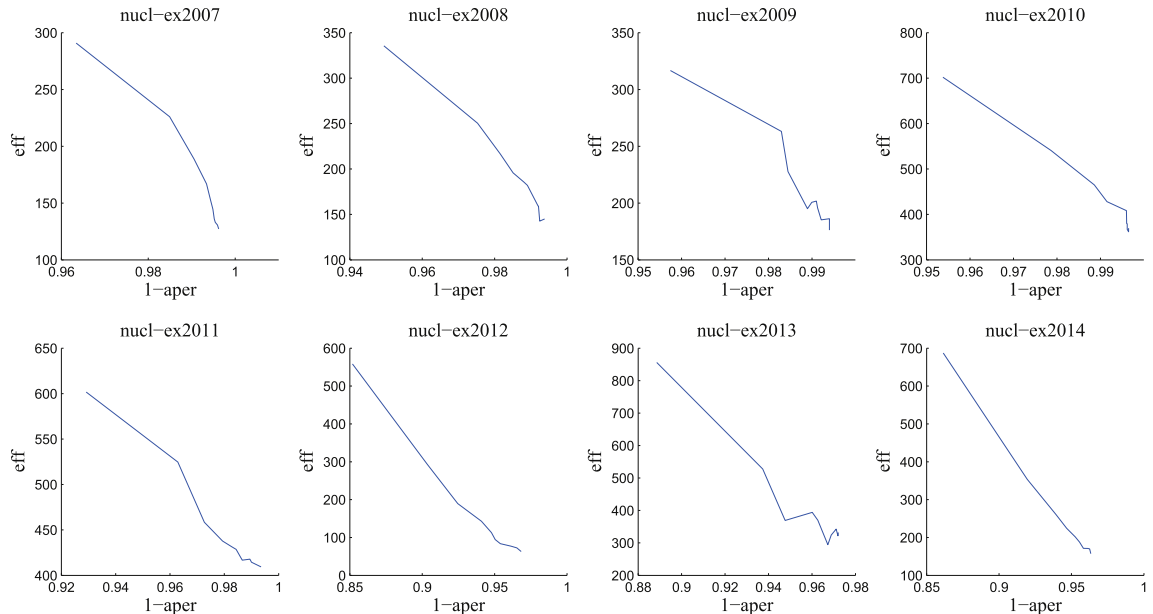
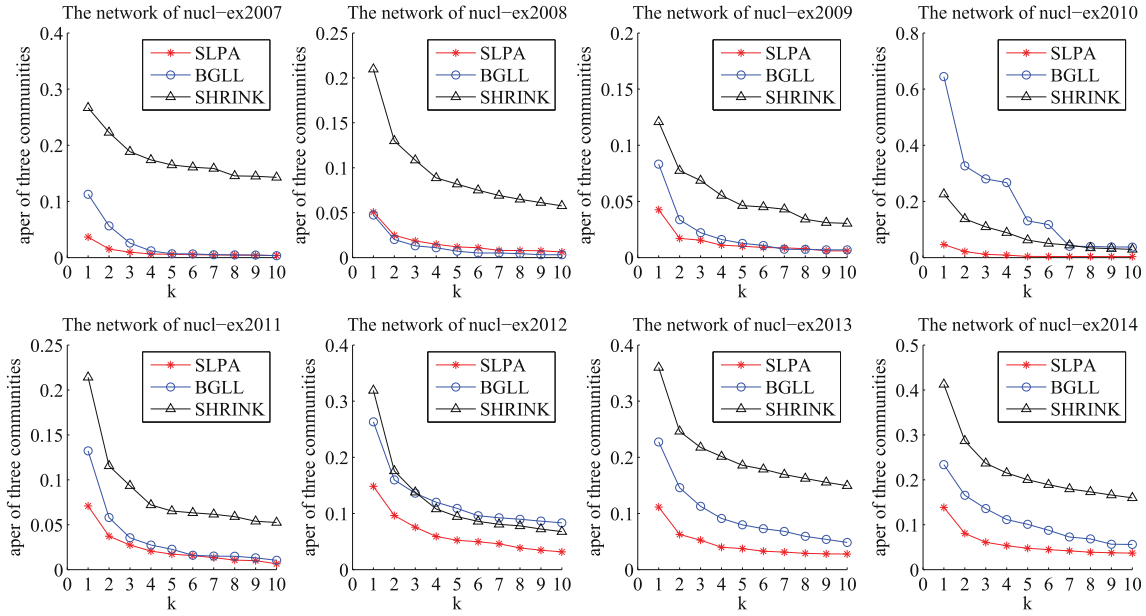
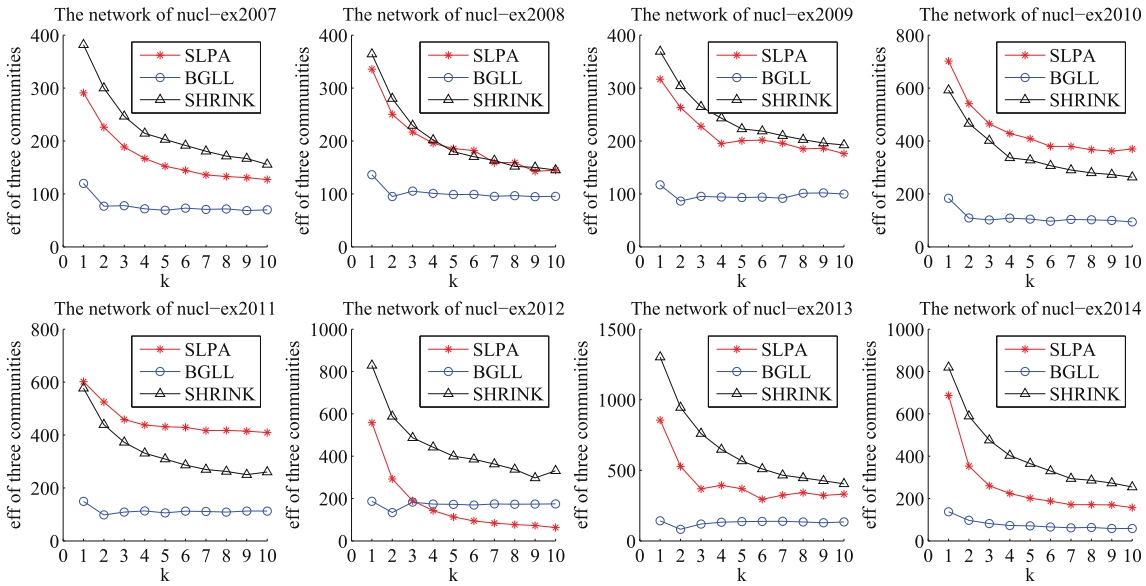


Fig. 9. The conflict between  $aper$  and  $eff$ .



Fig. 10. The *aper* of three communities on nul-ex datasets.Fig. 11. The *eff* of three communities on nul-ex datasets.

ex2012, so the SLPA is the most suitable community detection method for SPCD of *aper* and *eff* considers.

#### 4. Conclusions

In this paper, we proposed a novel and efficient method for P2P shortest path problem on social network. Our method is based on community structure, which could classify the edges into two parts according to cost. Then we construct community graph so as to narrow down the scope of SP and search the SP efficiently. A suitable shortest community path number  $k$  could guarantee the approximation error. Extensive experimental results on five real-world networks of collaboration

networks demonstrated the efficiency and approximation error. For given limit, our algorithm has excellent performance. In particular, various community detection methods could be used in this algorithm. In the future, we will focus on the development of a heuristic method for combining social network community information for SP problem in weighted social networks. Otherwise, it will be interesting to study the community detection algorithm suited to the social network.

#### References

- [1] J. Scott, P.J. Carrington, *The SAGE Handbook of Social Network Analysis*, SAGE publications, 2011.

- [2] J. Xie, B.K. Szymanski, X. Liu, Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, in: Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, IEEE, 2011, pp. 344–349.
- [3] M.E. Newman, Phys. Rev. E 64 (1) (2001) 016131.
- [4] M. Potamias, F. Bonchi, C. Castillo, A. Gionis, Fast shortest path distance estimation in large networks, in: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, 2009, pp. 867–876.
- [5] M.E. Newman, Phys. Rev. E 64 (1) (2001) 016132.
- [6] G.R. Jagadeesh, T. Srikanthan, K. Quek, Intelligent Transp. Syst. IEEE Trans. 3 (4) (2002) 301–309.
- [7] S. Jung, S. Pramanik, Knowl. Data Eng. IEEE Trans. 14 (5) (2002) 1029–1046.
- [8] H.A. Karimi, J. Intelligent Transp. Syst. 3 (2) (1996) 111–127.
- [9] B. Liu, IEEE Trans. 27 (4) (1997) 436–448.
- [10] P. Sanders, D. Schultes, Highway hierarchies hasten exact shortest path queries, in: Algorithms—Esa 2005, Springer, 2005, pp. 568–579.
- [11] L. Fu, D. Sun, L.R. Rilett, Comput. Operations Res. 33 (11) (2006) 3324–3343.
- [12] M. Girvan, M.E. Newman, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
- [13] M.E. Newman, Phys. Rev. E 69 (6) (2004) 066133.
- [14] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Proc. Natl. Acad. Sci. U S A 101 (9) (2004) 2658–2663.
- [15] M.E. Newman, M. Girvan, Phys. Rev. E 69 (2) (2004) 026113.
- [16] X. Xu, N. Yuruk, Z. Feng, T.A. Schweiger, Scan: A structural clustering algorithm for networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 824–833.
- [17] Y.-Y. Ahn, J.P. Bagrow, S. Lehmann, Nature 466 (7307) (2010) 761–764.
- [18] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, J. Stat. Mech. Theory Exp. 2008 (10) (2008) 10008.
- [19] J. Huang, H. Sun, J. Han, H. Deng, Y. Sun, Y. Liu, Shrink: A structural clustering algorithm for detecting hierarchical communities in networks, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 219–228.
- [20] M.-G. Gong, L.-J. Zhang, J.-J. Ma, L.-C. Jiao, J. Comput. Sci. Technol. 27 (3) (2012) 455–467.
- [21] M. Gong, J. Liu, L. Ma, Q. Cai, L. Jiao, Phys. A Stat. Mech. Appl. 403 (2014) 71–84.
- [22] J.Y. Yen, Manag. Sci. 17 (11) (1971) 712–716.
- [23] M.E. Newman, Phys. Rev. E 74 (3) (2006) 036104.
- [24] M.E. Newman, Proc. Natl. Acad. Sci. 98 (2) (2001) 404–409.
- [25] <http://arxiv.org/archive/astro-ph>.
- [26] <http://arxiv.org/archive/cond-mat>.
- [27] <http://arxiv.org/archive/nucl-ex>.
- [28] E.W. Dijkstra, Numer. Math. 1 (1) (1959) 269–271.
- [29] M.L. Fredman, R.E. Tarjan, J. ACM (JACM) 34 (3) (1987) 596–615.