云南大学学报(自然科学版),2015,37(6):825~831 DOI:10.7540/j.ynu.20150129
Journal of Yunnan University

统计网络 Motif 方法的研究与设计*

汪海涛, 唐华阳, 姜 瑛
(昆明理工大学 云南省计算机技术应用重点实验室,云南 昆明 650500)

摘要:在分析挖掘一个网络中的信息时,一个非常重要的信息就是统计 Motif.现有算法是将原始网络在给定的条件下进行边与顶点转换,再从转换后的网络中找出所有子图,如果子图不满足 Motif 的要求则删除,存在时间复杂度过高的问题.针对这种情况,提出了一种自底向上的剪枝算法,在不需要经过网络转换的前提下,首先找到最小的符合要求的子图,再推导出更大的子图,而且所找到的每个子图均满足 Motif 的要求.并通过时间效率分析得出,对于该问题而言,提出的算法优于现有的算法,具有一定的理论研究价值.

关键词:复杂网络;自底向上;子图同构;Motif;剪枝
中图分类号:TP 301.52 **文献标志码**:A **文章编号**:0258-7971(2015)06-0825-07

在复杂性网络分析时,通常会从中挖掘出各种有用的信息,例如:典型的当前路径长度(characteristic temporal path length),当前全局功效(temporal global efficiency),拓扑重叠(topological overlap),当前聚集度(temporal clustering)等.其中一个非常重要的信息就是 Motif,也就是一类子图的个数.

网络 Motif(Network Motifs)源于文献[1],定义为:在复杂网络中发现的某种相互连接的模式个数显著高于随机网络.在几年的发展后,定义中提到的模式主要有:①拓扑结构相同的子图;②拓扑结构及权重大小顺序相同的子图等.

对于不带权重的网络来说,就是统计所有拓扑结构相同的子图的个数,例如:

在图 1 中,(b)和(c)都是(a)的子图,而且(b)和(c)都是相同的拓扑结构,所以属于同一类子图,且(a)中该类子图的个数为 5.

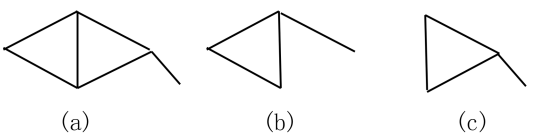


图 1 不带权重的网络
Fig.1 The network without weight

而对于带权重的静态网络来说,就是除了满足拓扑结构相同外,还需满足子图中的权重连续.在举例子前先引入 2 个概念:

(1) 相邻边是指有公共顶点的边;

(2) 图连续则是指对于图中任意两边 e1,e2,均可找到一个相邻边序列 $ei,\cdots,ej$,使得 ei 与 e1 相邻,ej 与 e2 相邻,而且该序列中任意 2 个相邻的边的权重差均小于给定的值 Δt.

例如,在图 2 中,(b)和(c)都是(a)的子图,如果定义 $\Delta t=4$,此时虽然(b)和(c)的拓扑结构相同,而且 2 个子图都是连续的,但对应边上的权重大小顺序不同,所以不是同类子图.

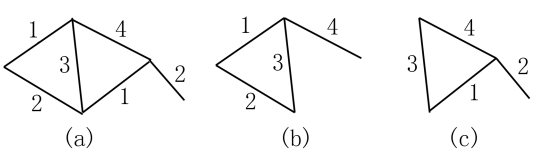


图 2 带权重的网络
Fig.2 The network with weight

对于带权重的动态网络而言,也就是每个边上带

* **收稿日期**:2015-03-15
基金项目:国家自然科学基金(61462049).
作者简介:汪海涛(1967-),女,云南人,副教授,研究方向:软件工程.E-mail:kmwht@163.com.

有多个相同度量的权重,也可以将 Motif 定义为一类拓扑结构和连续性相同的子图.但是统计 Motif 的计算复杂度按照以前的方法将会大大增加.

1 传统算法分析

现有的关于统计 Motif 比较实用软件只有计算静态网络的,例如文献[2-3]中介绍的 fanmod,采用了文献[4]中提到的 ESU 算法,而该算法是基于 ESU-tree 结构均匀抽样的,具有随机性.而且能够统计的 Motif 的度最大只能达到 8.对于动态网络,该软件也无能为力.

通常动态网络是指包含时间权重的复杂网络,在统计包含时间权重的网络 Motif 中,文献[5]提到一种寻找满足边连续的同构子图的自定向下思想,也就是先找到最大的有效子图,在从该子图中寻找其他更小的有效子图.例如图 3,对于(a)而言,如果定义,则其中一个最大有效子图是(b),再通过(b)找到更小的有效子图(c),(d),(e),(f)等.其具体步骤如下:

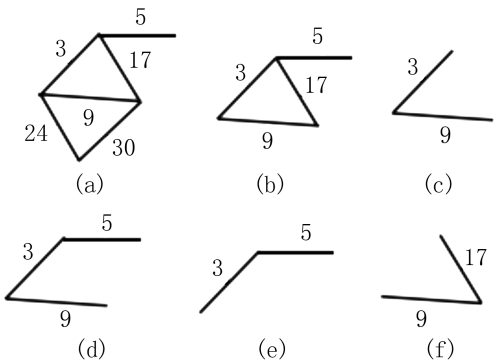


图 3 自顶向下统计 Motif

Fig.3 Counts the Motif obey top-down

步骤 1 构建时间权重网络 G;

步骤 2 将 G 中具有公共顶点的相邻边权重差值小于 Δt 的边视为邻接边;将 G 中边转换为顶点构成新的网络 H,连接 H 中具有邻接边性质的顶点,如图 4 进行边点转换;

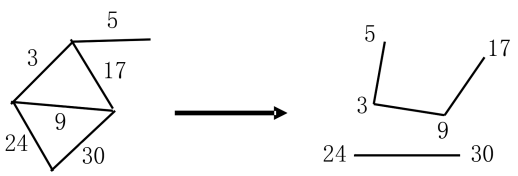


图 4 图转换

Fig.4 Transform graph

步骤 3 从 H 中找到所有最大的连接子图 Emax;

步骤 4 找到 Emax 中所有有效子图 E;

步骤 5 标识相应的 Motif.

在文献[2]中提到:所有的 Motif 一定是 H 的子图,但是 H 的子图不一定是 Motif.所以需要从 H 中找到最大的连接子图后再找更小的有效子图.

该算法巧妙地将原始网络利用 Δt 进行转换,将边转换为顶点,满足 Δt 相邻边的公共顶点转换为边.不过在该算法的步骤 4 中,实际上需要先找到 Emax 的所有子图,再判断这些子图是否满足 Motif 的条件,满足则是有效子图 E.遍历所有的子图是一件非常困难的事情,尤其是在规模很庞大的网络中,该算法根本就不能实际使用.仔细分析该算法,不难发现,可以不用做网络转换,更不需要找出所有的子图.实际上,如果就基于原始网络进行搜索,整个搜索空间就一个树结构,一旦发现某个子图不是有效子图时,相应分支以后的所有子图都可以不用搜索了.本文也就基于这个思想对该算法进行改进,提出一种自底向上的剪枝搜索算法,以提高算法的效率,降低时间复杂度.

2 2 个重要性质

在介绍本文提出的算法之前,先阐述 2 个重要的性质.

(1) 由 n 个点组成的子图可以由至少 1 个 $n-1$ 个点的子图导出.

例如:如果某网络中存在 5 个点的子图(图 5 左边(a)),那么在找 4 个点的子图时,一定可以找到图 5 右边的子图(b),且属于图 5 左边的子图(a),那么在进一步扩展时,很容易可以找到图 5 左边 5 个点的子图(a).

(2) 由 n 个点组成且边数为 m 的子图可以由至少一个由 n 个点组成且边数为 $m-1$ 的子图导出.

例如:如果某网络中存在 5 个点且边数为 6 的子图(图 6(a)),那么在找 5 个点且边数为 5 的子图时,

一定可以找到图 6(b),且属于图 6(a)的子图,那么同样在进一步扩展时,很容易可以找到图 6 左边的子图(a).

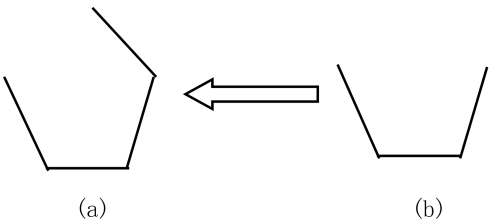


图 5　点推导

Fig.5 Deduce of nodes

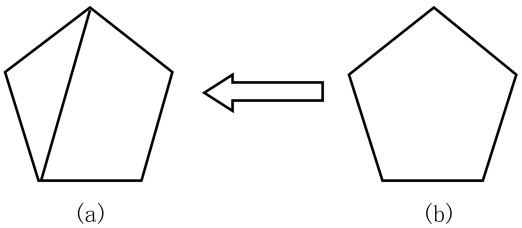


图 6　边推导

Fig.6 Deduce of edges

3　新算法研究

分析发现,有了以上 2 个性质还不够,以上 2 个例子对于找所有的 Motif 有些缺陷.例如在第 1 例中,如果网络中存在图 5 中左边(a)的情况的话,那么我们在找 4 个点的子图时可以通过它找到 2 个不同的子图.因为任意去掉一端的边后都可以看作一个 4 个点的子图,如图 7 中 2 个子图均是图 5 中(a)的子图.然而这 2 个子图扩展后得到却是同一个 5 个点的子图(图 5 左边子图(a)).所以在算法设计中必须去除这样的情况.

对于这种情况的处理方式:可以考虑把每种找到的子图用一个矩阵存下来,并且把对应的顶点以及每边的时间都建立一个相应的链表存储下来.当找到一个新的子图时就与之前的进行比较.如果重复,则摒弃当前的新子图;否则将新子图插入链表中.

还存在另一种重复的问题:如图 8,右边的 2 个子图(b)、(c)都能导出同一个子图(a).但是这 2 个子图不属于同类 Motif.对于这个问题,仔细观察会发现,图 8 右边 2 个子图(b)、(c),尽管顶点相同以及邻接矩阵同构,但是连接各顶点的边有差别,权重大小顺序不同,子图(b)中左边权重为 3 的边大于两相邻边的权重,而子图(c)中右边权重为 1 的边小于两相邻边的权重.所以在搜索到新的子图后,发现顶点相同及邻接矩阵同构时,可以考虑检查任意两个顶点之间是否有边,以及边上的时间是否相同,如果对应的任意两个顶点之间的边情况都相同,则认为是重复;否则不同.

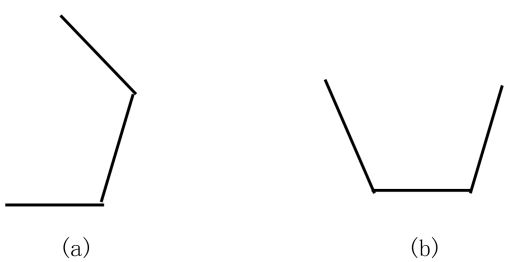


图 7　拓扑结构相同的子图

Fig.7 The same of topological structure of sub-graph

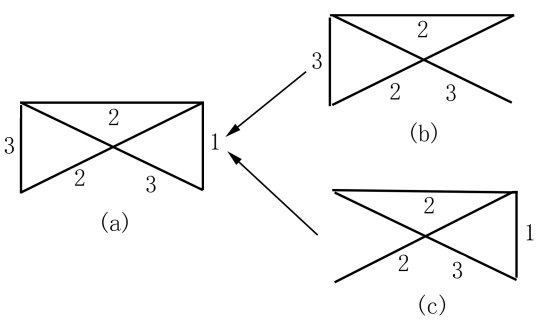


图 8　边推导缺陷

Fig.8 The drawback of deduce of edges

通过以上分析,在采用以上 2 个性质找子图时,主要的问题就是剔除重复的情况,而重复的情况存在 2 种:

(1) 由不同的"小"子图导出相同的"大"子图时,可能存在重复;

(2) 在网络中存在"大"子图,找"小"子图时,可能存在邻接矩阵同构,顶点也相同,但是属于两个不同的 Motif.

在分析出现重复的过程中,已经介绍的相应的解决方案.

经过以上分析后,只需要找到最简单、最小、最基本的 Motif,即可推导出所有的 Motif.而最基本的 Motif 即只有 1 个事件,或者说只有 2 个顶点 1 条边的子图.因此本文提出了如下算法:

主体框架:

步骤 1 初始化最基本的 Motif 队列;

步骤 2 取出 Motif 队列的第 1 个元素,并记为 MotifFirst;

步骤 3[点推导] 遍历 MotifFirst 中的每个顶点,如果该点能够扩展,且扩展之后的点不在 Motif 队列中,则将扩展的点插入 Motif 中;

步骤 4[边推导] 遍历 MotifFirst 中的每个顶点,如果该点还能与 Motif 中其它的点构成边,且该边不属于 MotifFirst,则将扩展的点插入 Motif 中;

步骤 5 如果 Motif 队列不为空,则跳转步骤 2;否则结束.

下面以图 3 中网络(a)为例来说明该算法的执行过程.为了方便说明,将原图上加上一些字符,如图 9.

经过初始化程序,得到最基本的 Motif,见表 1.

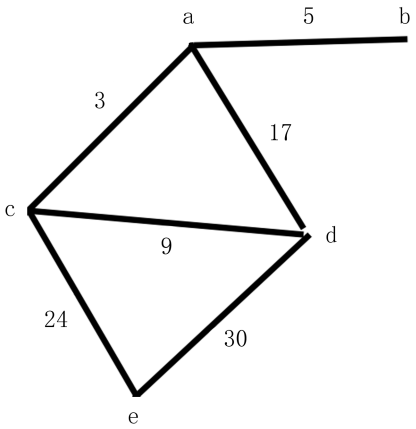


图 9 网络范例

Fig.9 Ainstance of network

表 1 最基本的 Motif

Tab.1 Most fundamental Motif

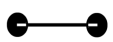
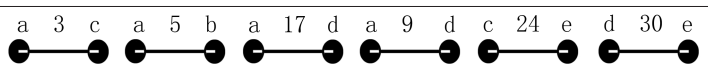
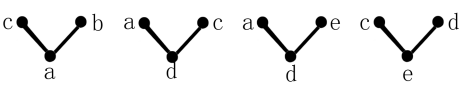
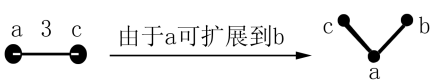
步骤	Motif	频数	列表
初始化		6	
点推导		4	
边推导		0	Null

表 1 中,初始化步骤,得到 6 个最基本的 Motif,在通过最基本的 Motif 进行“点推导”,例如:



但是当用初始化 Motif 队列第 2 元素进行“点推导”时,尽管此时点 a 可以扩展到 c 但是扩展之后得到的子图和之前初始化 Motif 队列第 1 个元素进行“点推导”时得到的结果重复.而在进行边推导时,没有一个子图可以进行边扩展,所以得到的是 Null 列表.当列表为空时,程序结束.

为了方便分析时间复杂度,这里假设为完全图.假设网络结点数为 N,边数为 M.

现有的算法由于需要找到转换图的所有子图,则转换图的顶点数为 M,边数由权重而定,假设顶点数为 i 的子图边数为 E_i.顶点数为 i 的子图数为

$$\mathrm{Num}_i = \begin{cases} 1, & i = M, \\ \prod_{j=1}^{M-i} (M - j + 1), & i \neq M. \end{cases}$$

则顶点数为 i 的所有子图的时间复杂度为:

$$T_i=\begin{cases}O(\mathrm{Num}_i), & \text{边数为 } E_i,\\ O\Big(\mathrm{Num}_i\prod_{j=1}^{E_i-i}*E_i-j+1)\Big), & \text{其他},\end{cases}$$

则总的时间复杂度为 $O\Big(\sum_{i=1}^{M}T_i\Big)$.

本文提出的算法由于不需要进行图转换,所以不需要找出所有子图,最差情况下,也就是不做任何剪枝,时间复杂度为:

$$O\Big(\sum_{i=2}^{M}\Big(\prod_{j=1}^{C_i^2-i}(C_i^2-j+1)\prod_{j=1}^{i}(N-i+1)\Big)\Big).$$

可以比较在最差的情况下,优于现有算法,而实际应用中往往会做大量的剪枝,所以本算法时间复杂度远小于现有的算法,即:

$$O\Big(\sum_{i=2}^{N}\Big(\prod_{j=1}^{C_i^2-i}(C_i^2-j+1)\prod_{j=1}^{i}(N-i+1)\Big)\Big)\ll O\Big(\sum_{i=1}^{M}T_i\Big).$$

4　实验设计及结果分析

由于本文提出的算法在整个搜索过程中,进行了大量的剪枝,所用时间更少,在以上的时间复杂度分析中证明了这一点,并且在以下产生各种随机网络的实验验证中也证明了这一点.

4.1　实验设计　采用 2 种方法来对比算法的好坏,即:①在不同顶点数的网络进行 Motif 挖掘对比算法效率;②在顶点数相同边数不同的网络中对比算法效率.

4.1.1　*当网络规模不同时对比算法效率*　随机产生各种大小的随机网络,再使用本文提出的算法对这些网络的 Motif 特征进行挖掘,同时也使用之前的网络 Motif 挖掘算法对相同的随机网络进行挖掘.由于对 Motif 的挖掘时间不仅取决于网络的大小,网络中各顶点的连接情况以及各边的权重都是非常重要的影响因素.所以,对每种大小的随机网络的计算时间取平均值,用平均值来比较以前的算法和本文的算法.

4.1.2　*当网络规模相同时,随着边数的增加来对比算法效率*　也即是,固定网络的顶点个数,增加网络中的边数,因为网络中边的数据对挖掘的影响非常大.同样也是采取对比平均时间.

4.2　实验实施　本实验的运行环境为:软件环境 Linux, gcc; 硬件环境 Intel(R) Core(TM) i5-4210M;内存 8G.

4.2.1　*通过增大网络的规模进行实验*　从结点数为 3 的网络到结点数为 9 的网络各随机产生 20 个网络.然后再使用本文提出的算法和之前的算法对这些网络进行 Motif 特征挖掘.计算结果见表 2.

表 2　2 种算法的计算结果(增大网络的规模)

Tab.2 The result of two algorithms(extend the scale of network)

算法 结点数	本文的剪枝算法 (pruning)	传统的边点转换算法 (convert edge-node)
3	0.006 15	0.028 1
4	0.010 15	0.017 25
5	0.021 95	0.882 75
6	0.120 4	—
7	0.183 55	—
8	1.318 9	—
9	34.188 25	—

从表 2 中可以看出,当结点数超过 5 时,传统的边点转换算法已经不能计算出 20 个随机网络,或者说时间很长,也就是传统的算法指数增长的速度非常大.而本文提出的算法在结点数为 9 时的网络 Motif 特征挖掘中平均时间也只有 34.18 s.更加直观的结果对比见图 10.从该图中可以看出,2 种算法在网络规模很小时,时间消耗上没有什么差别,主要的差别是在结点数为 5 开始,再继续往后扩大网络规模时,传统的算法已经不能应付了,而本文提出的算法还是能够在使用者能够接受的范围内运算着.

4.2.2 相同的结点数中,增加网络中的边数进行实验

由于传统的算法在顶点数大于 5 时已经不能够应付,所以该试验中固定顶点网络的顶点数为 5,产生边数从 2 到 8 的随机网络 20 个来进行算法对比.计算结果见表 3.

从表 3 中可以看出,当增加网络的边数时,本文提出的算法也明显优于传统的算法,更明显的对比见图 11.

4.3 实验结论 通过实验表明本文提出的算法明显优于现有算法,也证实了本文算法的时间复杂度远小于现有算法的时间复杂度.

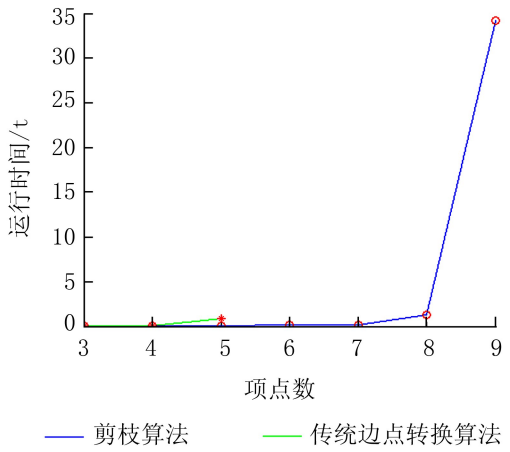


图 10 2 种算法的结果对比

Fig.10 Compare the result of two algorithms

5 总 结

在复杂网络分析中,统计 Motif 时,传统的软件、算法或者计算能力有限,或者算法效率不够高.本文提出的算法基于自底向上的剪枝算法,效率上优于以前的算法,对于不太复杂的网络,完全可以将所有种类的 Motif 都统计出来.在文献[2]中提到的算法,由于需要找出所有的子图,所以时间花销随着网络复杂度的增加成指数增长.而本文提出的算法,通常情况下会剪掉很多分支,所以相对先前的算法在时间开销上降低了很多.通过数据实验分析以及时间复杂度分析,明显地对比出,本文提出的算法优于传统的算法,具体提高的倍数与实际遇到的网络有关.

表 3 2 种算法的计算结果(增大网络的边数)

Tab.3 The result of two algorithms(add the edge of network)

算法 结点数	本文的剪枝算法 (pruning)	传统的边点转换算法 (convert edge-node)
2	0.001 2	0.002 7
3	0.004 8	0.003 4
4	0.005 6	0.006 7
5	0.008 9	0.006 9
6	0.018	0.314 6
7	0.012	0.062 6
8	0.084	4.994 7

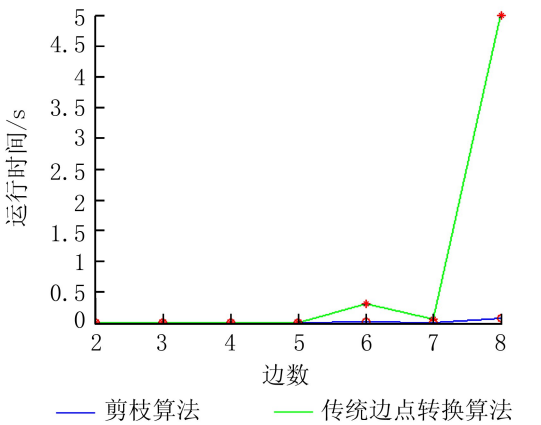


图 11 当网络顶点个数为 5 时,随着边数的增加对比 2 种算法的运行时间

Fig.11 Compare the running time of two algorithms with add edge of network when the number of node of network is 5

参考文献:

[1] QIN G M,GAO L.An algorithm for network motif discovery in biological networks[J].Data Mining and Bioinformatics,2012,6(1):1-16.

[2] RASCHE F,WERNICKE S.FANMOD fast network motif detection-manual[EB/OL].(2014-04-22).http://wenku.baidu.com/link url = SGJeqI02sTpwewaea4ZHmcd9TTisDlKhoPHDjEGimFrmaqSlSNgZx dwWXtBGYCSAhw1FO6 _ BwS4BY2UmfxYSsfmNeskZuFfoRr2CwqqTyQi.

[3] WERNICKE S.A faster algorithm for detecting network motifs[C].Proc 5th WABI-05 in LNBI,2005,3692:165-177.

[4] MILO R,SHEN-ORR S,LTZKOVITZ S,et al.Network motifs:Simple building blocks of complex networks[J].Science,2002,298(5594):824-827.

[5] KOVANEN L,KARSAI M,KASKI K,et al.Temporal motifs in time-dependent networks[J].Journal of Statistical Mechanics Theory Experiment,2011(11):1 293-1 307.

[6] WERNICKE S,RASCHE F.FANMOD:a tool for fast network motif detection[J].Bioinformatics,2006,22(9):1 152-1 153.

[7] WONG E,BAUR B,QUADER S,et al.Biological network motif detection: principles and practice[J].Briefings in Bioinformatics,2012,13(2):202-215.

[8] OMIDI S,SCHREIBER F,MASOUDI-NEJAD A.MODA:An efficient algorithm for network motif discovery in biological networks[J].Genes Genet.Syst,2009,84:385-395.

[9] 覃桂敏,高琳,呼加璐.生物网络模体发现算法研究综述[J].电子学报,2009,37(10):2 258-2 265.
QIN G M,GAO L,HU J L,et al.A review on algorithms for network motif discovery in biological networks[J].Acta Electronica Sinica,2009,37(10):2 258-2 265.

[10] 覃桂敏,高琳,周晓锋.非树型网络模体发现算法[J].电子学报,2009,37(11):2 420-2 426.
QIN G M,GAO L,ZHOU X F.Non-Treelike network motif detection algorithm[J].Acta Electronica Sinica,2009,37(10):2 420-2 426.

Research and design of a statistical way for Motif in Network

WANG Hai-tao, TANG Hua-yang, JIANG Ying

(Key Laboratory of Computer Technology Application of Yunnan Province,Kunming University of Science and Technology,Kunming 650500,China)

Abstract:Motif is a very important information when analysis and mining for network.The exiting algorithm for it is used to transform the edge and vertex in original network under given conditions,and then,find all subgraph and get rid of the subgraph that don't follow the Motif require.The new algorithm in this paper is a pruning approach based bottom-up rather than don't using the transformation.First we find all minimum subgraph that follow the Motif require,and then deduce other big subgraph.All subgraph that find use the new algorithm is follow Motif require.And through time efficiency analysis concluded that for this problem,the new algorithm is superior to existing algorithms,has some theoretical research value.

Key words:complex network;bottom-up;subgraph isomorphism;Motif;pruning