

基于社区划分与连边逆序放回的网络分解算法

王志晓^{1,2}, 张 磊^{1,4}, 孙成成¹, 芮晓彬¹, 黄珍珍^{1,3}, 张孙贤¹

(1. 中国矿业大学计算机学院, 江苏徐州 221116; 2. 教育部矿山数字化工程研究中心, 江苏徐州 221116;
3. 中国矿业大学图书馆, 江苏徐州, 221116; 4. 中国矿业大学徐海学院, 江苏徐州 221008)

摘要: 网络分解是通过删除网络中最少规模的节点或者连边, 将网络破坏至最大连通分支的规模不超过设定阈值. 传统基于节点删除的网络分解算法忽略了删除代价. 实际上, 节点的删除导致相应连边的删除, 代价是不同的. 传统基于连边删除的网络分解算法虽然考虑删除代价, 但是, 无论是迭代计算连边中心性值, 还是迭代划分最大连通分量, 其性能和效率都亟待改善. 本文提出了一种基于社区划分与连边逆序放回的网络分解算法, 该算法是一种基于连边删除的方法, 包含两个步骤, 首先, 利用社区划分算法将网络划分为多个社区, 删除社区之间的全部连边使社区独立, 破坏社区间的连通性; 然后, 每个社区内部采用连边逆序放回策略破坏其内部连通性, 从而完成整个网络的分解. 真实网络及人工网络上的实验结果表明: 一方面, 本文提出的网络分解算法能够以最小的连边删除代价将网络分解至设定阈值; 另一方面, 随着网络规模、网络结构以及分解阈值的变化, 算法展现出良好的稳定性.

关键词: 社交网络; 网络分解; 删除代价; 社区划分; 连边逆序放回; 网络连通性

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112(2022)03-0540-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20201201

Network Dismantling Algorithm Based on Community Detection and Inverse Reinsertion of Edges

WANG Zhi-xiao^{1,2}, ZHANG Lei^{1,4}, SUN Cheng-cheng¹, RUI Xiao-bin¹,
HUANG Zhen-zhen^{1,3}, ZHANG Sun-xian¹

(1. College of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China;
2. Mining Digital Engineering Research Center of Ministry of Education, Xuzhou, Jiangsu 221116, China;
3. Library of China University of Mining and Technology, Xuzhou, Jiangsu 221116, China;
4. College of Xuhai, China University of Mining and Technology, Xuzhou, Jiangsu 221008, China)

Abstract: Network dismantling aims to find the minimal set of nodes or edges that, if removed, will break the network into small components and the scale of the giant connected component shall not exceed the pre-defined threshold. Traditional node-deleting based methods ignore the cost of deletion. In fact, when we delete a node, the corresponding edges linked to this node should also be deleted. The cost is different. Although traditional edge-deleting based methods take the cost into consideration, performance and efficiency need to be further improved. This paper proposes an edge-deleting based network dismantling algorithm, which contains two stages: community detection and inverse reinsertion of edges. In the first stage, the whole network is divided into different communities by using community detection algorithm and then edges between communities are removed to destroy the connectivity of communities. In the second stage, the strategy of inverse reinsertion of edges is used to destroy the connectivity within each community. Thus, we can dismantle the whole network into pieces. Experiment results on real-world and artificial networks show that, on one hand, our proposed method can dismantle networks by removing a smaller set of edges than that of other state-of-the-art methods. On the other hand, our proposed method exhibits stable performance with the variation of network scale, network structure and the threshold of network dismantling.

Key words: social network; network dismantling; deletion cost; community detection; inverse reinsertion of edge; network connectivity

1 引言

影响力最大化^[1,2]和网络分解^[3]是社交网络分析领域两个重要的研究分支,前者通过选择种子节点进行信息传播,使得最终的影响范围最大.后者则通过删除网络中最小规模的节点或连边,将网络破坏成一系列小规模、不连通的分支.网络分解可以应用于舆情控制^[4],蛋白质结构解析以及交通线路防护^[5,6]等领域.

网络分解算法主要有两类:基于节点删除的网络分解算法^[3,4,7,8]和基于连边删除的网络分解算法^[9,11].前者利用中心性指标迭代删除网络中的重要节点,直到网络中最大连通分支^[7](Giant Connected Components, GCC)的规模小于事先设定的阈值.大多数研究将该阈值设定为整个网络规模的0.01^[7,8].现有基于节点删除的网络分解方法大多不考虑删除代价. Ren 等人^[4]指出网络分解过程中每个节点的删除代价是不同的,节点度值越大,连边越多,其删除代价越大.基于连边删除的网络分解方法是一类考虑删除代价的方法,该方法迭代删除影响网络连通性的关键连边,直到网络中最大连通分支的规模小于事先设定的阈值.大多数此类研究也将阈值设定为0.01^[9,11].基于连边删除的网络分解可以进一步细分为基于边中心性指标的方法^[9]和基于最大连通分量划分的方法^[11].基于连边删除的网络分解方法存在以下不足:(1)基于边中心性指标的方法需要在整个网络范围内计算每条连边的中心性值,复杂性较高,不适合大规模网络;(2)基于最大连通分量划分的方法需要迭代划分网络中的最大连通分支,连边删除缺乏针对性,容易删除一些不必要的连边.

社区划分技术已经成功应用于社区结构识别^[12,13]以及蛋白质复合物识别^[14,15]等诸多领域.在网络社区结构中,社区内节点间的连边稠密,不同社区间的连边稀疏.如果先利用社区划分技术对网络进行社区划分,就可以区分出社区内的连边和社区间的连边,针对两类连边的特点分别进行处理,从而达到快速分解网络的目的.鉴于此,本文将社区划分技术与连边逆序放回策略相结合,提出了一种基于连边删除的网络分解算法 CD-IRE (Network Dismantling Based on Community Detection and Inverse Reinsertion of Edges).首先,利用社区划分算法挖掘网络的社区结构.社区划分不但能将整个网络划分为多个小规模社区,而且能有效识别影响社区间连通性的连边,删除这些连边即可将网络分解为多个独立的社区.然后,采用逆序放回策略进行社区内的分解:先移除社区内的所有连边,接着依次放回使社区内部 GCC 规模增长最小的连边,直到放回任何一条连边都会使社区内 GCC 规模超过设定阈值.最后,删除那些未被放回的连边,即可完成社区分解,进而实现整个网络的分解.逆序放回策略能准确识别

并删除对社区内 GCC 增长贡献最大的连边,从而有效破坏社区内部的连通性.

真实网络上的实验结果表明,相比较于其他典型网络分解算法,本文方法删除较小规模的连边就能将网络分解至设定阈值;人工网络上的实验结果表明,随着网络规模的变化、网络结构的变化以及 GCC 阈值的变化,本文方法均表现出较强的稳定性.

2 算法描述

首先,挖掘网络的社区结构,删除社区之间的所有连边,使网络分解成一个个互不相连的局部区域.然后,采用逆序放回策略分解每个社区,最终实现对整个网络的分解.

2.1 社区划分

Louvain^[16]是一种基于模块度的社区划分方法,具有准确性高、复杂度低等优点.本文借鉴 Louvain 算法思想完成社区结构挖掘.首先,将网络中的每个节点都分配到不同的社区中,对每一个节点 i ,计算将它加入到其邻居节点所在社区所带来的模块度变化,加入模块度增益最大的社区.若加入所有邻居节点所在社区的模块度增益均为0,则节点 i 保持在原来的社区内.重复上述操作,直到网络社区结构不再变化为止.划分社区后,连接不同社区的连边清晰地显露出来,删除这些连边即可以最小的代价快速地将整个网络分解为一个个小规模局部区域,从而极大地破坏整个网络的连通性.

如图1所示,划分出的社区 C_1 和 C_2 内部的连边非常密集,两个社区则通过边 E 连接在一起.仅仅删除边 E 就可以彻底破坏社区 C_1 和 C_2 间的连通性,边 E 即是非常强连通性的连边^[9],起连接两个连边密集社区的作用.

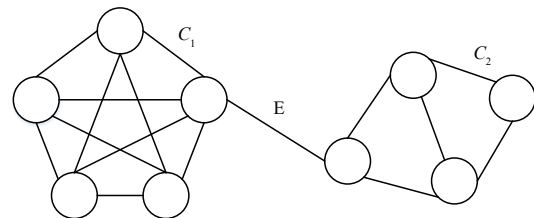


图1 社区间的连边示意图

2.2 社区内部连边逆序放回

破坏社区间的连通性后,需要进一步破坏每个社区内部的连通性才能完成整个网络的分解.本文采用一种逆向策略来解决这一问题,即连边逆序放回 IRE (Inverse Reinsertion of Edges),该策略依次放回对社区内连通性影响最小的连边直到 GCC 超过事先设定的阈值.

连边逆序放回策略的步骤如下:

步骤1: 将社区 i 内部的连边全部删除放入集合

$E^0(i)$ 中,社区中仅剩下一个个互不相连的孤立节点;

步骤2:从集合 $E^0(i)$ 中选择一条使社区内 GCC 规模增长最小的连边放回社区;

步骤3:若存在多条连边放回后都能使 GCC 规模增长最小,则从这些连边中选择两端所连节点度值之和最小的连边放回社区;

步骤4:若满足步骤3的连边仍然有多条,则从这些连边中选择两端所连节点度值之差绝对值最小的连边放回社区;

步骤5:重复步骤1~4,直到从 $E^0(i)$ 中选择任何一条连边放回原社区都会使社区内 GCC 的规模大于设定阈值.

步骤6:删除 $E^0(i)$ 集合中的剩余连边,完成社区内的分解.

上述逆序放回策略尽可能地将位于社区边缘、连

接节点度值较小,且对社区内部连通性影响小的连边保留下来,从而最大限度地删除处于中心位置、连接节点度值较大、对社区内部连通性影响较大的连边.

下面举例说明逆序放回策略的执行过程.图2状态1是一个节点规模为400的网络划分完社区之后得到的一个社区,该社区包含6个节点和10条连边,每条连边旁边的数字表示该连边的编号. GCC的阈值设定为0.01. 采用逆序放回策略的第一步是将全部连边放入 E^0 ,此时,放回任何一条连边 GCC 规模的增长都相同,因此,按照步骤3选择两端所连节点的度值之和最小的连边,即编号为1的连边,将其首先放回. 以此类推,后面依次放回编号为4,7,3,8,9,10的连边. 此后,如果继续放回,则 GCC 的规模将大于设定的阈值0.01,放回过程结束. 此时, E^0 中剩余的连边为{2,5,6},予以删除,从而完成对该社区的分解,见图2状态2. 算法描述如算法1所示.

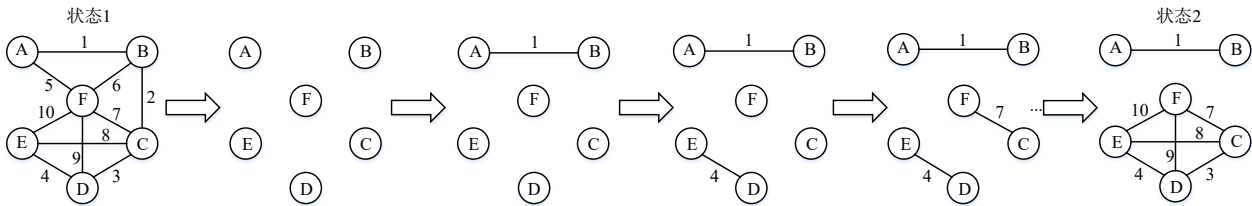


图2 连边逆序放回过程示意图

2.3 算法时间复杂度分析

本文所提出的网络分解算法包含两个阶段:社区划分和连边逆序放回. 社区划分阶段采用 Louvain 算法将整个网络划分为 k 个社区,删除社区间的所有连边,时间复杂度是 $O(N+N')$, N 是网络的节点规模, N' 是迭代节点的个数. 逆序放回阶段,完成单个社区 i 分解的时间复杂度是 $O(m_i N_i)$, m_i 表示社区 i 内的连边数量, N_i 代表社区 i 内的节点数量. 在最坏情况下, k 个社区都需要进行分解,那么,完成 k 个社区连边逆序放回的时间复杂度就是 $O(\sum_{i=1}^k (m_i N_i))$. 这样一来,整个算法的时间复杂度为

$$O(N+N') + O(\sum_{i=1}^k (m_i N_i)). \quad (\text{其中 } \sum_{i=1}^k (m_i) \leq m, \text{ 当且仅当 } k=1 \text{ 时 } \sum_{i=1}^k (m_i) = m, \sum_{i=1}^k (N_i) = N.)$$

3 实验与结果分析

3.1 数据集和评价指标

实验所用数据包括真实网络和人工网络. 真实网络如表1所示, n 代表网络节点规模,从几百到几万不等. m 代表网络连边总数, c 代表网络节点的平均连边数(平均度值). 人工网络有两类,一类是 ER(Erdős-Rényi) 人工网络,节点规模分别是 2000、4000、6000 和 10000,对于

表1 真实网络数据集

Network	n	m	c
Corruption	309	3281	21.24
Crime	754	2127	5.64
Political-Blogs	1222	16714	27.36
Petster-hamster	2000	16098	16.10
PPI	2224	6609	5.94
Power-Grid	4941	6594	2.67
Autonomous-Systems	6474	12572	3.88
Authors	21363	93439	8.74

给定规模的 ER 网络,平均连边数(平均度值) c 的变化区间为 $[3, 10]$. 另一类是 BA(Barabási-Albert) 人工网络,节点规模最小是 1000,最大是 10000,对于给定规模的 BA 网络,平均连边数(平均度值) c 的取值为 4、6 和 8. 真实网络来源于 <http://konect.uni-koblenz.de/>,评价指标包括最大连通分支规模 GCC 和网络分解代价 Cost.

GCC 评价指标定义如式(1).

$$GCC = \frac{N_{GCC}}{n} \quad (1)$$

其中, N_{GCC} 是最大连通分支 GCC 中的节点数量, n 是网络节点总数.

代价 Cost 表示完成网络分解需要删除的连边数量占整个网络连边总数的比例:

算法 1 基于社区发现与连边逆序放回的网络分解算法

输入:网络 $G(V, E), \theta$

/*网络 G 为非时序网络, V 为节点集合, E 为连边集合. 不失一般性, 这些连边无权、无向*/

/* θ 为 GCC 的阈值, 一般为 0.01*/

输出:网络分解需要删除的连边集合 E^d

(1) $\{ \text{community_detection}(G) \rightarrow C, E^c ;$

/*对网络进行社区划分并识别社区之间的连边, 得到社区集合 C 和社区之间的连边集合 E^c */

(2) $\text{delete_edges}(E^c) \rightarrow E^d ;$

/*删除社区之间的连边 E^c , 并放入集合 E^d 中, E^d 代表需要删除边的集合*/

(3) $i=1 ;$

/* i 为社区集合 C 中每个社区的编号, 最小值为 1, 最大值为社区总数, 即 $\text{size}(C)$ */

(4) $\text{while}(\text{giant_connected_component}(C(i)) > \theta \ \&\& \ i \leq \text{size}(C(i)))$

/* $C(i)$ 表示编号为 i 的社区, $\text{giant_connected_component}(C(i))$ 为该社区中最大连通分支的规模*/

(5) $\{ \text{delete_edges}(C(i)) \rightarrow E^0(i) ;$

/*将社区 $C(i)$ 中的所有连边删除并放入集合 $E^0(i)$, 此时, 社区 $C(i)$ 仅剩一个个互不相连的节点, 最大连通分支的规模为 $1/\text{size}(C(i))$ */

(6) $\text{while}(\text{giant_connected_component}(C(i)) \leq \theta)$

(7) $\text{IRE}(E^0(i), C(i)) ;$

/*根据连边逆序放回策略 IRE, 从 $E^0(i)$ 中选择一条连边放回至原社区 $C(i)$. 重复此过程, 直至再放回任何一条连边至原社区 $C(i)$, 该社区中最大连通分支的规模都将超过阈值 θ */

(8) $E^0(i) \rightarrow E^d ;$

/*将 $E^0(i)$ 中剩余的连边加入集合 E^d 中. $E^0(i)$ 中剩余的连边表示分解当前社区需要删除的连边*/

(9) $i++ ; \}$

(10) $\text{out} \rightarrow E^d$

(11) $\}$

/*删除集合 E^d 中的连边即可实现对网络 G 的分解*/

$$\text{Cost} = \frac{dl}{m} \quad (2)$$

其中, dl 是分解过程中被删除的连边数量, m 是网络的连边总数.

3.2 真实网络实验结果分析**3.2.1 与基于节点删除的网络分解方法性能对比**

实验中选取 5 种基于节点删除的网络分解算法, 包括 GND (Generalized Network Dismantling) 算法^[4]、GNDR (Generalized Network Dismantling with Reinsertion) 算法^[4]、EGP (Equal Graph Partitioning) 算法^[17]、Min-Sum 算法^[3] 和 BPD (Belief Propagation-guided Decimation) 算法^[7]. Min-Sum 和 BPD 是典型的基于去环的方法, 相关参数均按照原文的最优值进行设置: Min-Sum 算法第一个阈值 $C_1=0.5\%$, 第二个阈值 $C_2=1\%$. BPD 权重参数 $x=12$.

首先, 分析最大连通分支规模 GCC 随删除代价 Cost 增大的变化情况. 图 3 显示了所选算法在 Crime 和 Power-Grid 网络上的实验结果, 横坐标代表网络分解代价 Cost, 纵坐标代表最大连通分支规模 GCC. 从图 3 可以看出, 随着网络分解代价 Cost 的增大 (删除连边的数量增多), 最大连通分支规模 GCC 呈现下降趋势. 相对于其他 5 种算法, CD-IRE 算法删除较小规模的连边就能最大限度地破坏网络的连通性, 性能最优. 以 Crime 网络为例, CD-IRE 算法仅花费不到 0.1% 的代价就可以将网络分解至 GCC 小于 0.1, GND 算法实现这个目标需要 0.2% 的代价, 其余 4 种算法的代价均超过 0.5. CD-IRE 算法在 Power-Grid 网络上的这一优势表现得更为突出.

然后, 进一步分析达到网络分解阈值 (即 0.01) 时不同网络分解算法需要的删除代价 Cost. 表 2 显示了上述 6 种算法分解 Crime、Corruption、Petster-hamster、Power-Grid、Political Blogs 和 Autonomous Systems 网络所需要的代价 Cost. 可以看出, 本文的 CD-IRE 算法只需要删除较小规模的连边就可以完成对 6 个网络的分解, 表现出最好的性能. GNDR 算法也表现出了不错的性能, BPD 算法紧随其后. 另外, 从表 2 可以看出, 无论何种算法, 分解 Corruption 网络和 Political Blogs 网络的代价 Cost 都超过 0.92, 原因是这两个网络节点的平均连边数 (平均度值) 非常大, 分别达到了 21.24 和 27.36 (见表 1), 从而导致网络分解的难度较大.

表 2 CD-IRE 算法与基于节点删除的网络分解方法 Cost 对比

算法	Crime	Corruption	Petster-hamster	Power-Grid	Political Blogs	Autonomous Systems
CD-IRE	0.5232	0.9487	0.7489	0.1554	0.9200	0.5222
GND	0.6422	0.9829	0.9142	0.4209	0.9900	0.9700
GNDR	0.5853	0.9747	0.8130	0.2583	0.9800	0.8010
EGP	0.9123	0.9813	0.9233	0.9423	0.9900	0.9122
Min-Sum	0.6282	0.9802	0.8537	0.3788	0.9800	0.7824
BPD	0.5913	0.9703	0.8329	0.2929	0.9800	0.7813

3.2.2 与基于连边删除的网络分解方法性能比较

实验中选取 4 种基于连边删除的网络分解算法, 分别是 Ncut 算法^[11]、Bond percolation 算法^[18]、Betweenness 算法^[10] 和 Bridgeness 算法^[9].

首先, 分析最大连通分支规模 GCC 随删除代价 Cost 增大的变化情况. 图 4 显示了所选算法在 Autonomous-Systems 和 Petster-hamster 网络上的实验结果. 随着 Cost 的增加, CD-IRE 算法的 GCC 值快速下降, 这一现象与图 3 中的结果类似, 说明删除社区间的连边可以极大地破坏整个网络的连通性. Ncut 算法采用了谱平分思想, 每次迭代将网络近似分为大小相等的两部分, 因此 Ncut 算法也能使 GCC 值快速下降. CD-IRE 算

法删除当前社区的全部外部连边后才会更新 GCC 的值,因此会导致 GCC 值的更新略微滞后. 总体而言,

CD-IRE 算法和 Ncut 算法性能接近,明显优于 Bond percolation 算法、Betweenness 算法和 Bridgeness 算法.

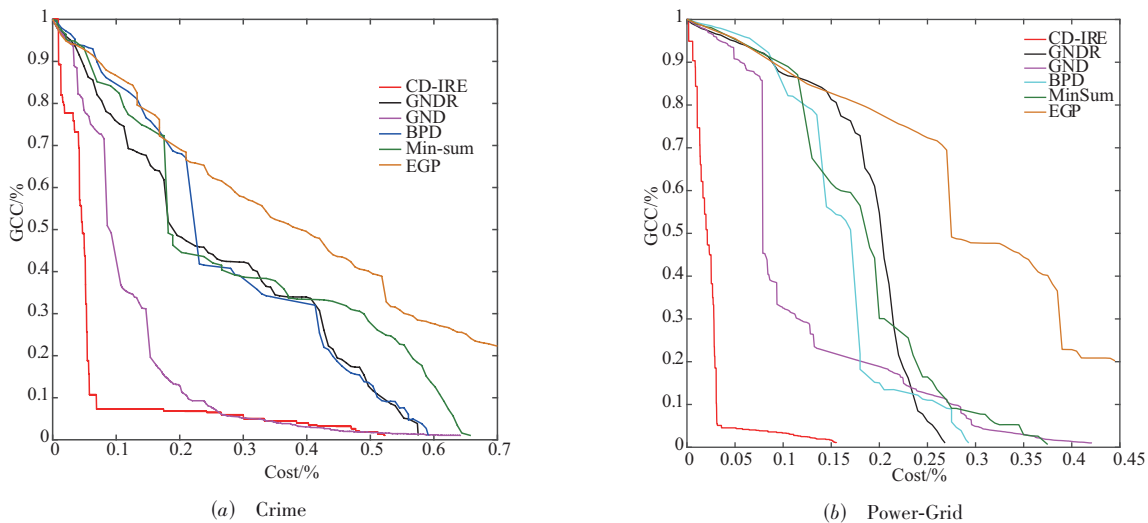


图3 CD-IRE算法与基于节点删除的网络分解方法性能对比

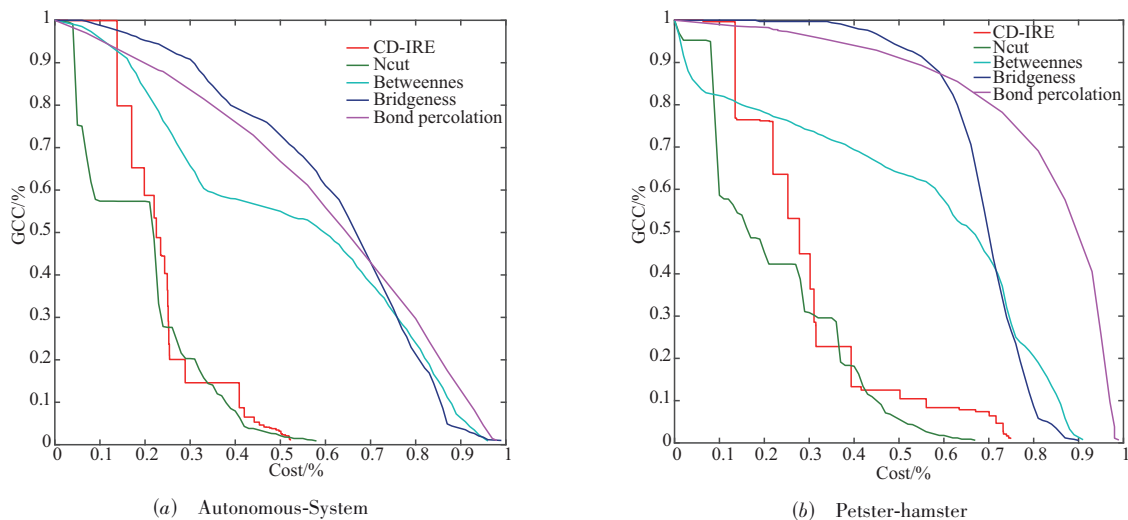


图4 CD-IRE算法与基于连边删除的网络分解方法性能对比

然后,进一步分析达到网络分解阈值0.01时不同网络分解算法需要的删除代价 Cost. 表3显示了上述5种算法分解 Crime、Corruption、Petster-hamster、Power-Grid、Political Blogs 和 Autonomous Systems 网络所需要的代价 Cost. CD-IRE 算法在 Crime、Corruption、Political Blogs 和 Autonomous Systems 等4个网络上表现出最好的性能,在 Petster-hamster 和 Power-Grid 网络上的性能仅次于 Ncut 算法,这一结果与图4中对 CD-IRE 算法和 Ncut 算法性能的分析一致.

3.2.3 GCC 阈值对网络分解算法性能的影响

此处选取上述实验中性能突出的 CD-IRE 算法、

GNDR 算法、Min-Sum 算法、BPD 算法和 Ncut 算法,分析 GCC 阈值对网络分解算法性能的影响. 选取的真实网络包括 PPI、Power-Grid 和 Authors. GCC 的阈值分别设定为 0.2、0.4、0.6 和 0.8.

表4~表6分别展示了所选算法在 PPI、Power-Grid 和 Authors 网络上的 Cost 值. 可以看出,随着 GCC 阈值变小,网络分解标准逐渐提高,绝大多数算法的 Cost 呈现增大趋势. 在 PPI 网络中,CD-IRE 算法和 Ncut 算法性能相当,优于其他算法. 在 Power-Grid 网络中,CD-IRE 算法最优,GNDR 算法紧随其后. Ncut 算法以 0.0302 的 Cost 代价直接将 Power-Grid 网络的 GCC 降至 0.2 以下,因

表3 CD-IRE 算法与基于连边删除的网路分解方法 Cost 对比

算法	Crime	Corruption	Petster-hamster	Power-Grid	Political Blogs	Autonomous Systems
CD-IRE	0.5232	0.9487	0.7489	0.1554	0.92	0.5222
Ncut	0.5646	0.9753	0.6713	0.1213	0.93	0.5751
Btweenness	0.9031	0.9834	0.9188	0.8827	0.99	0.9712
Bond percolation	0.9756	0.9903	0.9912	0.8733	0.99	0.9823
Bridgeness	0.9812	0.9883	0.9022	0.9889	0.99	0.9906

此0.2以上的阈值变化对该算法没有影响(见表5). 总体而言,随着GCC 阈值的变化,CD-IRE 算法始终能够保持良好的性能,以较小的代价完成网路分解.

3.3 人工网路实验结果分析

3.3.1 ER 网路

此处选取上述实验中性能突出的 CD-IRE 算法、GND 算法、GNDR 算法和 Ncut 算法,分析网路规模和结构特性对其性能的影响. ER 网路的节点规模分别

取 2000、4000、6000 和 10000,平均连边数(平均度值) c 从 3 变化到 10,步长为 1. GCC 阈值为 0.01. 考虑到所生成人工网路的随机性,实验取 48 次结果的平均值.

图 5 显示了 ER 网路上的相应结果. 在不同的网路规模下,随着平均连边数(平均度值) c 的增大,网路分解的代价 Cost 逐渐增加. 很显然,节点邻居数量变多增大了网路分解的难度. 但是,无论节点规模和平均连边数(平均度值) c 如何变化,CD-IRE 算法和 Ncut 算法均

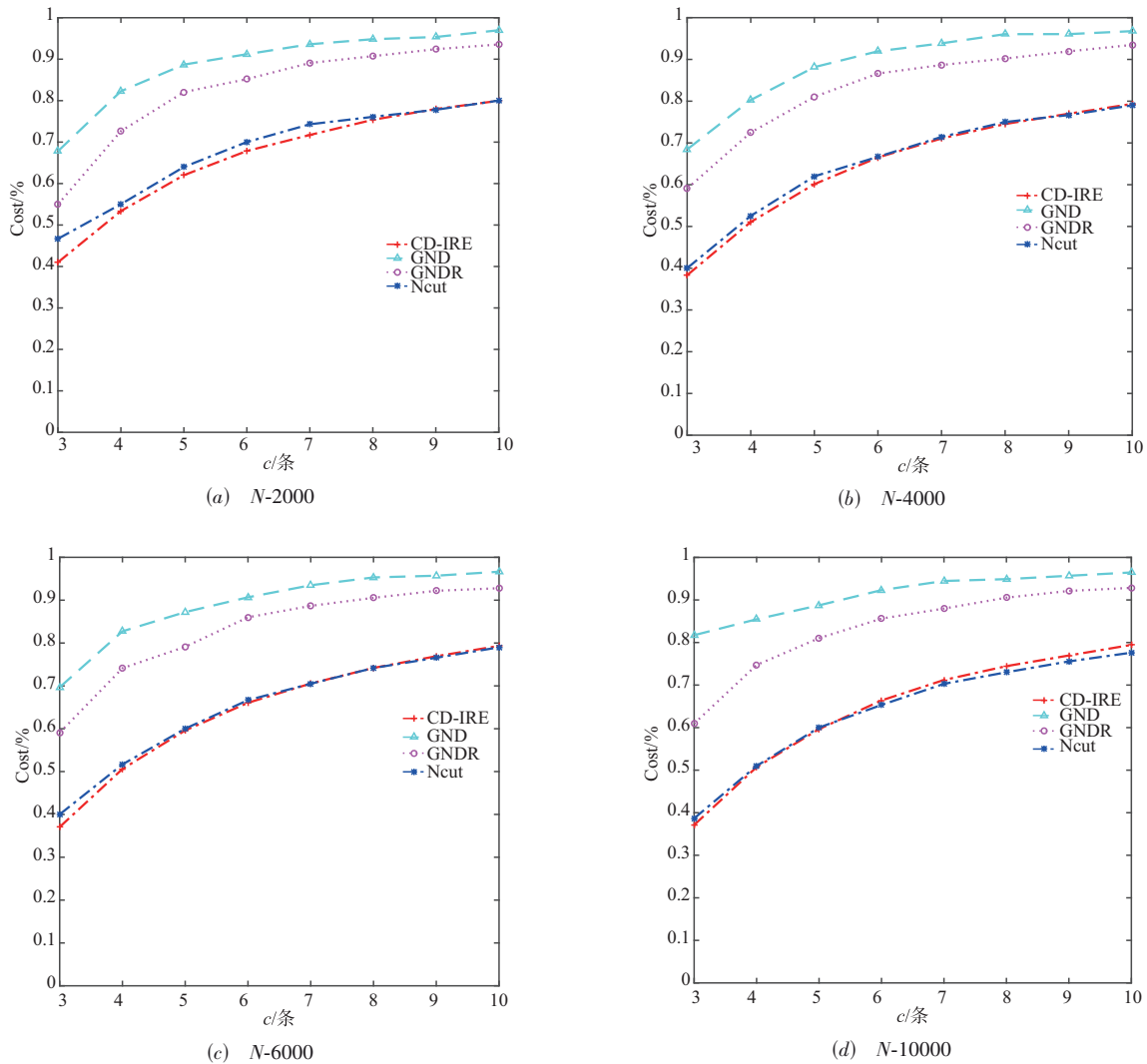


图5 ER 人工网路上的实验结果

表现出良好的性能, 优于其他算法. 另外, 从这两种算法曲线的位置变化可以看出, 它们对网络规模不敏感.

表 4 GCC 阈值对网络分解算法性能的影响(PPI 网络)

GCC	Cost				
	Min-sum	BPD	GNDR	CD-IRE	Ncut
0.8	0.3166	0.3304	0.1202	0.1783	0.0908
0.6	0.6159	0.6725	0.2914	0.2625	0.2118
0.4	0.7415	0.7079	0.4944	0.3061	0.3126
0.2	0.7606	0.7515	0.6448	0.3426	0.3631

表 5 GCC 阈值对网络分解算法性能的影响(Power-Grid 网络)

GCC	Cost				
	Min-sum	BPD	GNDR	CD-IRE	Ncut
0.8	0.0175	0.0636	0.0148	0.0108	0.0302
0.6	0.0175	0.0636	0.0148	0.0167	0.0302
0.4	0.0468	0.0677	0.0394	0.0253	0.0302
0.2	0.0912	0.0905	0.1507	0.0291	0.0302

表 6 GCC 阈值对网络分解算法性能的影响(Authors 网络)

GCC	Cost				
	Min-sum	BPD	GNDR	CD-IRE	Ncut
0.8	0.4339	0.4356	0.1144	0.2463	-
0.6	0.5383	0.5288	0.3057	0.2956	-
0.4	0.5807	0.5724	0.4609	0.3227	-
0.2	0.5973	0.5953	0.5051	0.3574	-

3.3.2 BA 网络

BA 网络的节点规模在 1000 至 10000 间递增, 每次增加 1000 个节点. 平均连边数(平均度值) c 分别取 4、6 和 8. 考虑到 Ncut 算法和 CD-IRE 算法性能最接近, 此处仅分析这两种算法在 BA 网络上的性能.

同样地, 图 6 显示了 48 组实验的平均值. CD-IRE- c_4 中的“ c_4 ”表示平均连边数(平均度值)为 4, 其余曲线标识的含义类似. 可以看出, 两种算法在 BA 网络上的性能非常稳定. 当平均连边数(平均度值) c 为 6 和 8 时,

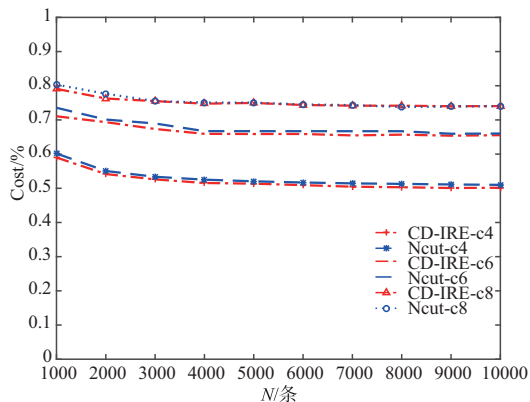


图 6 BA 人工网络上的实验结果

CD-IRE 算法的性能优于 Ncut 算法. 当平均连边数(平均度值) c 为 4 时, 两个算法的性能相当.

4 结论

网络分解是社交网络分析中的重要问题, 其研究成果在很多领域都有着广泛应用, 包括控制计算机病毒在通信网络中的传播范围、干预谣言在社交网络中的扩散程度等^[19,20]. 本文提出了基于社区发现与连边逆序放回的网络分解算法, 该算法借助社区划分将网络分解为若干个局部社区, 删除社区间的连边, 破坏社区间的连通性. 然后, 采用逆序放回策略, 以最小的代价进一步破坏社区内部的连通性, 最终完成对整个网络的分解. 实验结果表明, 本文算法以删除较小规模的连边为代价, 就能将网络分解至事先设定阈值. 并且, 随着网络规模的变化、网络结构的变化以及设定阈值的变化表现出较强的稳定性.

参考文献

- [1] 朱恩强, 吴艳蕾, 许宇光, 等. 基于树核度的社交网络影响最大化问题[J]. 电子学报, 2019, 47(01): 161-168.
ZHU E Q, WU Y L, XU Y G, et al. Tree-coreitvity-based influence maximization in social networks[J]. Acta Electronica Sinca, 2019, 47(01): 161-168. (in Chinese)
- [2] WANG N, SUN Q, ZHOU Y, et al. A study on influential user identification in online social networks[J]. Chinese Journal of Electronics, 2016, 25(3): 467-473.
- [3] BRAUNSTEIN A, DALL'ASTA L, SEMERJIAN G, et al. Network dismantling[J]. Proceedings of the National Academy of Sciences, 2016, 113(44): 12368-12373.
- [4] REN X L, GLEINIG N, HELBING D, et al. Generalized network dismantling[J]. Proceedings of the National Academy of Sciences of the United States of America, 2019, 116(14): 6554-6559.
- [5] CAO Y, SUN Y K, MA L C. A fault diagnosis method for train plug doors via sound signals[J]. IEEE Intelligent Transportation Systems Magazine, 2020, 13(3): 107-117.
- [6] CAO Y, WANG Z C, LIU F, et al. Bio-inspired speed curve optimization and sliding mode tracking control for subway trains[J]. IEEE Transactions on Vehicular Technology, 2019, 68(7): 6331-6342.
- [7] MUGISHA S, ZHOU H J. Identifying optimal targets of network attack by belief propagation[J]. Physical Review E, 2016, 94(1): 012305.
- [8] MORONE F, MAKSE H A. Influence maximization in complex networks through optimal percolation[J]. Nature, 2015, 524(7563): 65-68.

- [9] CHENG X, REN F, SHEN H, et al. Bridgeness: a local index on edge significance in maintaining global connectivity [J]. *Physics*, 2010, 10(5): 10011.
- [10] LÜ L, CHEN D, REN X L, et al. Vital nodes identification in complex networks[J]. *Physics Reports*, 2016, 650: 1-63.
- [11] REN X L, GLEINIG N, TOLÍCD, et al. Underestimated cost of targeted attacks on complex networks[J]. *Complexity*, 2018, 2018: 2826243.
- [12] 汪林玉, 谷科, 余飞, 等. 基于个人意愿的社会网络团体结构与信息检测方案[J]. *电子学报*, 2019, 47(04): 886-895.
WANG L Y, GU K, YU F, et al. Social community structure and information detection scheme based on personal willingness[J]. *Acta Electronica Sinica*, 2019, 47(04): 886-895. (in Chinese)
- [13] WANG F F, ZHANG B H, CHAI S C. Deep auto-encoded clustering algorithm for community detection in complex networks[J]. *Chinese Journal of Electronics*, 2019, 28(3): 489-496.
- [14] 雷秀娟, 高银, 郭玲. 基于拓扑势加权的动态 PPI 网络复合物挖掘方法[J]. *电子学报*, 2018, 46(01): 145-151.
LEI X J, GAO Y, GUO L. Mining protein complexes based on topology potential weight in dynamic protein-protein interaction networks[J]. *Acta electronica Sinica*, 2018, 46(01): 145-151. (in Chinese)
- [15] WANG X S, CHEN Y H, SUN W F. Identification of overlapping protein complexes using structural and functional information of ppi network[J]. *Chinese Journal of Electronics*, 2015, 24(3): 564-568.
- [16] BLONDEL V D, GUILLAUME J L, LAMBIOTTE R, et al. Fast unfolding of communities in large networks[J]. *Journal of Statistical Mechanics: Theory and Experiment*, 2008, 2008(10): P10008.
- [17] CHEN Y, PAUL G, HAVLIN S, et al. Finding a better immunization strategy[J]. *Physical Review Letters*, 2008, 101(5): 058701.
- [18] CALLAWAY D S, NEWMAN M E J, STROGATZ S H, et al. Network robustness and fragility: percolation on random graphs[J]. *Physical Review Letters*, 2000, 85(25): 5468-5471.
- [19] ZDEBOROVÁ L, ZHANG P, ZHOU H J. Fast and simple decycling and dismantling of networks[J]. *Scientific Reports*, 2016, 6(1): 37954.
- [20] RICHARD J L, DONALD J R, ROBERT E T. Generalized nested dissection[J]. *SIAM Journal on Numerical*

Analysis, 1979, 16(2): 346-358.

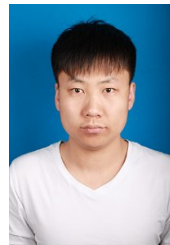
作者简介



王志晓 男, 1979 年生于河南叶县, 博士中国矿业大学计算机学院教授、博士生导师。主要研究方向为社交网络分析与数据挖掘。在国内外重要学术期刊与会议上发表学术论文 30 余篇。
E-mail: zhxwang@cumt.edu.cn



张磊 男, 1982 年生于江苏泰兴, 中国矿业大学计算机学院博士研究生。主要研究方向为社交网络分析和机器学习。
E-mail: zhangleixp@126.com



孙成成 男, 1993 年生于山东邹城, 中国矿业大学计算机学院博士研究生, 主要研究方向为社会网络分析、图表示学习和数据挖掘。
E-mail: scc@cumt.edu.cn