

控制与决策

Control and Decision

基于有限节点集的网络毁伤最大化问题研究

刘凤增, 肖兵, 金宏斌, 李浩

引用本文:

刘凤增, 肖兵, 金宏斌, 等. 基于有限节点集的网络毁伤最大化问题研究[J]. *控制与决策*, 2020, 35(4): 937–942.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2018.0966>

您可能感兴趣的其他文章

Articles you may be interested in

基于能耗区域感知的无线传感器网络路由算法

Energy balance routing algorithm based on forward-aware for wireless sensor networks

控制与决策. 2019, 34(7): 1425–1432 <https://doi.org/10.13195/j.kzyjc.2017.1700>

无线传感器网络栅栏覆盖改进

Improving barrier coverage in wireless sensor networks

控制与决策. 2019, 34(5): 1037–1042 <https://doi.org/10.13195/j.kzyjc.2017.1511>

割点失效对复杂网络可控性的影响

Effect of cut vertexes-removal on controllability of complex networks

控制与决策. 2019, 34(11): 2310–2316 <https://doi.org/10.13195/j.kzyjc.2019.0132>

UCUBG:基于等级划分的水下传感器网络非均匀分簇算法

UCUBG: An uneven clustering algorithm for UWSNs based on grading

控制与决策. 2019, 34(1): 89–96 <https://doi.org/10.13195/j.kzyjc.2017.0999>

基于IMM-IKF的无线传感器网络非视距节点定位方法

Non-line of sight node localization method based on IMM-IKF for wireless sensor networks

控制与决策. 2018, 33(6): 1069–1074 <https://doi.org/10.13195/j.kzyjc.2017.0173>

WSNs中基于期望网络覆盖和分簇压缩感知的数据收集方案

Data collection scheme based on expected network coverage and cluster compressive sensing for WSNs

控制与决策. 2018, 33(3): 422–430 <https://doi.org/10.13195/j.kzyjc.2017.0029>

基于人群搜索优化的无线传感器网络三点定位算法

Three points localization algorithm based on seeker optimization algorithm for wireless sensor networks

控制与决策. 2017, 32(8): 1518–1522 <https://doi.org/10.13195/j.kzyjc.2016.0781>

基于粒子群优化的无线传感器网络非视距节点定位算法

Non-line of sight node localization algorithm based on particle swarm optimization for wireless sensor networks

控制与决策. 2015(6): 1106–1110 <https://doi.org/10.13195/j.kzyjc.2014.0616>

基于有限节点集的网络毁伤最大化问题研究

刘凤增^{1,2†}, 肖兵¹, 金宏斌¹, 李浩¹

(1. 空军预警学院 预警情报系, 武汉 430019; 2. 国防科技大学 信息通信学院, 武汉 430010)

摘要: 对网络实施攻击时,人们希望在有限的资源下获得最大的毁伤效果,而节点排序策略并不能实现毁伤最大.针对这种情况,定义攻击有限节点集的网络毁伤最大化问题,并给出问题的近似求解算法.由于近似求解算法计算复杂度较高,进一步提出基于重要节点的贪婪算法(greedy algorithm based on important nodes, GABIN).对无标度网络的实验表明:GABIN 算法能够有效地减少计算时间,且效果接近于近似求解算法;当无标度网络的度指数 $\gamma \geq 2.5$ 时,GABIN 算法的效果明显优于排序算法,所得节点集中超过 30% 的节点不同于排序算法.对 Power 网络的毁伤实验表明,GABIN 算法适用于较大规模的实际网络,且效果显著优于度、介数、接近度、删除节点等排序算法.实验发现,利用 GABIN 算法获得的关键节点集包含大量的非中心性节点,这为网络攻击或网络防护提供了一个新的思路.

关键词: 毁伤最大化; 有限节点集; 节点重要性; 贪婪算法; 复杂网络; 计算复杂度

中图分类号: TP393

文献标志码: A

Network damage maximization based on finite node set

LIU Feng-zeng^{1,2†}, XIAO Bing¹, JIN Hong-bin¹, LI Hao¹

(1. Department of Early-Warning Intelligence, Air Force Early-Warning Academy, Wuhan 430019, China; 2. College of Information and Communication, National University of Defense Technology, Wuhan 430010, China)

Abstract: When attacking the network, people hope to get the maximum damage effect under the limited resources, and the node sorting strategy cannot achieve the maximum damage. In view of this situation, the network damage maximization problem of attacking a finite node set is defined, and an approximate algorithm for solving the problem is given. Due to the high computational complexity of the approximate algorithm, a greedy algorithm based on important nodes (GABIN) is proposed. Experiments on scale-free networks show that the GABIN algorithm can effectively reduce computing time, and its effect is close to the approximate algorithm. When the degree exponent of scale-free networks is greater than or equal to 2.5, the GABIN algorithm is better than sorting algorithms, and more than 30% of the nodes in the node set are different from sorting algorithms. The damage experiments on the Power grid show that the GABIN algorithm is suitable for large-scale real networks, and its effect is significantly better than sorting algorithms such as degree, betweenness, closeness and deleting nodes. The experiment shows that the key node set obtained by using the GABIN algorithm contains a large number of non-central nodes, which provides a new idea for network attack or network protection.

Keywords: damage maximization; finite node set; node importance; greedy algorithm; complex network; computational complexity

0 引言

复杂网络存在于人们生活中的各个方面,如国际互联网、移动通信网、电网等.同时,作为刻画复杂系统相互作用的重要工具,很多复杂系统,如交通系统、军事作战体系等,也被抽象为复杂网络来进行研究^[1].发掘网络中的关键节点或关键节点集,对于提升网络防护能力或网络攻击效果具有重要的现实意

义^[2].已有的工作大多是利用各种中心性指标或度量方法对网络节点进行排序,如度中心性指标、接近度中心性指标^[3-4]、介数中心性指标^[5]、特征向量中心性指标^[6]、 k 核指标^[7]、删除节点法^[8]等,然后根据排序结果从前到后依次对网络节点实施攻击.在节点攻击代价均等的情况下,当仅能攻击 K 个节点时,按照上述策略选取的关键节点集并不能使网络毁伤最大

收稿日期: 2018-07-13; 修回日期: 2018-11-21.

基金项目: 国家自然科学基金项目(61502522).

责任编辑: 李忠奎.

[†]通讯作者. E-mail: fengzeng_liu@126.com.

化^[9]. 包含 K 个节点的最关键节点集并不等于排序靠前的 K 个节点组成的集合^[10]. 在节点攻击代价不均等的情况下, 度排序策略的毁伤效果甚至不如平均度策略^[11]. 文献[12]从网络攻防博弈的视角研究了实现毁伤最大化的攻击策略选择问题, 文献[13]研究了不完全信息条件下的最优攻击策略问题, 这些工作大多是从给定的候选策略集中或特定条件下进行寻优. Deng等^[14]基于禁忌搜索方法从策略全集中寻找近似最优解, 取得了更好的网络毁伤效果.

在社会网络研究中, 对于 K 节点集的影响最大化问题已有较为深入的研究^[15]. 参考节点影响力最大化问题, 针对仅能攻击 K 个节点的情况, 定义了有限节点集下的网络毁伤最大化问题. 由于该问题是一个NP难的组合优化问题, 最优解算法和近似求解算法都存在计算复杂度高、难以应用于大规模网络的困难^[16]. 为了避免求最优解的计算困境, 并考虑工程可行性, 提出一种基于重要节点的贪婪算法. 基于模型网络 and 实际网络的实验表明, GABIN算法能够有效降低计算复杂度, 效果接近于近似求解算法, 显著优于按排序选取节点的算法.

1 网络毁伤最大化问题

1.1 问题定义

假设复杂网络 $G = (V, E)$ 是由 $|V| = N$ 个节点和 $|E| = M$ 条边组成的无自环简单网络, $V = \{v_1, v_2, \dots, v_N\}$ 和 $E = \{e_1, e_2, \dots, e_M\}$ 分别是节点和边的集合. 网络的邻接矩阵为 $A = \{a_{ij}\}_{N \times N}$, 其中 $a_{ij} = 1$ 表示节点 v_i 和 v_j 之间有边连接, 否则 $a_{ij} = 0$. 网络毁伤最大化问题可定义如下.

给定一个复杂网络 G 和毁伤节点数量 K , 从网络 G 中寻找一个节点集合 S^* , 使得

$$S^* = \arg \max_{S \subseteq V, |S|=K} f(S), \quad (1)$$

其中 $f(S)$ 为目标函数, 表示从网络 G 中删除节点集 S 后网络能力的下降值. 网络能力的度量指标可以有多种, 这里采用网络效率^[17] 指标. 对于节点 v_i 和 v_j , 节点间的距离 d_{ij} 越短, 其传输效率越高. 一对节点间的传输效率可表示为 $1/d_{ij}$, 整个网络的效率可表示为所有节点对效率的平均值, 即

$$\varepsilon = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}}, \quad (2)$$

其中 $d_{ij} \in [1, +\infty]$, 即节点间的最短距离为1, 两节点不连通时距离为无穷大. 显然, ε 的取值范围为 $[0, 1]$, 当 $\varepsilon = 1$ 时, 网络 G 为全连通图, 效率最高; 当 $\varepsilon = 0$ 时, 网络 G 由孤立节点构成, 效率最差.

为使网络效率指标在网络结构变动时具有可比性, 在进行网络节点删除操作时, 只断开节点间的连边, 使其成为孤立节点. 令 G' 表示网络 G 删除节点集 S 后的拓扑结构, 则目标函数可定义为网络效率下降的比例, 即

$$f(S) = 1 - \varepsilon(G')/\varepsilon(G). \quad (3)$$

1.2 近似求解

网络毁伤最大化问题可归纳为从 N 个节点中找出使目标函数最大化的 K 个节点, 这是一个NP难的组合优化问题. 当 N 足够大时, 以当前的计算能力几乎是无法找到最优解的, 因此只能近似求解. Nemhauser等^[18]的工作证明了利用贪婪算法能够获得近似解. 若 \hat{S} 是由贪婪算法求得的节点集, 则有

$$f(\hat{S}) \geq (1 - 1/e)f(S^*), \quad (4)$$

其中 e 为自然底数. 在 S^* 无法求得的情况下, 可利用贪婪算法求出近似解, 其目标函数的值可达到最优值的63%以上. 贪婪算法的设计思想是每次选择毁伤力最大的节点加入节点集, 其计算过程如下.

算法1 贪婪算法.

输入: 网络 G , 节点个数 K ;

输出: 节点集 S .

step 1: 初始化 $S = \{\}, i = 1$;

step 2: $v = \arg \max_{v_j \in V-S} (f(S \cup \{v_j\}) - f(S))$;

step 3: $S = S \cup \{v\}$;

step 4: 如果 $i < K$, 令 $i = i + 1$, 返回 step 2;

step 5: 输出 S .

由贪婪算法的计算过程可知, 每选择一个节点都要遍历剩余节点计算网络效率. 利用广度优先搜索算法计算网络效率的复杂度为 $O(N(N+M))$, 则贪婪算法的计算复杂度约为 $O\left(\frac{K}{2}(2N-K+1)N(N+M)\right)$. 当 $K \ll N$ 时, 贪婪算法的计算复杂度约为 $O(KN^2(N+M))$, 当网络规模较大时, 贪婪算法难以适用.

2 基于重要节点的贪婪算法

现实中大多数复杂网络具有节点异质性, 比如在无标度网络中, 其度分布具有幂律性, 少数节点的度值较高, 多数节点的度值较低. 一般认为, 攻击度值较大的节点能够获得更大的网络毁伤. 因此, 可利用重要性排序算法对节点进行筛选, 缩小候选节点的范围, 从而提高贪婪算法的计算效率.

2.1 重要节点筛选

采用度指标和介数指标对节点进行重要性排序, 依据排序筛选出重要节点. 度是衡量节点重要性的

一个直观指标,定义为邻居节点的数量,节点 v_i 的度表示为

$$k_i = \sum_{j=1, j \neq i}^N a_{ij}. \quad (5)$$

度值低的节点对于网络效率 ϵ 的贡献较低,如删除 $k_i = 1$ 的节点 v_i 时,网络仅遭受直接损失,而不会受到间接损失. 直接损失为节点 v_i 到其他节点之间的最短距离变为无穷大,使得 $\sum_{j \neq i} \frac{1}{d_{ij}} = 0$. 间接损失是指当节点 v_i 位于其他节点对的最短路径上时,删除节点 v_i 后导致其他节点对之间的最短路径变长,从而降低节点对之间的效率. 由于 $k_i = 1$, 节点 v_i 不会出现在其他节点对的最短路径上,不会引起间接损失,其入选节点集 S 的可能性也较小. 因此,可基于度指标对节点进行筛选.

仅仅依靠度指标有可能引起误筛选,度低的节点对网络效率也可能有大的贡献. 如图1所示,对于桥节点 v ,其度值仅为2,但删除节点后网络变为两个不连通的分支,网络效率大幅下降. 针对这一情况,引入介数指标进行修正.

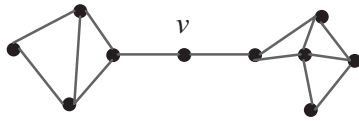


图1 桥节点示意图

节点 v_i 的介数是网络中所有最短路径中经过节点 v_i 的数量,记为 b_i ,表示为

$$b_i = \sum_{j \neq k \in V} \frac{\sum_{l \in P_{jk}} \delta_l^{v_i}}{|P_{jk}|}. \quad (6)$$

其中: P_{jk} 表示节点 v_j 和 v_k 的最短路径集合, $|P_{jk}|$ 表示 P_{jk} 中路径的数量. δ_l^v 的定义为

$$\delta_l^v = \begin{cases} 1, & \text{节点 } v \text{ 在路径 } l \text{ 中;} \\ 0, & \text{节点 } v \text{ 不在路径 } l \text{ 中.} \end{cases} \quad (7)$$

基于度和介数,对节点重要性进行排序的综合指标为

$$r_i = k_i + \frac{b_i^*}{k^*} b_i, \quad (8)$$

其中 b^* 和 k^* 分别表示所有节点中最大的度值和介数值. 按照综合指标排序后,选取 c 个重要节点作为候选节点, c 设置为

$$c = \alpha K, \quad (9)$$

其中 α 为调节候选节点数量的参数. 当 α 较大时,保留的候选节点多,但是计算时间长;当 α 较小时,保留的候选节点少,计算效率高.

2.2 算法过程

基于重要节点的贪婪算法与算法1的过程类似,不同的是将候选节点从全部节点集 V 调整为筛选出的重要节点集 V_c . 对于网络 G ,随着节点的删除,节点的重要度值 r 会发生变化. 因此,为进一步提高计算效率,每选取一个节点加入 S ,就对 V_c 中的剩余节点进行重排序和再次筛选. 令 $R(V_c, c)$ 表示对 V_c 中的节点按照式(8)排序并筛选出前 c 个节点的函数. 基于重要节点贪婪算法的计算过程如下.

算法2 基于重要节点的贪婪算法 (GABIN 算法).

输入: 网络 G , 节点个数 K , 调节参数 α ;

输出: 节点集 S .

step 1: 初始化 $S = \{v_i\}, i = 1, V_c = V$;

step 2: $c = \alpha(K - i + 1)$;

step 3: $V_c = R(V_c, c)$;

step 4: $v = \arg \max_{v_j \in V_c} (f(S \cup \{v_j\}) - f(S))$;

step 5: $S = S \cup \{v\}$, 从 V_c 中删除 v ;

step 6: 如果 $i < K$, 令 $i = i + 1$, 返回 step 2;

step 7: 输出 S .

对 GABIN 算法的计算复杂度进行分析. 网络效率计算复杂度为 $O(N(N + M))$; 节点排序中,度的计算耗时可忽略不计,介数的计算复杂度相当于 $O(N(N + M))$. GABIN 算法的计算复杂度约为 $O(K(c + 1)N(N + M))$, 其中 $c = \frac{1}{2}\alpha K$. 当 $K \ll N$ 时,相对于贪婪算法, GABIN 算法的计算复杂度下降约一个量级.

3 实验与结果分析

为验证 GABIN 算法的性能,进行计算复杂度对比实验和网络毁伤对比实验. 实验计算机处理器为 Intel(R) Core(TM) i5-3470@3.20 GHz, RAM 为 4 GB; 操作系统为 Windows7 64 bit; 软件环境为 Matlab 2014b 和 Lev Muchnik 开发的复杂网络工具箱^[9]. 实验网络分别为模型网络和实证网络,模型网络为无标度网络,实证网络为文献[20]中用到的 Power 网络. 计算复杂度实验中,与 GABIN 算法进行对比的算法分别为贪婪算法(Greedy)、最优化算法(Optimal); 网络毁伤实验中,对比算法分别为度(Degree)、介数(Betweenness)、接近度(Closeness)、删除节点法(DeleteNode).

3.1 计算复杂度实验

在 1.2 节和 2.2 节中分别给出了贪婪算法和 GABIN 算法计算复杂度的理论分析,但理论结果仅是概略的,并不能准确反映算法的实际运行时间. 这

里以无标度网络为对象进行算法运行实验,实验参数设置为毁伤节点数 $K = 3, \alpha = 5$,无标度网络的度指数 $\gamma = 2$. 在不同网络规模 N 下的算法运行时间如表1所示. **Optimal**算法的计算过程为: 1) 从 N 个节点中找出所有节点数为 K 的组合; 2) 计算每个节点组合的毁伤值; 3) 对毁伤值进行排序,找到最优组合. 由于算法程序结构原因, **GABIN**算法和 **Greedy**算法仅利用一个CPU核心进行运算;对于最优化算法,利用Matlab并行计算工具开启4个CPU进行运算提速. 从表1可以看到,相对于 **Greedy**算法, **Optimal**算法的计算耗时随着 N 的增大快速提高. 在 $N = 90$ 时, **Optimal**算法的耗时已高出 **Greedy**算法200倍,考虑计算条件差异,真实差距接近800倍. 当 N 的增大步长调整为100时, **Optimal**算法的耗时已难以在实验计算机获得. 此时, **Greedy**算法的耗时也随着 N 的增大而快速提高,在 $N = 700$ 时耗时2569.27 s. 实际网络规模远大于700个节点,因此 **Optimal**算法和 **Greedy**算法在实际网络中是难以适用的.

表1 3种方法运行时间对比 单位: s

N	GABIN	Greedy	Optimal
30	0.12	0.27	5.116
40	0.17	0.59	20.06
50	0.24	1.11	58.23
60	0.35	1.79	150.78
70	0.46	2.88	326.54
80	0.56	3.90	615.48
90	0.64	5.49	1107.95
100	0.83	7.36	
200	3.62	57.75	
300	8.07	166.68	
400	13.70	418.96	
500	20.89	832.23	
600	28.82	1624.53	
700	40.68	2569.27	

相对于 **Optimal**算法和 **Greedy**算法, **GABIN**算法的运行时间明显较少. 同时, **GABIN**算法的耗时增加量也比较小. 当 $N = 100$ 时, **Greedy**算法耗时约为 **GABIN**算法的9倍;当 $N = 700$ 时,倍数上升到63倍,说明倍数随着 N 的增大在不断扩大. 因此,对于规模越大的网络, **GABIN**算法的时间开销优势越大.

3.2 无标度网络毁伤实验

利用网络模型可以考察不同网络结构对算法效果的影响,并获得统计意义上的算法运行结果. 现实世界中很多网络具有无标度特征,因此基于无标度网络模型展开仿真实验. 算法参数设置为 $K = 3, \alpha = 5$,考虑计算时间将网络规模设置为 $N = 300$. 通过改变度指数 γ 控制网络结构变化,针对每一组参

数 (N, γ) 生成100个网络. 对比算法的计算过程均为: 1) 按照相应指标对节点进行排序; 2) 选取前 K 个节点作为节点集; 3) 删除节点集,计算网络效率. 各算法毁伤效果的平均值如图2所示,为观察不同节点集的差异性,表2给出了排序算法与 **GABIN**算法所得节点集的节点重合百分比.

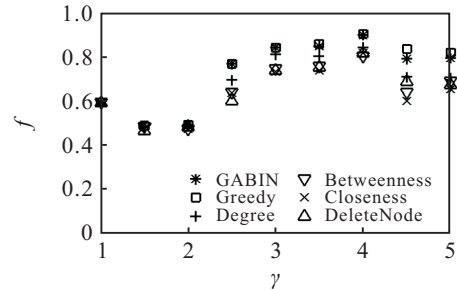


图2 各算法对于不同结构无标度网络的毁伤效果

表2 排序算法与GABIN算法节点集的重合比例

	γ								
	1	1.5	2	2.5	3	3.5	4	4.5	5
Degree	98	88	85	70	68	59	72	54	55
Betweenness	100	86	80	64	65	63	69	49	48
Closeness	97	81	77	62	62	62	65	50	51
DeleteNode	98	83	80	70	70	67	72	52	54

从图2可以看到: 当 $\gamma = 1$ 时,各算法的效果基本相同;随着 γ 增大,各算法的效果表现出越来越大的差异性;当 $\gamma \geq 2.5$ 时,各算法的效果出现明显的差异, **Greedy**算法和 **GABIN**算法的毁伤效果显著优于其他算法. 这说明, γ 越小,网络的度分布越偏倚,度大的节点越显著,各算法均较准确地找到这些节点; γ 越大,网络的度分布相对均匀,节点之间的差异性越小,按照排序选取的节点组合并不一定是毁伤最大的节点集,而 **Greedy**算法和 **GABIN**算法找到的节点集更接近于最大毁伤节点集. 在计算时间上,结合表1可以发现, **GABIN**算法以1/20的计算时间获得的效果等于或接近于 **Greedy**算法的效果.

从表2可以看到,当 $\gamma = 1$ 时, **GABIN**算法所得节点集中98%、100%、97%、98%的节点分别在 **Degree**、**Betweenness**、**Closeness**、**DeleteNode**算法所得节点集中出现,此时各算法所得节点集几乎是相同的. 随着 γ 变大,节点集的重合比例呈下降趋势,在 $\gamma = 2$ 时,重合比例约为80%;在 $\gamma \geq 2.5$ 时,重合比例下降到为70%及以下. 结合图2发现,此时毁伤效果也表现出了较大的差异,说明毁伤效果与节点集的重合比例具有一致性.

影响 **GABIN**算法效果的关键参数是调节候选节点数量的 α . 为测试 α 的影响,设置无标度网络的生成参数 $N = 300, \gamma = 4$. 生成100个网络,各算法

毁伤效果的平均值如图3所示. 当 $\alpha = 1$ 时, GABIN算法退化为按照排序结果选取节点, 因此其毁伤效果与Degree方法接近; 随着 α 增大, GABIN算法的效果不断提高并接近于Greedy算法, 且显著优于其他算法. 值得注意的是, 当 $\alpha = 8, 9, 10$ 时, GABIN算法的毁伤效果分别为0.4261、0.4273、0.4260, 均超过了Greedy算法的0.4258. 由于Greedy算法求得的是近似解, 并不是最优解, 出现上述结果是合理的. 这说明, 相对于基于所有节点, 基于重要节点有可能寻找到更优的节点集.

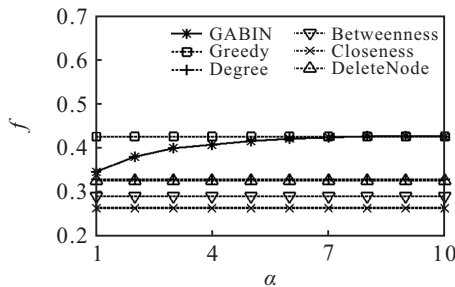


图3 α 对GABIN算法效果的影响

图3从统计均值的意义上给出了算法效果的对比, 表3给出了100个网络的具体对比结果, f_{GABIN} 表示GABIN算法的效果, f_{Greedy} 表示Greedy算法的效果, f_{others}^{max} 表示其他算法中的最好效果. 当 $\alpha = 1$ 时, GABIN算法效果等于或优于Greedy算法的网络数量为17个; 当 $\alpha = 10$ 时, 这一数字上升到83个, 其中在74个网络中两个算法是等同的, 在9个网络中GABIN算法是更优的, 在其余的13个网络中Greedy算法是更优的. 随着 α 增大, GABIN算法效果提升的情况同样发生在与其他算法的对比中. 取其他算法的最高值与GABIN算法进行对比, 在 $\alpha = 1$ 时, GABIN算法效果等于或优于其他算法最高值的网络数量为54个; 在 $\alpha = 5$ 时, 上升为93个; 当 $\alpha > 5$ 时, 虽然总数没有变化, 但其中GABIN算法效果更优的网络数量继续上升. 因此, 可以得出与图3类似的结论, 随着 α 的增大, GABIN算法的效果接近Greedy算法, 并大幅优于其他算法.

表3 各算法在100个网络中的效果对比

	α									
	1	2	3	4	5	6	7	8	9	10
$f_{GABIN} \geq f_{Greedy}$	17	32	42	52	61	67	74	79	83	83
$f_{GABIN} < f_{Greedy}$	8	12	16	15	17	16	14	12	12	9
$f_{GABIN} \geq f_{others}^{max}$	54	73	87	89	93	93	93	93	93	93
$f_{GABIN} < f_{others}^{max}$	26	53	70	77	80	82	85	86	88	87

要注意的是, 并不是 α 越大, GABIN算法效果越好, 如表3中, GABIN算法效果最好的参数是 $\alpha = 9$. 在 $\alpha = 9$ 时, GABIN算法等同或优于Greedy算法和

其他算法最大值的网络数量分别为83、93, 虽然与 $\alpha = 10$ 时相同, 但GABIN算法效果更优的网络数量更多, 分别为12、88. 目前在实验中尚未发现确定 α 最优值的规律, 一个可行的做法是随着 K 和网络节点数 N 的增大, 适当提高 α 的取值.

3.3 Power网络毁伤实验

为验证GABIN算法在实际网络中的应用效果, 针对Power网络展开毁伤实验. Power网络^[20]是从美国西部地区电力网抽象出来的无权无向网络, 包含了4941个节点和6594条边. 考虑计算时间, 未进行Greedy算法实验, GABIN算法参数设置为 K 取1到10, $\alpha = 10$, 实验结果如图4所示. 当 $K \leq 3$ 时, GABIN算法的效果与DeleteNode算法相近, 但优于其他算法; 当 $K > 3$ 时, GABIN算法的效果显著优于所有其他算法. 在计算时间方面, Degree、Betweenness、Closeness算法的计算时间小于GABIN算法; 当 $K = 1$ 时, DeleteNode算法的计算时间约是GABIN算法的500倍; 当 $K = 10$ 时, DeleteNode算法的计算时间约是GABIN算法的5倍. 可以得出结论, GABIN算法比DeleteNode算法效果更好、计算时间更少, 在效果上大幅优于Degree、Betweenness、Closeness等算法.

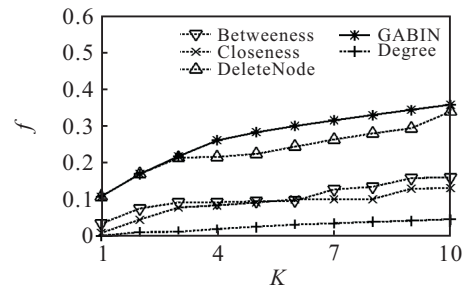


图4 各算法对于Power网络的毁伤效果

在 $K = 10$ 时, GABIN算法耗时约为5.93h, Greedy算法的计算时间据推算约是GABIN算法的50倍, 约为296h. 因此, 在给定的实验计算条件下, Greedy算法难以应用到Power网络上.

表4给出了 $K = 10$ 时GABIN算法所得节点按照中心性指标排序的结果. 按照度指标排序, 仅有一个节点的排序为49, 其他节点均排在100以后; 按照介数指标排序, 排序前10的节点仅有3个, 排序45及以后的节点有5个; 按接近度指标排序, 排序前10的节点仅有2个, 其他节点均排在37及以后, 甚至出现了排序为900和631的节点. 现实中, 网络的中心节点往往得到更多的防护, 从而使攻击代价升高. GABIN算法所得的节点集中包含了大量的非中心节点, 相比排序算法更有可行性, 毁伤的效果也更好.

表4 Power网络中GABIN算法所得节点的排序值

	节点序号									
	727	2544	427	693	2236	1244	2605	70	1314	1092
Degree	193	270	331	321	108	300	101	209	189	49
Betweenness	67	2	9	97	17	3	78	13	64	45
Closeness	631	3	67	900	70	9	76	50	37	55

4 结论

在攻击网络时,人们希望以一定的代价获得最大的效果.在以往的工作中,研究者们多侧重于对网络节点进行重要性排序,并选取排序靠前的节点进行攻击.但是,排序靠前的节点并不一定能够造成最大的毁伤效果.针对这一情况,在给定攻击节点数量的情况下,本文定义了网络毁伤最大化问题,并参照节点影响力最大化问题给出了求解网络毁伤最大化问题的近似求解方法,即贪婪算法.贪婪算法计算复杂度较高,难以应用于大规模网络.针对这一问题,提出了基于重要节点的贪婪算法,在计算复杂度方面比传统贪婪算法降低了约一个量级.基于模型网络 and 实际网络进行了实验,结果表明GABIN算法能够有效地降低计算时间.在毁伤效果方面,通过调节参数 α ,GABIN算法的效果能够接近于甚至优于Greedy算法;对于不同结构的无标度网络和Power网络,GABIN算法在各种参数设置下均优于按照排序选取节点攻击的算法.

相对于Greedy算法,所提出的GABIN算法在效果相当的情况下,能够大幅度地降低计算时间,使其可应用于大规模的实际网络.但还存在的问题是,参数 α 的最优值尚不能自动确定,目前只能根据攻击节点的数量和网络规模进行估算.自动估计参数 α ,提升算法效果,将是下一步研究的重点.

参考文献(References)

- [1] Chen G R, Wang X F, Li X. Introduction to complex networks models, structures and dynamics[M]. Beijing: Higher Education Press, 2015: 3-13.
- [2] Liu J, Jiang Z Y, Kato N, et al. Reliability evaluation for NFV deployment of future mobile broadband networks[J]. IEEE Wireless Communications, 2016, 23(3): 90-96.
- [3] Sabidussi G. The centrality index of a graph[J]. Psychometrika, 1966, 31(4): 581-603.
- [4] Stella M, Domenico M D. Distance entropy cartography characterises centrality in complex networks[J]. Entropy, 2018, 20(4): 268.
- [5] Freeman L C. A set of measures of centrality based on betweenness[J]. Sociometry, 1977, 40(1): 35-41.
- [6] Borgatti S P. Centrality and network flow[J]. Social Networks, 2005, 27(1): 55-71.
- [7] Kitsak M, Gallos L K, Havlin S. Identification of influential spreaders in complex networks[J]. Nature

Physics, 2010, 6(11): 888-893.

- [8] Restrepo J G, Ott E, Hunt B R. Characterizing the dynamical importance of network nodes and links[J]. Physics Review Letters, 2006, 97(9): 94-102.
- [9] Jiang Z Y, Liu Z Q, He X, et al. Cascade phenomenon against subsequent failures in complex networks[J]. Physica A: Statistical Mechanics and its Applications, 2018, 499: 472-480.
- [10] Jiang Z Y, Ma J F. Deployment of check-in nodes in complex networks[J]. Scientific Reports, 2017, 7: 40428.
- [11] Deng Y, Wu J, Xiao Y, et al. Efficient disintegration strategies with cost constraint in complex networks: The crucial role of nodes near average degree[J]. Chaos, 2018, 28(6): 061101.
- [12] Li Y P, Tan S Y, Deng Y, et al. Attacker-defender game from a network science perspective[J]. Chaos, 2018, 28(5): 051102.
- [13] Li J, Wu J, Li Y, et al. Optimal attack strategy in random scale-free networks based on incomplete information[J]. Chinese Physics Letters, 2011, 28(6): 068902.
- [14] Deng Y, Wu J, Tan Y J. Optimal attack strategy of complex networks based on tabu search[J]. Physica A: Statistical Mechanics and its Applications, 2016, 442: 74-81.
- [15] Ma Q, Ma J. Identifying and ranking influential spreaders in complex networks with consideration of spreading probability[J]. Physica A: Statistical Mechanics and its Applications, 2017, 465: 312-330.
- [16] Nguyen D, Nguyen T, Do T, et al. Probability-based multi-hop diffusion method for influence maximization in social networks[J]. Wireless Personal Communications, 2017, 93(4): 903-916.
- [17] Latora V, Marchiori M. A measure of centrality based on network efficiency[J]. New Journal of Physics, 2007, 9(6): 188.
- [18] Nemhauser G, Wolsey L, Fisher M. An analysis of the approximations for maximizing submodular set functions[J]. Mathematical Programming, 1978, 14(1): 265-294.
- [19] Muchnik L. Complex networks package[DB/OL]. (2013-02-26)[2018-07-13]. <http://www.levmuchnik.net/Content/Networks/ComplexNetworksPackage.html>.
- [20] Watts D J, Strogatz S H. Collective dynamics of 'small-world' networks[J]. Nature, 1998, 393(6684): 440-442.

作者简介

刘凤增(1987-),男,讲师,博士生,从事军事信息系统及复杂网络的研究,E-mail: fengzeng_liu@126.com;

肖兵(1966-),女,教授,博士生导师,从事军事信息系统综合集成等研究,E-mail: xb_sky@126.com;

金宏斌(1976-),男,副教授,博士,从事军事信息系统的研究,E-mail: jhb0817@tom.com;

李浩(1981-),男,讲师,博士,从事群体智能的研究,E-mail: snk.poisson@163.com.