相互作用网络的级联失效问题

摘要

本文通过建立级联失效模型、混合整数规划模型、蒙特卡洛模拟算法以及遗传算法，对相互作用网络级联失效、攻击优化问题和级联失效问题高效算法评价等问题进行分析研究，并给出科学合理的解决方法。

针对问题一：首先对级联失效问题进行机理性分析，认识到级联失效的关键在于节点及链接的破坏，定义级联失效中链接剔除准则，给出最大相互连通集团的识别方法，建立针对一个节点随机攻击的级联失效模型和算法。以图 1 网络为例攻击，计算最终最大相互连通集团的大小，结果证明我们的模型算法是准确、高效的。

针对问题二：通过对网络度、平均度、度分布属性进行综合分析后，选择级联抗毁性最差、即级联失效效应最显著的网络 4，基于问题一中的级联失效模型，利用改进的蒙特卡洛模拟法进行足够多次数的仿真模拟，得出 p、r 关系图，研究图线性质可以得出网络 4 的攻击阈值 p_c 在(0.5,0.6)范围内。

针对问题三：考虑到每个节点的链接差异性，通过设立度大原则、关键位置原则、最大连团原则的优先攻击原则，以最终最大相互连通集团的节点数量 g 最少为目标，建立混合整数规划模型。用蒙特卡洛模拟法随机生成节点组合，以最大相互连通集团节点数最少为最终输出条件并进行足够多次数的循环，以此来逼近不同 m 值下最优的攻击点策略，得到 m、g 关系图，研究 g 随 m 变化而变化的趋势、突变点等属性，可以看出网络 1、2、3 鲁棒性较好，网络 4 较差。

针对问题四：考虑到已知网络的复杂性，我们选择准确性高、收敛速度快的遗传算法，利用 matlab 编程求解第三问中优化问题。为了检验所建立的遗传算法效率，通过选择时间复杂度、空间复杂度、正确性及健壮性四个指标，利用层次分析法对用到的遗传算法和蒙特卡洛算法进行综合评分，得出 0.6595>0.3406，故遗传算法更高效。

最后，对本文建立的模型和算法进行了较为全面的评价，并对其良好的实用性进行了推广。

关键词：级联失效模型、混合整数规划、攻击阈值、蒙特卡洛算法、AHP、遗传算法

一、问题重述

1.1 问题的提出

近年来，相互作用网络受到广泛关注与研究。在相互作用网络中，除了每个网络中内部节点间存在边外，网络之间还存在相互依赖的边。如果一个节点失效，而且该点存在依赖边指向另外一个网络，这就会导致依赖边所指向的节点失效，从而放大了失效节点的后果。这种效应我们通常称为级联失效，级联失效往往导致整个相互依赖系统的崩溃。例如，2003 年意大利和北美停电事故中，均存在电力-计算机网络的大面积崩溃，其原因是电力-计算机网络构成了相互作用网络，级联失效的示意图，如图 1-1 所示[1]。

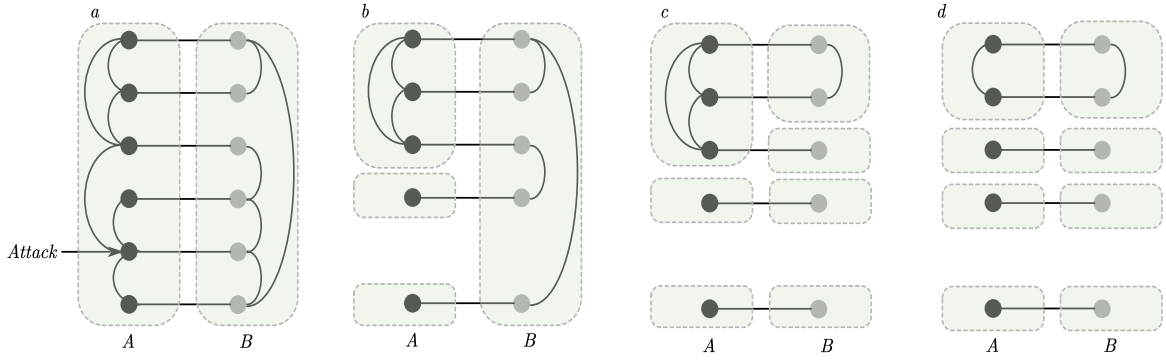


图 1-1 级联失效实例

1.2 待解决的问题

基于上述资料，完成下述问题：

(1)给定一个相互作用网络，任选一个节点，攻击该节点后，产生级联失效后产生的最大相互连通集团，用 matlab 编程求该最大相互连通集团的大小。使用图 1-1 的例子检验程序的正确性。

(2)附件 1（注意数据说明，数据必须用 matlab 打开）中包含了 4 个相互作用网络，请基于(1)中的程序，随机攻击其中一个网络中所有节点中比例为 p(0<p<=1)的节点，得出最大相互连通集团的节点数与总节点数的比例 r。考虑到随机性，固定 p，给出随机攻击 100 次后 r 的均值，并画出 p 和 r 的均值关系图。

(3)建立优化模型，选择 m 个节点进行攻击，使得最终最大相互连通集团的节点数量 g 最少。基于该模型，当 m 为 1-50 时，求出附件 1 中 4 个相互作用网络的 g，并画出 m 和 g 的关系图。

(4)评价上述模型的求解效率，建立更高效的求解算法。

二、问题分析

2.1 问题一的分析

问题一要求给定一个相互作用网络，任选一个节点进行攻击，级联失效发生后，用matlab 编程求最终最大相互连通集团的大小，并用图 1-1 的例子检验程序的正确性。

级联失效模型的核心是失效链接的删除。在相互作用网络中，各点均在另外一个网络存在一个依赖节点，若依赖节点与该点不属于一个连通集团，则剔除两点相应的出边，使两点从属于一个连通集团。因此，各点与其依赖节点从属于一个连通集团是执行目标，其具体表现为删除被攻击的点及其出边、依赖点及其出边、导致各对相互依赖的点不属于一个连通集团的链接。

考虑到问题的一般性，首先利用蒙特卡洛模拟法随机生成一个相互作用网络，该网络由顶点数都为 N 的网络 A 和网络 B 组成。对网络的各类参数进行定义后，定义级联失效中链接剔除准则，以及最大相互连通集团的识别方法，建立针对一个节点进行随机攻击的级联失效模型和算法。最后，将图 1-1 中的相互作用网络转化成邻接矩阵，验证上述模型及算法的准确性。

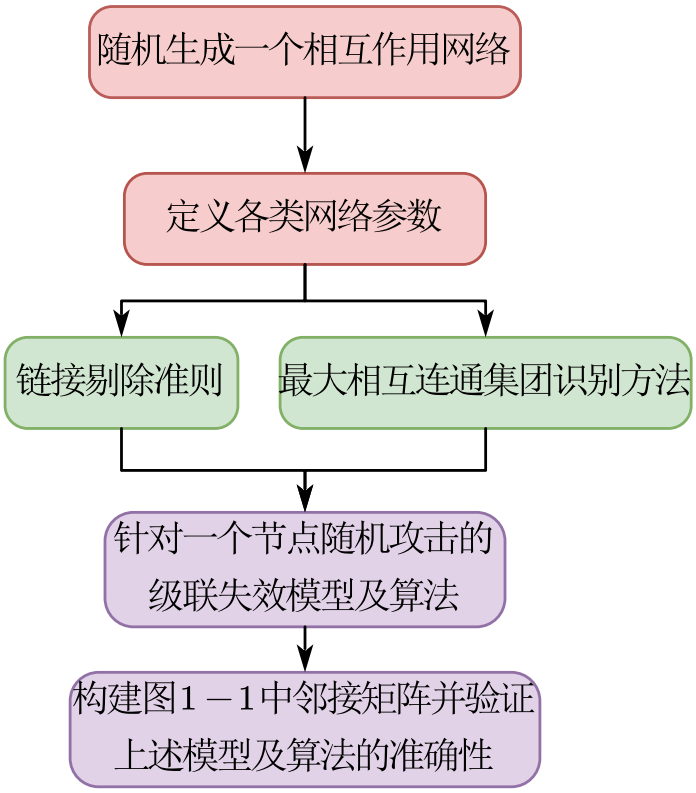


图 2-1 问题一思路流程图

2.2 问题二的分析

针对节点间链接关系固定的已知网络，给出对多个节点进行随机攻击的级联失效模型，并研究被攻击节点数所占总节点数比例 p 与攻击后的最终状态下最大连通集团所含节点数占总节点数比例 r 的关系。

观察到附件中给出的四个网络的表示既有稀疏矩阵形式，也有邻接矩阵形式，全部处理成邻接矩阵形式。考虑到网络度大节点受攻击后对网络破坏效率更大，那么平均度

更小的网络受到相同攻击后级联失效现象更显著，因此求解四个网络的平均度<k>，选择平均度最大的网络进行攻击。问题二中认为该网络节点都是无差别的，所以每个节点被随机攻击到的概率是相等的，基于问题一的级联失效模型，通过蒙特卡洛模拟法给出随机参数 p，求解受到攻击后最终的网络状态及该状态下的 r 值大小。最后根据得出的 p、r 关系图对该网络的级联失效进程、攻击阈值等网络属性进行分析。

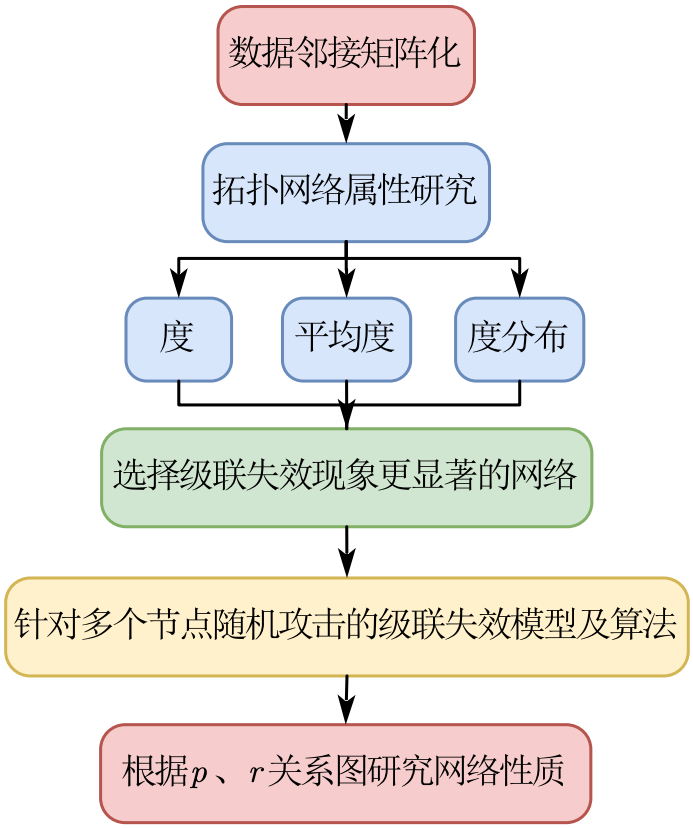


图 2-2 问题二思路图

2.3 问题三的分析

问题三要求建立以最终最大相互连通集团的节点数量 g 最少为目标的选择 m 个点的优化模型，当攻击点数 m 为 1-50 时，基于该模型求解 4 个已知网络的 g，并研究 m、g 的相互关系。

考虑到每个节点的链接差异性，为了实现最终最大相互连通集团的节点数量最少的目标，在度大攻击原则、关键位置攻击原则和最大连团攻击原则约束下，建立混合整数规划模型，利用蒙特卡洛模拟法随机生成节点组合，以最大相互连通集团节点数最少为最终输出条件并进行足够多次数的循环，以此来逼近不同 m 值下最优的攻击点策略。画出 m 和 g 的关系图后，研究 g 随 m 变化而变化的趋势、突变点等属性，做出相关的理论分析。

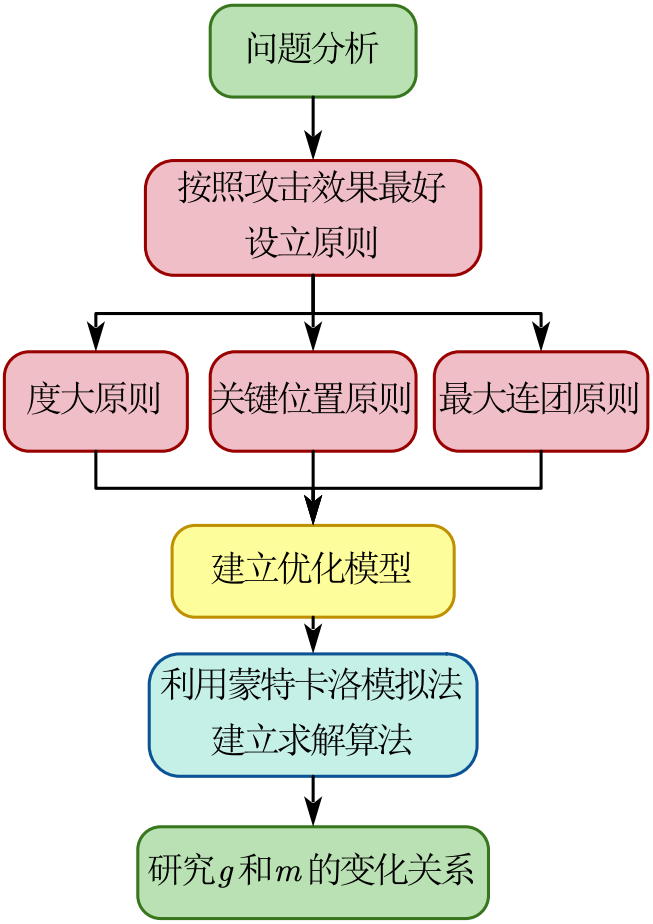


图 2-3 问题三思路流程图

2.4 问题四的分析

问题四要求评价问题三算法效率，并针对问题三的问题给出更高效的算法。

在相互作用网络中，攻击节点的选择方案多种多样，且产生级联失效后产生的最大相互连通集团中节点数量的优化常常涉及大量的信息，计算量大且复杂一般、一般需要借助优化算法解决该问题，如线性规划、动态规划、Dijkstra、遗传算法、模拟退火算法等算法，这些算法大多数建立在图论的基础上寻找最优解，通过建立邻接矩阵，设置限制条件确定合理的选择，考虑到已知网络的复杂性，我们选择了准确性和收敛速度最快的遗传算法，利用 matlab 通过编码、设置适应度函数等步骤求解第三问中优化问题，针对 data4 数据进行结果的比较。为了检验所建立的遗传算法效率，通过选择合适的指标，利用 AHP 层次分析法对遗传算法和第三问中的蒙特卡洛算法进行综合评分。

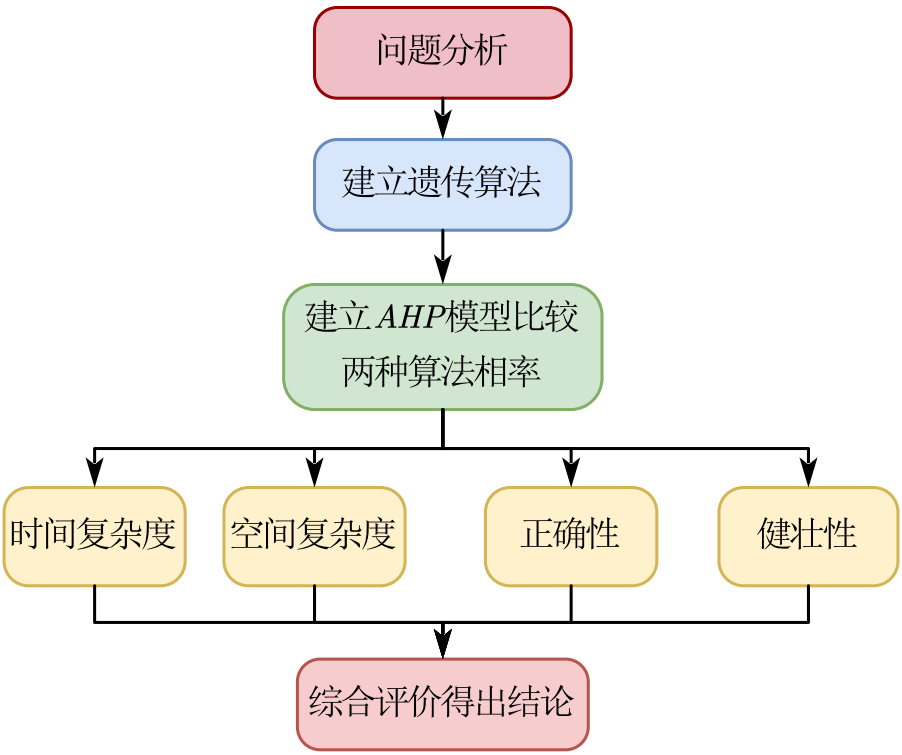


图 2-4 问题四分析图

三、模型假设

假设 1：文中相互作用网络都是由相同位置节点具有相互依赖关系的两个子网络组成。

假设 2：只考虑对节点的随机攻击，各节点被攻击的概率相等。

假设 3：假设每次攻击都是有效的。

假设 4：位置相同的两节点间相互依赖关系是稳定的，只有其中一个节点受到攻击时才会切断该链接。

四、符号说明

符号	符号说明
a_{ij}	A 网络节点
b_{mn}	B 网络节点
$L_{f_{\max}}$	连通分量
A_t	被攻击点的集合

$c_{i_1 i_2}$	A 网络中链接
$d_{m_1 m_2}$	B 网络中链接
e_{im}	相同位置节点依赖关系
k_i	节点的度数
$\langle k \rangle$	网络的平均度
p_k	度分布
q	总链接数

注：未列出符号及重复的符号以出现处为准

五、模型的建立与求解

5.1 问题一的模型建立与求解

5.1.1 定义相互作用网络及网络参数

（1）相互作用网络

分析级联网络失效问题，数学上常用图来描述，任何一个网络都可以看成是由一些节点按照某种方式连接构成的一个网络，相互作用网络也不例外。具体网络的抽象图表示，就是用抽象的节点表示具体网络中的节点，并用节点之间的连线（或称之为边），来表示网络节点之间的关系。一个具体网络可以抽象为一个由点集 N 和边集 C 组成的图 $G_A = (N, C)$。

考虑到网络的一般性，建立一种由相互依赖的 A、B 网络组成的相互作用网络，其中 A 网络和 B 网络中的相同位置的节点相互依赖，各网络中所有节点间相互联系是随机的。

（2）网络节点 a_{ij}，b_{mn}

将相互作用网络中 A、B 网络包含的各节点分别编号，A 网络中的节点定义为 a_{ij}，B 网络中的节点定义为 b_{mn}。其中 i,m 分别表示该节点在 A 网络和 B 网络中的位置，j,n 表示该节点所属连通分量的编号。

（3）连通分量 L

在 A、B 各自网络中，若某几个点之间是相互连通且任意两点之间是可达的，这种情况下，我们称这几个点属于一个连通分量，该连通分量用这几个点中最大的编号值来表示，如连通分量中节点的最大编号为 4，则该连通分量为 L4。具体表示如下图：

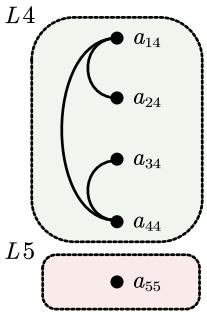


图 5-1　连通分量示意图

（4）联系（边）c_{i_1,i_2}，d_{m_1,m_2}，$e_{i,m}$

定义网络中各节点间的联系，包含各自网络中各节点的联系、各节点与其自身联系以及网络间相同位置节点间的相互依赖关系，通过 0-1 变量表示联系的成立与否，而且当级联失效时，联系的破坏也可以用 0-1 变量表示。

$$c_{i_1,i_2}=\begin{cases}1，A\text{网络中节点}i_1\text{和}i_2\text{之间存在联系}\\0，A\text{网络中节点}i_1\text{和}i_2\text{之间不存在联系}\end{cases}\tag{1}$$

$$d_{m_1,m_2}=\begin{cases}1，B\text{网络中节点}m_1\text{和}m_2\text{之间存在联系}\\0，B\text{网络中节点}m_1\text{和}m_2\text{之间不存在联系}\end{cases}\tag{2}$$

用$e_{i,m}$表示 A、B 网络相同位置节点的相互依赖的关系，一般位置相同的两节点间相互依赖关系是稳定的，只有在该两个节点其中一个受到攻击时才会切断该联系。

5.1.2 级联失效模型

（1）攻击模型：攻击节点，链接失效

$$\begin{cases}\sum_{i\in A_t}c_{ii}=0 & & (3\text{-}1)\\ \sum_{i_2=1}^{N}\sum_{i_1\in A_t}c_{i_1i_2}=0 & & (3\text{-}2)\\ \sum_{i\in A_t}e_{i,m}=0 & i=m & (3\text{-}3)\\ \sum_{i\in A_t}d_{m_1m_2}=0 & m_1,m_2\in\{1,...,N\} & (3\text{-}4)\\ \sum_{m_1=1}^{N}d_{m_1m}=0 & i=m,j\neq n & (3\text{-}5)\\ \sum_{i_1=1}^{N}c_{i_1i}=0 & i=m,j\neq n & (3\text{-}6)\\ c_{ii_1}=d_{mm_1}=1 & i=m,i_1=m_1 & (3\text{-}7)\end{cases}\tag{3}$$

（3-1）、（3-2）表示被攻击节点失效（A_t指的是被攻击点的集合），同时被攻击点的

所有出边失效。（3-3）、（3-4）表示删除其余节点对被攻击节点的依赖关系（$e_{im}=1$表示节点 i 和节点 m 间存在相互依赖关系），紧接着删除与被攻击节点有依赖关系的节点所有出边。（3-5）、（3-6）、（3-7）表示删边规则，即相同位置的不同子网络节点若不属于同一连通集团则删掉该两个节点的所有出边，除两个节点的依赖联系和共有的连向相同位置的出边。

删边终止：记$s_{ij}=\begin{cases}1 & i=m,j=n\\ 0 & \text{其他情况}\end{cases}$，当$\sum_{i=1}^{N}s_{ij}=N$时不再进行删边，此时网络 A 和网络 B 同一位置节点所属的连通集团为同一个。

（2）识别最大连通集团模型：链接枢纽，合并连通

为了通过边的关系来识别连通集团，首先我们建立只考虑单个子网络节点间有向链接的邻接矩阵，即为一个对角线上全为 0 的上三角$N\times N$方阵，方阵中的数为u_{rc}。

1）由单个点组成的孤立连通集团：$\sum_{c=1}^{N}u_{rc}=0$，$(r\text{为常数})$

2）度大节点的节点组合：若$u_{rc}=1$，$(r\text{为常数，}c=1,...,N)$，则$F_r\{f_1,f_2.....\}=\{r,c_1.....\}$，即该节点组合中包含该行自身代表节点以及所有与其有链接的节点。

3）合并节点组合，形成连通集团：若$f_1=f_2,(f_1\in F_1,f_2\in F_2)$，则$F_1=F_1\cup F_2$，且该连通集团的节点个数$L_{f_{\max}}=card(F)$。

4）比较得出最大连通集团：$L_A=L_B=\max(L_f)$

3）级联失效总概模型：

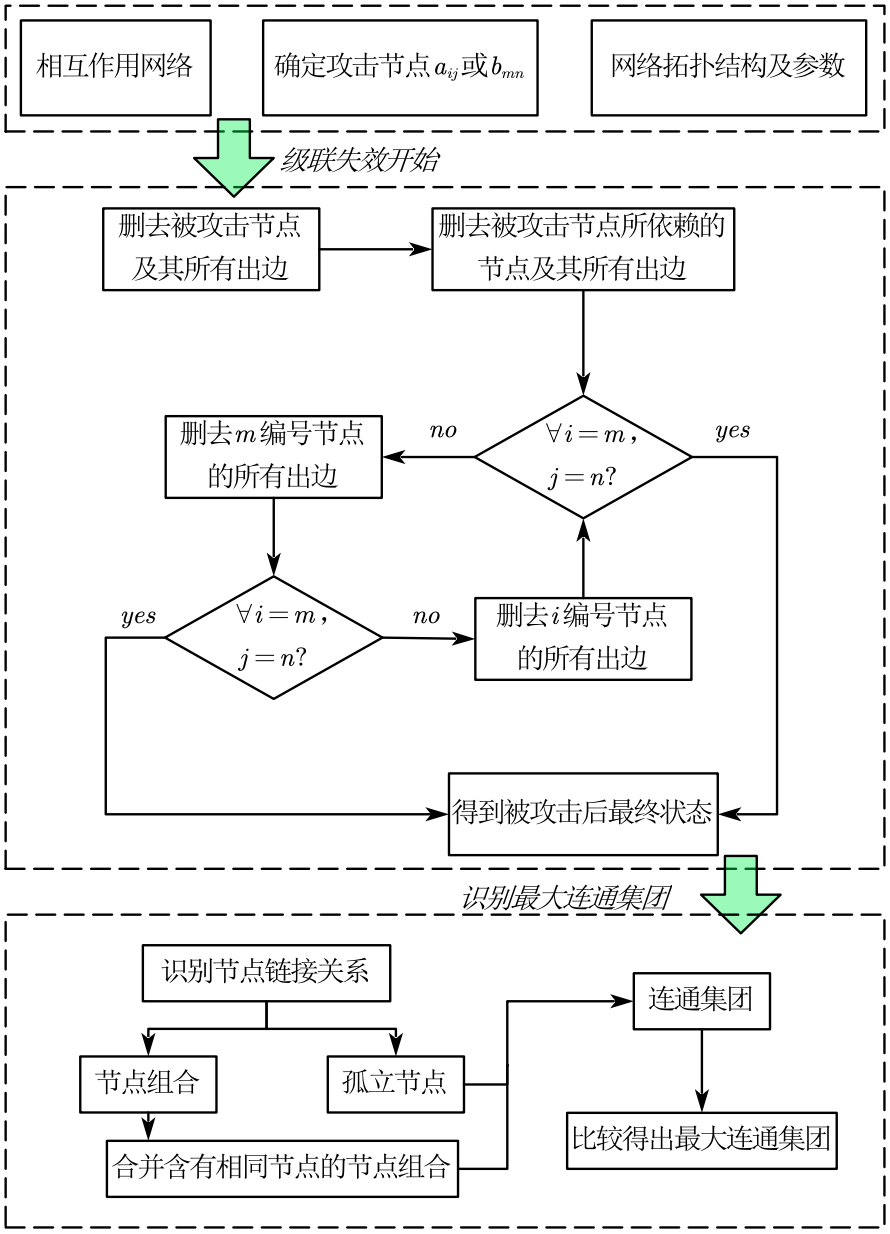


图 5-2　级联失效模型图

5.1.3 算法的建立

（1）算法描述

级联失效算法，即按照先节点后联系、先攻击后依赖的原则来剔除已知失效节点及其失效的联系。考虑到一般模型中两个子网络对节点所属连通集团的双重判断在算法程序中会拖慢运行速度，所以我们设置了针对于整个网络的链接剔除规则，以两个子网络中相同位置的每对节点所属连通集团编号都相等为网络级联失效最终停止条件，设置判断条件及循环，得到网络受到攻击后的最终状态。通过遍历各节点代表该节点所属连通集团编号的下标，来识别最终产生的各相互连通集团。

（2）具体形式

STEP 1：读入给定的网络邻接矩阵或随机生成相互作用网络的邻接矩阵，根据蒙特

卡洛模拟法选择网络中任一节点施行攻击；

STEP 2：将被攻击的节点及其所有出边、该被攻击节点在另一子网络中相同位置节点及其所有出边删除；

STEP 3：根据节点下标 i 或 m 遍历该子网络中所有节点，判断当前节点与其在另一子网络中相同位置的节点属于同一连通集团，若是，则终止算法，达到最终状态；若否，则考虑把该对点除相互依赖联系外所有联系删除；

STEP 4：判断 STEP 3 中待删除的联系中是否存在起点端点相同的联系，保留这种特征的联系，删除其他联系，转到 STEP 3；

STEP 5：计算最终产生的各相互连通集团大小，得出该最大值，算法结束。

5.1.4 模型结果与分析

使用上述算法运行赛题中例子，得出结果为：

表 5-1 攻击后节点数表

攻击的点	点1	点2	点3	点4	点5	点6
所剩最大连通集团节点数	8	10	6	4	4	6

攻击各点后，所剩最大连通集团所含的节点：

表 5-2 攻击后节点表

攻击的点	点1	点2	点3	点4	点5	点6
所剩最大连通集团节点	3、4 5、6	1、3、4 5、6	4、5、6	1、2； 5、6	1、2	3、4、5

经过画图分析实证，我们得出的结果与分析结果完全吻合，因此判定我们问题一的程序是准确的。

5.2 问题二的模型建立与求解

5.2.1 选择相互作用网络

为使相同节点数遭到攻击后级联失效的现象更显著，可以通过研究四个网络相关的结构属性来选择最合适的网络进行攻击。根据级联失效节点失效、其所有边失效以及该点相连的边失效的破坏特性，我们从度、平均度及度分布三个方面对给出的四个网络进行研究。

（1）度

度是网络节点的一个关键属性，表示该节点和其他节点之间的链接个数，我们使用 k_i 表示网络中第 i 个节点的度。那么在给定的四个网络中，总链接数 q 可以用节点度数之和来表示：

$$q=\frac{1}{2}\sum_{i=1}^{N}k_i \tag{4}$$

（2）平均度

平均度反映了一个相互作用网络的节点之间相互链接密切程度，也在一定程度上反映了相互作用网络的级联抗毁性，往往平均度高的相互作用网络相互联系更密切、级联抗毁性更好[1]。平均度可以定义为：

$$\langle k\rangle=\frac{1}{N}\sum_{i=1}^{N}k_i=\frac{2q}{N} \tag{5}$$

（3）度分布

度分布p_K表示“网络中随机选出的一个节点其度为k”的概率，度分布这一网络属性反映了网络的链接均匀性、整体链接程度。度分布是一个概率，则其必须满足归一化约束，即：

$$\sum_{k=0}^{\infty}p_k=1 \tag{6}$$

对于题中有N个节点的网络而言，其度分布可以表示为：

$$p_k=\frac{N_k}{N} \tag{7}$$

（4）网络选择

通过查阅资料并联系实际情况可知，相互作用网络的平均度与网络的级联抗毁性正相关，即网络越均匀、链接越稠密，级联抗毁性越强[2]。

因此我们从网络平均度的角度对四个网络的级联抗毁性进行初步判断。

首先将网络可视化，利用 matlab 根据邻接矩阵建图生成网络的图像（详见附录）。

表 5-3 四个网络的平均度表

Data 1	Data 2	Data 3	Data 4
20	20	19.9550	11.3864

从表 5-中可以看出网络 1 和网络 2 的平均度数值最大，那么 1、2 网络的级联抗毁性最强，网络 4 的平均度数值最小，那么 4 号网络的级联抗毁性最弱，为使级联失效的现象更加显著，基于此处分析我们选择网络 4。

为了进一步确定选择的网络是否满足级联失效现象的明显展示，我们对四个网络的度分布属性进行了研究，并得出了以下四个网络的度分布图。

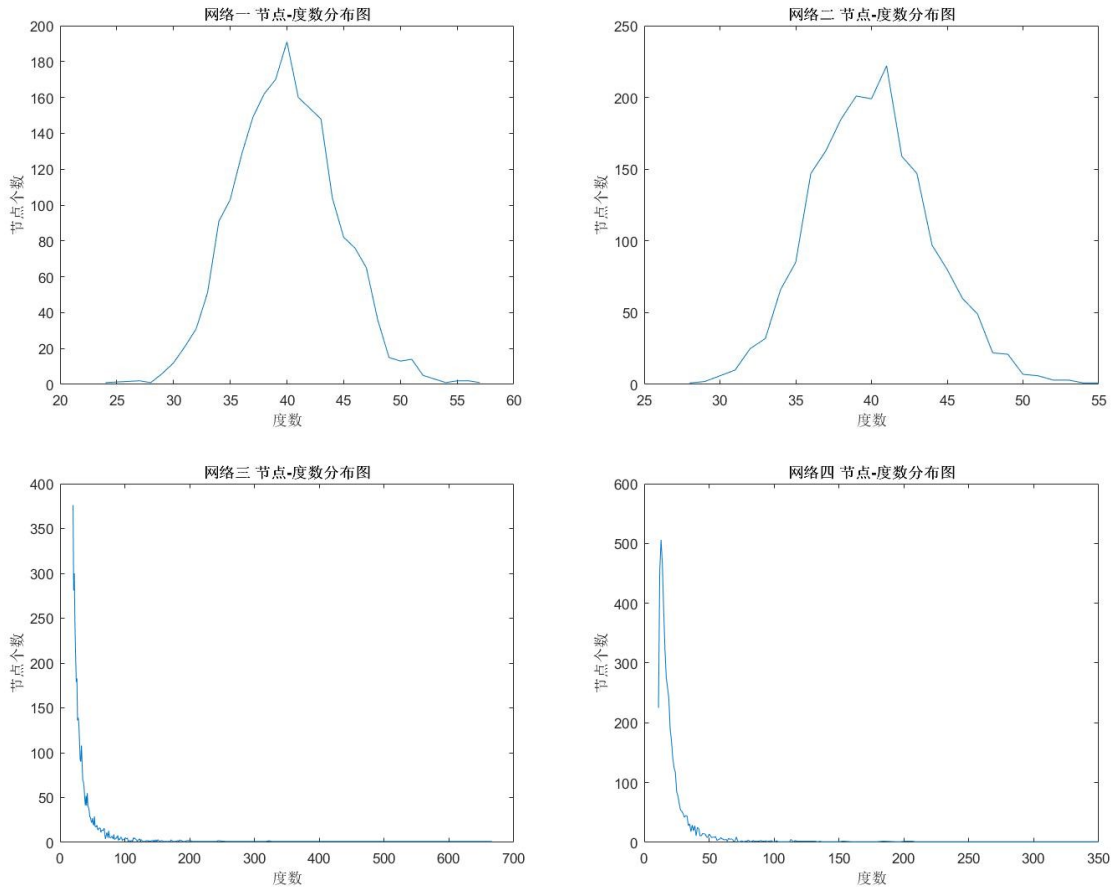


图 5-3 四个网络的度分布图

复杂网络中根据各个节点的关联程度将网络分为标度网络和无标度网络，其中标度网络是指网络中各个节点之间的连接都很紧密，而无标度网络指的是网络中的大部分节点只和少部分节点连接，少数的节点和非常多的节点相连。

结合得出的度数分布图，其中网络 1、2 点的度数分布比较均匀，节点的度都集中在某个特殊值附近，成钟形的泊松分布规律，据此可以得出结论网络 1、2 的节点各个网络之间的联系十分紧密，网络的稳健性较好，在攻击 1-50 个节点的范围内不易发生大面积级联失效的情况，稳健性类似于标度网络，而网络 3、4 由于存在少数的度数大的中心节点，所以在攻击节点个数为 1-50 的情况下会出现较大程度的级联失效。相应地，网络 4 存在个别与其他节点链接异常密切的点，若该类点遭到攻击破坏，那么整个网络势必发生强烈的级联失效现象。

综上，网络 4 的平均度最小、网络分布最不均匀，即网络 4 的级联抗毁性最差，其级联失效现象更显著，因此我们选择网络 4 进行问题二的研究。

5.2.2 针对多个节点随机攻击的级联失效模型的建立

蒙特卡洛概率模型

为了量化级联失效的过程，我们接下来通过建立针对多个节点随机攻击的级联失效模型，研究网络中最大相互连通集团的节点数与总节点数的比例 r 是如何随着被攻击点

的个数比例 p 变化而变化的。考虑到被随机攻击节点所占总节点数比例 p 是（0,1]任意的数值，我们采取蒙特卡洛模型随机选取 0 到 1 之间的数从而固定足够多数量的 p，使得数据图线平滑。

（1）随机数的定义

蒙特卡洛方法模拟某个过程的时候，需要产生许多种概率分布类型的随机变量。随机数在蒙特卡洛方法中起着十分重要的作用，它必须具备独立性、均匀性两个特点。

假设随机序列$\delta\{\delta_1,\delta_2.....\}$是相互独立且具有相同单位均匀分布的随机数序列，那么对于任意的自然数 t，由 t 个随机数组成的 t 维空间上的点$(\delta_{n+1},\delta_{n+2}.....\delta_{n+s})$，在 p 维空间的单位立方体$G_S$上均匀分布，即对于任意的$b_i,0\leqslant b_i\leqslant 1,i=1,2...t$，有如下等式成立：

$$P(\delta_{n-1}\leqslant a_i)=\prod_{i=1}^{t}a_i \quad i=1,...,t \tag{8}$$

（2）变量和参数的设立

1. $\delta\{\delta_1,\delta_2.....\}$表示随机序列；

2. $P(M)$表示 p=M 的概率；

3. $f(x)$表示随机抽样的概率密度函数；

4. $F(x)$表示随机抽样的分布函数。

（3）模型的建立

STEP 1:建立概率统计模型

考虑到数值 p 取值的随机性，我们选择以下概率模型作为随机抽样的模型：

p 在(0,1)上均匀分布（单位均匀分布），其分布密度函数为：

$$f(x)=\begin{cases}1,0<x<1\\0,\text{其他}\end{cases} \tag{9}$$

则其分布函数为：

$$F(x)=\begin{cases}0,x<0\\x,0<x\leqslant 1\\1,x>1\end{cases} \tag{10}$$

STEP 2:考虑特殊情况

但考虑到蒙特卡洛的随机性，所以函数的两端 1 和趋近于 0 可能无法取到，这会在一定程度上影响我们结果的准确性，因此我们将两种极端情况单独分析：

1）当$p=1$时，即攻击了网络中全部的节点，此时$\langle k\rangle=0$，所有节点都是孤立的。

因此，最大相互连通集团的节点数与总节点数的比例$r \to 0$。

2）当$p \to 0$时，即不攻击网络中任何的点，此时$\langle k \rangle = N - 1$，网络是完全连通图，所有节点属于同一个连通集团。因此，最大相互连通集团的节点数与总节点数的比例$r = 1$。

STEP 3:产生随机数 p

根据建立的概率模型，根据随机抽样方法产生 50 个随机数 p。

STEP 4:对每个 pN 完成 100 次随机选点攻击

因为问题二要研究的是被攻击节点占总节点数比例与最终最大相互连通集团的节点数占总节点数比例的关系，但是考虑到同一数目的不同被攻击节点对于级联失效的影响也不同，所以我们对于每一个 p 利用蒙特卡洛模拟法分别进行 100 次被攻击的节点选择，然后求其均值，这样就能够基本排除节点自身属性不同的影响，较准确地反应 p 与 r 纯粹数值间的关系。

5.2.3 利用 Matlab 确定被攻击点数比例

（1）确定被攻击节点数

根据多次实验，我们认为生成随机的 50 个 p 便能描绘出较为平滑的 p、r 关系图，因此确定模拟次数为 50。计算机模拟为问题求解提供了较好的选择，各种分布的随机数可以由 Matlab 软件通过编程直接产生。我们利用 Matlab 中 Randperm 函数，完成不重复的重排采样，即得到 50 个服从(0,1)均匀分布的随机数 p。

（2）产生随机数

由于我们只知道随机变量的取值在(0,1)，但不知它在何处出现的概率小，也不知在何处出现的概率大，所以，我们用 U（a,b）来模拟它。利用 Matlab 随机选取了$t1, t2 \cdots ti$个随机数，这些随机数服从(0,1)的均匀分布。

5.2.4 模型结果与分析

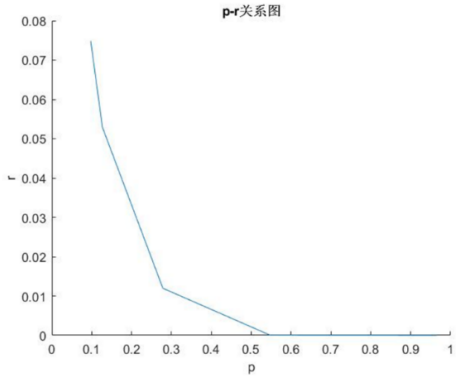


图 5-4 问题二结果图（data 4）

如图所示为所选网络 data4 的 p-r 均值关系图，从图中可以看出，当$0.5 < p < 0.6$

时，r 趋近于 0，即节点间的链接几乎全被破坏，说明 data4 网络的攻击阈值位于区间(0.5,0.6)。图中 p-r 曲线在区间(0,0.3)斜率绝对值大，r 值急剧下降，说明网络 data4 级联抗毁性差，容易引起大范围的级联失效故障。

5.3 问题三的模型建立与求解

5.3.1 参数定义

为了研究被攻击节点选择策略的最优化问题，我们将不再采用节点一次全部攻击的方式，而是一次次的攻击当前最优的节点，并通过严格执行其他约束条件，来达到最优化的目标。

具体表现：被攻击节点的选择是以最终最大相互连通集团的节点数量 g 最少为原则，从这一角度出发我们全面地选取度大、关键位置、最大连团为节点选择原则。根据这些原则，每攻击一次，通过计算来判定剩余各节点的的优先级，选择每一选择点的最优状态作为攻击节点，以达到整体的相对最优状态，实现高效率。以下是我们设定的节点优先攻击的准则：

（1）度大原则

度大节点是与其他节点链接更密切的节点，若该类点遭到攻击破坏，那么整个网络势必发生强烈的级联失效现象，因此我们设立度大原则来为度大节点提高优先级。每当到达一个选择时刻，通过计算各节点的度数作为优先级评定的标准之一。以 k_i（i=1,…,N）表示第 i 个节点的度数。示意图如下：

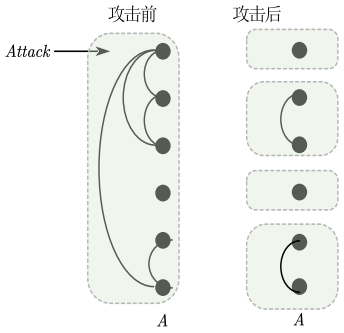


图 5-5 度大原则示意图

（2）关键位置原则

设定了节点的数值（度）与选择优先相关准则后，我们发现，存在一些度小但处在关键位置的节点也应该优先删除，即此时删除关键位置节点比删除度大定点效率更高。示意图如下：

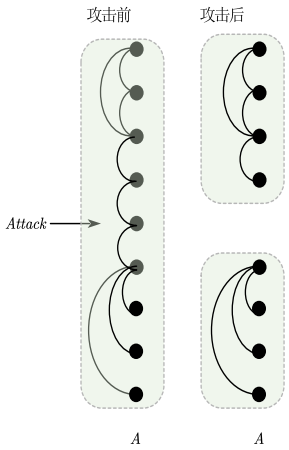


图 5-6 关键位置原则示意图

（3）最大连团原则

随机攻击的最终目标就是使得网络中最大的连通集团数所含节点数最少，那么我们总是优先攻击最大连通集团中的节点。当选择被攻击节点时，总是先识别出当前网络中最大的连通集团，并对该连通集团进行选点攻击。

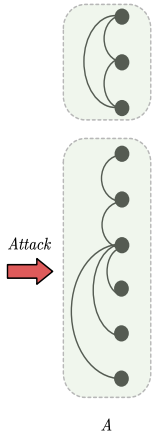


图 5-7 最大连团原则示意图

5.3.2 攻击优化模型

依据最终最大相互连通集团的节点数量 g 最小化为目标函数，以期攻击效率最高。

（1）参数设立

设 m 个节点的攻击顺序$R=(R_1,...,R_m)$，其中 m 为要攻击的节点数量，R_m是节点在攻击中的序号，u，v表示整个相互作用网络中所有节点的编号。我们的最终目标是找一个可行的攻击方案使得最终最大相互连通集团的节点数量$g_{\max}$最小。

$$x_{uv}=\begin{cases}1,\text{节点}u\text{在节点}v\text{之前被攻击}\\0,\text{节点}u\text{在节点}v\text{之后被攻击}\end{cases} \tag{11}$$

（2）建立混合整数规划模型

$$\min\ g_{\max}$$

$$\begin{cases} \sum_{u=1}^{2n} x_{uv} = 1 & u \neq v, v = 1,...,2N \\ \sum_{v=1}^{2n} x_{uv} \leqslant 1 & u \neq v, u = 1,...,2N \\ k_u + |k_u - k_v|(1 - x_{uv}) \geqslant k_v & \\ g_j = \sum_{i=1}^{N} s_{ij} & \\ g_{\max} \geqslant g_j & \\ s_{ij} \in \{0,1\} & \\ k \in \{1,...,N\} & \end{cases} \tag{12}$$

通过上述约束严格执行，以求目标最优。

5.3.3 算法的建立与求解

在相互作用网络中，攻击节点的选择方案多种多样，且级联失效后产生的最大相互连通集团中节点数量的优化常常涉及大量的信息，考虑到建立算法的复杂度，我们选择能够解决该问题且复杂度最小的蒙特卡洛模拟法。基于大数定律，样本数量越多，则得到的答案就会有越高的概率接近期望值，当取足够多模拟值时蒙特卡洛模拟法完全可以求得全局最优解。

（1）算法描述

结合上述描述，我们根据第二问中给出的度大原则、关键位置原则和最大连团原则建立如下算法：

Step 1：按照度数大小对节点降序排序，并返回存下标的向量 idx，初始化每个 1-50 攻击节点数的最优解 res 为无穷大，定义参数 N，即在度数前 N 大的节点中随机生成攻击点的序列，M 即迭代的次数。

Step 2：根据设定的参数 M 开始循环迭代。

Step 3：利用 randperm 生成前 N 大的节点编号的全排列，选取前 m 个作为攻击的节点。

Step 4：根据要攻击点的序列调用 attack 函数对节点进行攻击

Step 5：对攻击后得到的邻接矩阵进行级联失效

Step 6：将 res 赋值为当前答案和全局变量 res 的最小值

Step 7：达到边界停止，否则重复 Step2、3、4、5、6。

（2）算法流程图：

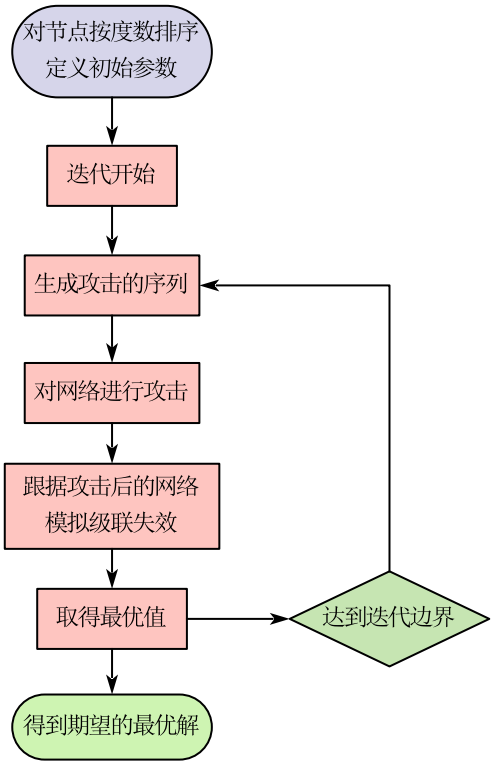
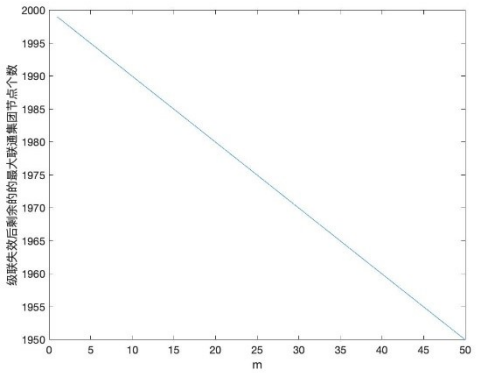


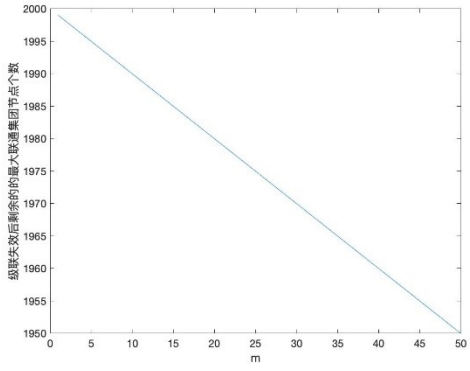
图 5-8 问题三算法流程图

（3）求解与结果分析

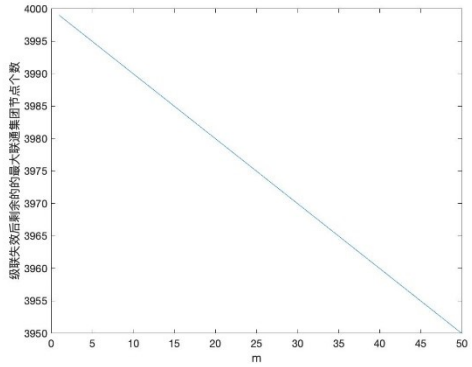
严格执行上述算法，得出 4 个相互作用网络的 m 和 g 关系图：



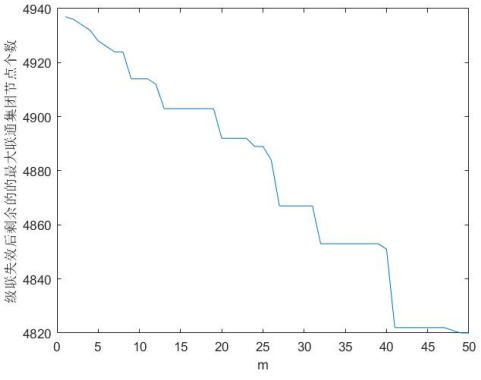
(a)



(b)



(c)



(d)

图 5-9 四个相互作用网络的 m-g 关系图

如图 5-9 四图所示，四个网络均不存在攻击阈值，网络 1、2、3 的稳定性非常好，说明这三个网络中节点间联系具有多元性且密切，网络 4 的图线具有跌宕性，说明网络 4 中存在有级联失效隐患的度大节点，该网络稳定性较差。

5.4 问题四的模型建立与求解

5.4.1 遗传算法的建立

（1）算法描述：

种群大小 M，迭代次数 C，交叉率 Pc=0.8，编译概率 Pmutation=0.2，节点个数 N，要攻击的节点个数 cnt

1）编码策略

采用实数编码，一个值域在[1,N]中大小为 cnt 的序列表示要攻击的点的集合，

2）初始种群

即在 1~N 中随机选取当前要攻击的节点个数 cnt 个节点作为要攻击的节点，例如 cnt=5，N=100，其中一个染色体为[2,78,64,32,17]

（3）适应度函数

在本问题中，已知每个染色体，即要攻击的节点集合，即可求出最大的联通集团大小，最大联通集团大小以其连通块内部点的个数计算，结合目标函数，其值越小越优。

（4）选择操作

采用轮盘赌法，各个个体被选择的概率与其适应度成比例。

（5）交叉操作

随机选择两个个体，在对应位置交换若干个基因片段，同时保证节点不重复

（6）变异操作

对于变异操作，随机选取个体，同时随机选取个体的两个基因来实现变异

遗传算法流程图

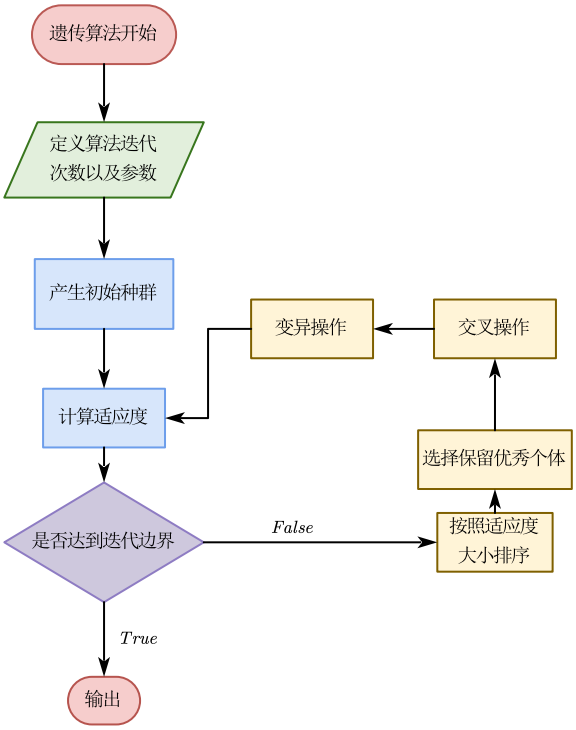


图 5-10 遗传算法流程图

5.4.2 基于 AHP 的算法评价模型：

对第三问的蒙特卡洛和遗传算法模型的评价，我们选择了层次分析法来建立模型解决问题。层次分析法（AHP）是对定性问题进行定量分析的一种简便、灵活而又实用的多准则决策方法。它适合我们进一步量化各个指标之间的重要性程度。

（1）指标的选择与比较

1）指标一：时间复杂度

每次实现攻击操作的时间复杂度为 $O(V^2)$，进行级联失效的时间复杂度为 $O(V^2)$，matlab 内置函数 conncomp 的时间复杂度相较于攻击和级联失效的复杂度可以视作常数，在考虑渐进时间复杂度时可忽略，最终攻击操作和级联失效操作的渐进时间复杂度约为 $O(V^2)$。

在第三问的算法和以上给出的启发式搜索算法：遗传算法在同样攻击节点数量为 x 的时候的收敛速度曲线可以看到，相对于依靠大数定律的蒙特卡洛概率模型，遗传算法可以有更高的收敛速度，并且根据交叉算法，在陷入局部最优解时有一定概率跳出局部最优，向全局最优收敛，依靠蒙特卡洛和度大原则、关键节点原则的算法迭代次数要在 10^6 以上才能说明得到的解趋近于最优值，而遗传算法在迭代 500 次的时候就趋近最优值，所以蒙特卡洛算法的总时间复杂度约为 $O(10^6 \cdot V^2)$,遗传算法的总时间复杂度约为 $O(500 \cdot V^2)$，所以二者从时间复杂度上来看，遗传算法要远优于蒙特卡洛算法。

2）指标二：空间复杂度

遗传算法在迭代的过程中相较蒙特卡洛算法需要额外的开一个空间用于存储当前选择的染色体以及交叉、变异后产生的染色体

3）指标三：正确性

蒙特卡洛算法是建立在大数定理基础之上的，是一个有一定概率取得最优值的算法，但是具体能不能取得最优值，有很大的随机性，而遗传算法通过选取和是的适应度计算、选择、交叉、变异方法可以保证解趋近最优解。

4）指标四：健壮性

蒙特卡洛算法的迭代是通过随机生成攻击序列，而遗传算法的迭代是上一代经过选择的较优值，在适应度函数选择不当的情况下有可能收敛于局部最优值，而不能达到全局最优。而蒙特卡洛是期望能够达到最优解。

（2）模型的求解：

Step1：确定指标集合

对于算法的评价，大致是从五个方面进行评价，分别是时间复杂度、空间复杂度、正确性、可读性、健壮性来进行的评价。其中可读性对与问题无关，忽略不计。

Step2：建立判断矩阵

对于每个算法的每个指标，优于待评价集合只有 2 个，采取 10 分制进行打分

Step3：对判断矩阵进行一致性检验

在求取权重之前，我们为每个判断矩阵计算了一致性指标 CI，并查找出对应的平均随机一致性指标 RI，计算出每个判断矩阵的一致性比例 CR，以此来检测我们判断矩阵的可靠性。

$$\text{一致性指标：}\quad CI=\frac{\lambda_{\max}-n}{n-1} \tag{13}$$

$$\text{一致性比例：}\quad CR=\frac{CI}{CR}\begin{cases}<0.1,\text{一致性检验通过}\\ \geq 0.1，\text{未通过一致性检验}\end{cases} \tag{14}$$

表 5-4：平均随机一致性指标 RI

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0	0	0.52	0.89	1.12	1.26	1.36	1.41	1.46	1.49	1.52	1.54	1.56	1.58	1.59

Step4：计算两个待选目标在各个中综合权重

层次分析法流程图

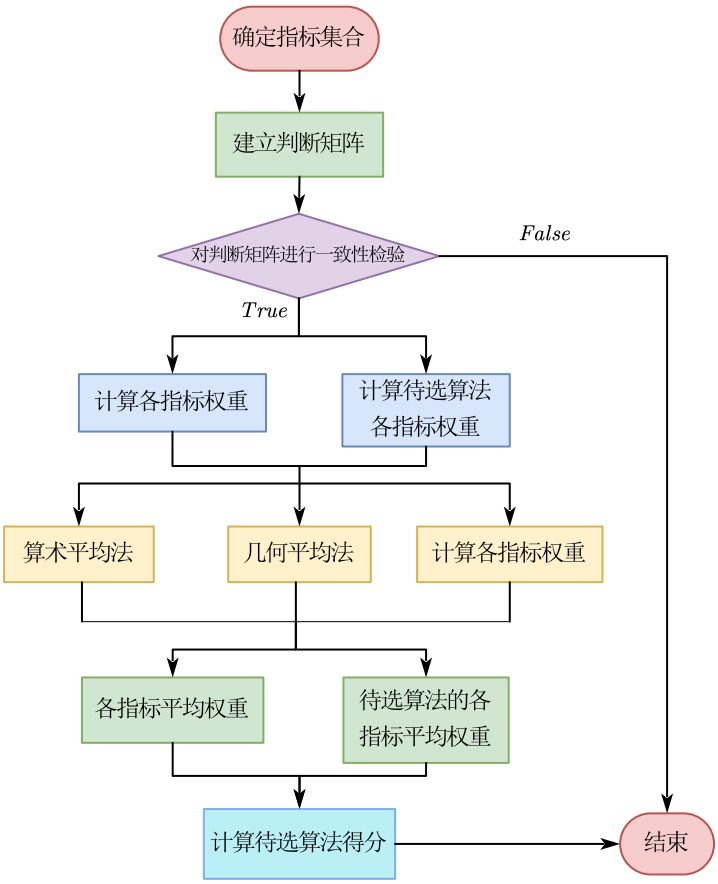


图 5-11 AHP 流程图

综合权重的计算方法：

1、算数平均法

将判断矩阵按照列归一化，然后将归一化的各列相加(按行求和)，最后将相加后得到的向量中每个元素除以 n 即可得到算数平均法的权重向量。

$$\omega_i = \frac{1}{n}\sum_{j=1}^{n}\frac{a_{ij}}{\sum_{k=1}^{n}a_{kj}} \tag{15}$$

2、几何平均法

将 A 的元素按照行相乘得到一个新的列向量，再将新的向量的每个分量开 n 次方，对该列向量进行归一化即可得到集合平均法的权重向量。

$$\omega_i = \frac{\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}}}{\sum_{k=1}^{n}\left(\prod_{j=1}^{n} a_{kj}\right)^{\frac{1}{n}}} \quad (i=1,2,...,n) \tag{16}$$

3、特征值法

求出矩阵 A 的最大特征值以及其对应的特征向量，再对求出的特征向量进行归一化即可得到特征值法的权重。

为了保证结果的稳健性，以及评价的合理性，我们分别用三个方法求出综合权重求一个均值作为两个算法评价的综合权重。

Step5：根据综合权重做出评价

表 5-5 蒙特卡洛算法

蒙特卡洛算法	遗传算法
0.3406	0.6595

从图 5-4 中可以看出，遗传算法的综合得分最高，说明遗传算法的效率更高。由于篇幅的原因，我们将各判断矩阵及其权重指标表放在了附录上。

六、模型评价与推广

6.1 模型的优点

优点一：在例子的研究基础上，重新定义了一次性攻击所有要被攻击点的删边规则，大大减少了迭代次数，提高了算法的收敛速度。

优点二：问题二中选择网络时根据度、平均度、度分布三个属性，综合评选出级联失效现象最明显的 data 4，问题考虑较为全面。

优点三：设定节点攻击优先原则：度大原则、关键位置原则及最大连团原则，进行优化模型的建立，具有一定的创新性。

优点四：论文中相关图表较多，直观性较强。

6.2 模型的缺点

没有考虑实际存在的蓄意攻击情形，对模型假设较多，将攻击情况设置得较为理想化。

6.3 模型的推广

本文是对相互作用网络中级联失效问题进行了分析与评价，我们可以将其推广至通信-电力网络等随机故障问题以及关键基础设施级联失效等问题，并加以改进，具有很强的现实意义。

七、参考文献

[1]Buldyrev,S.V.,Parshani,R.,Paul,G.,Stanley,H.E.,Havlin,S.Catastrophic cascade of failures in interdependent networks(2010)Nature,464(7291),pp. 1025-1028.

[2]彭兴钊,姚宏,张志浩,杜军.基于节点蓄意攻击的无标度网络级联抗毁性研究[J].系统工程与电子技术,2013.09

[3] Albert-László Barabási. 巴拉巴西网络科学(Network Science).河南科学技术出版社,2020.02

[4]司守奎,孙兆亮.数学建模算法与应用(第二版).国防工业出版社.2017.01

附录

一、代码部分

attack.m 函数

```
function [matA,matB] = attack(mata,matb,atk)
    atk_len=size(atk,2);
    mat_len=size(mata,1);
    for i=1:atk_len
        for j=1:mat_len
            mata(atk(i),j)=0;
            mata(j,atk(i))=0;
            matb(atk(i),j)=0;
            matb(j,atk(i))=0;
        end
    end
    matA = mata;
    matB = matb;
end
```

cal.m 函数

```
function [mA,mB,res] = cal(matA,matB)
    gA=graph(matA);
    gB=graph(matB);
    len = size(matA,1); %A 和 B 共同的点的数量
    binA = conncomp(gA);
    binB = conncomp(gB);


    binA=getbin(binA,len); %求出 A 变化后的连通分量
    binB=getbin(binB,len); %求出 B 变化后的连通分量


    A=matA;
    B=matB;

    for i = 1:len
        if binA(i) ~= binB(i)
            for j=1:len
                cif A(i,j) ~= B(i,j)
                    A(i,j)=0;
                    A(j,i)=0;
                    B(i,j)=0;
                    B(j,i)=0;
                end
```

```
            end
            gA=graph(A);
            gB=graph(B);
            binA = conncomp(gA);
            binB = conncomp(gB);
            binA=getbin(binA,len);
            binB=getbin(binB,len);
        end
    end

    ans_gA=graph(A);
    ans_gB=graph(B);
    ans_binA = conncomp(ans_gA);
    ans_binB = conncomp(ans_gB);
    ans_binA=getbin(ans_binA,len);
    ans_binB=getbin(ans_binB,len);

    rec=zeros(1,len);
    for i=1:len
        rec(ans_binA(i))=rec(ans_binA(i))+1;
    end
    res=-inf;
    for i=1:len
        res=max(res,rec(i));
    end
    mA=A;
    mB=B;
end
```

getbin.m 将连通块代表设置为内部的最大编号节点

```
function [result] = getbin(bin,size)
    record = 1./zeros(1,size); %初始化为无穷大
    cnt=-inf;

    for i=1:size
        cnt=max(cnt,bin(i));
    end


    for i=1:size
        record(bin(i))=i;
    end

    for i=1:size
        record(bin(i))= max(record(bin(i)),i);
```

```
    end
    res=zeros(1,size);
    for i=1:size
        res(i) = record(bin(i));
    end
    result=res;
end


```

main1.m 第一问主函数

```
clc;
clear;
close all;

a = xlsread('A1.xlsx');%读取数据
b = xlsread('A2.xlsx');

atk= 1; %确定要攻击的点
[a,b]=attack(a,b,atk);%调用函数执行攻击
[A,B,answer] = cal(a,b); %调用函数实现求最大连通分支
answer
[a,b]=attack(a,b,atk);%调用函数执行攻击

g1=graph(A);
g2=graph(B);
figure,plot(g1,'NodeColor','k','EdgeAlpha',0.6);  %可视化当前网络。
hold on
% figure,plot(g2,'NodeColor','k','EdgeAlpha',0.6);  %可视化当前网络。
% hold on
```

average_d.m 求出网络的平均度

```
clc;
clear;
load('data1.mat');
% load('data2.mat');
% load('data3.mat');
% load('data4.mat');
a=A1;b=A2;
len=size(A1,1);
d=0;
for i=1:len
    for j=1:len
        if i~=j && a(i,j)
```

```
            d=d+1;
        end
    end
end

for i=1:len
    for j=1:len
        if i~=j && b(i,j)
            d=d+1;
        end
    end
end
d
len=len*2
d=d/len
```

main2.m 第二问主函数

```
clc;
clear;
close all;
% load('data1.mat');
% load('data2.mat');
% load('data3.mat');
load('data4.mat');
a=A1;b=A2;
len=size(a,1);

N=10;
p=rand(1,N);
count=round(len.*p);
r = zeros(1,N);
% res = zeros(1,N);
backupa=a;
backupb=b;
count = sort(count,2);

for i=1:N % 换成 size
    for j=1:100
        atk = randperm(len);
        atk = atk(1:count(i)) %确定随机攻击的点

        [a,b]=attack(a,b,atk);%调用函数执行攻击
        [A,B,answer] = cal(a,b); %调用函数实现求最大连通分支
        r(i)= r(i)+ answer/len;
        a=backupa;
```

```
        b=backupb;
    end
    r(i)=r(i)/100;
end
for i=1:N
    p(i)=count(i)/len;
end
figure;
hold on
plot(p,r);
title('p-r 关系图');
xlabel('p');
ylabel('r');
```

plot_.m 网络可视化

```
clc;
clear;
close all;
load('data1.mat');
h1 = graph(A1);
figure(1);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 1-1');

h1 = graph(A2);
figure(2);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 1-2');

load('data2.mat');
h1 = graph(A1);
figure(3);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 2-1');
h1 = graph(A2);
figure(4);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 2-2');
load('data3.mat');

h1 = graph(A1);
figure(5);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 3-1');
h1 = graph(A2);
```

```
figure(6);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 3-2');
load('data4.mat');
h1 = graph(A1);
figure(7);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 4-1');
h1 = graph(A2);
figure(8);
plot(h1,'NodeColor','k','EdgeAlpha',0.1);
title('网络 4-2');
```

d .m 求出四个网络的度分布图

```
clc;
clear;

% load('data1.mat');
% load('data2.mat');
% load('data3.mat');
load('data4.mat');
sz=size(A1,2);

d=zeros(1,sz);

for i=1:sz
    for j=1:sz
        if A1(i,j)==1
            d(i)=d(i)+1;
        end
    end
end

for i=1:sz
    for j=1:sz
        if A2(i,j)==1
            d(i)=d(i)+1;
        end
    end
end

d_=sort(d);
d_=unique(d_);
cnt=size(d_,2);
idx=zeros(1,max(d_));
```

```
32.
33. for i=1:cnt
34.     idx(d_(i))=i;
35. end
36. time=zeros(1,cnt);
37.
38. for i=1:sz
39.     time( idx(d(i)) ) = time(idx(d(i)))+1;
40. end
41. figure(1);
42. plot(d_,time);
43. % title('网络一 节点-度数分布图');
44. % title('网络二 节点-度数分布图');
45. % title('网络三 节点-度数分布图');
46. title('网络四 节点-度数分布图');
47. xlabel('度数');
48. ylabel('节点个数');
```

main3.m 问题三主函数

```
1.  clc;
2.  clear;
3.  close all;
4.  load('data1.mat'); % 假设附件 1 中的数据解压缩在当前路径下。
5.  a=A1;b=A2;
6.  len=size(a,1);
7.
8.  R=1000;
9.  N=50;
10. backupa=a;
11. backupb=b;
12. min_gmax=zeros(1,N);
13. for i=1:N
14.     min_gmax(i)=+inf;
15. end
16.
17. d=zeros(1,len);
18.
19.
20. for i=1:len
21.     for j=1:len
22.         if i~=j && (a(i,j) || b(i,j))
23.             d(i)=d(i)+1;
24.         end
25.     end
26. end
```

```
[d,idx]=sort(d,'descend');
l=N*3;
idx_=idx(1:N*4);

for m=1:N % 换成 size
    fprintf('攻击%d 个点\n',m);
    for j=1:R % 生成 R 次具体攻击的节点 min_gmax
        fprintf('攻击%d 个点,第%d 次迭代\n',m,j);
        vector = randperm(l);
        A = zeros(size(idx_,2));
        for i=1:l
            A(i)=idx_(vector(i));
        end
        atk = A(1:m); %确定随机攻击的点
        [a,b]=attack(a,b,atk);%调用函数执行攻击
        [A,B,answer] = cal(a,b); %调用函数实现求最大连通分支
        if min_gmax(m)>answer
            min_gmax(m) = answer;
        end
        min_gmax(m)=min(min_gmax);
        a = backupa;
        b = backupb;
    end
end

m=zeros(1,50);
for i=1:50
    m(i)=i;
end
figure(1);
plot(m,min_gmax);
xlabel('m');
ylabel('级联失效后剩余的的最大联通集团节点个数');
```

AHP 层次分析法

```
clc;
clear;
% A=xlsread('指标判断矩阵.xlsx');
% A=xlsread('时间复杂度对比.xlsx');
% A=xlsread('空间复杂度对比.xlsx');
% A=xlsread('正确性对比.xlsx');
% A=xlsread('鲁棒性对比.xlsx');

[n,n] = size(A);
%算术平均法求权重
```

```
Summery_A = sum(A);
SUMMERY_A = repmat(Summery_A,n,1);
Stand_A = A ./ SUMMERY_A;
x1=sum(Stand_A,2)./n
%几何平均法求权重
P_A = prod(A,2);
P_n_A = P_A .^ (1/n);
x2=P_n_A ./ sum(P_n_A)
%特征值法求权重
[V,D] = eig(A);
Max_eig = max(max(D));
[r,c]=find(D == Max_eig , 1);
x3=V(:,c) ./ sum(V(:,c))
%计算一致性比例 CR
CI = (Max_eig - n) / (n-1);
RI=[0 0.0001 0.52 0.89 1.12 1.26 1.36 1.41 1.46 1.49 1.52
1.54 1.56 1.58 1.59];
CR=CI/RI(n);
disp('一致性指标 CI=');disp(CI);
disp('一致性比例 CR=');disp(CR);
if CR<0.10
    disp('CR<0.10，Accept');
else
    disp('CR >= 0.10，Wrong');
end
res=x1+x2+x3
res=res./3
```

GA.m 遗传算法的实现

```
clear;
clc;
%参数定义
M=20;                %种群的个数
C=100;               %迭代次数
```

```
C_old=C;
m=2;                    %适应值归一化淘汰加速指数
Pc=0.8;                 %交叉概率
Pmutation=0.2;          %变异概率

res=zeros(1,50);
for i=1:50
    res(i)=+inf;
end
load('data1.mat');
% load('data2.mat');
% load('data3.mat');
% load('data4.mat');
sz = size(A1,1);
backupa=A1;
backupb=A2;
min_ans =zeros(1,50);
for i=1:sz
    atk = i;
    [a,b]=attack(A1,A2,atk);%调用函数执行攻击
    [A,B,answer] = cal(a,b); %调用函数实现求最大连通分支
    res(1)=min(res(1),answer);
    A1=backupa;
    A2=backupb;
end
for cnt=2:50
    fprintf('攻击%d 个点\n',cnt);
    %生成初始群体
    popm=zeros(M,cnt);
    for i=1:M
        atk = randperm(sz);
        popm(i,:)=atk(1:cnt);
    end
    %初始化种群及其适应函数
    fitness=zeros(M,1); %初始化所有的适应度为 0
    len=zeros(M,1);
    for i=1:M
        len(i,1)=myLength(A1,A2,popm(i,:));
    end
    maxlen=max(len);
    minlen=min(len);
    fitness=len;
    rr = find(len==minlen);
    R = popm(rr(1,1),:);

```

```
    fitness = fitness/sum(fitness)
    try_res=sum(fitness)
    distance_min=zeros(C,1);  %%各次迭代的最小的种群的距离
    %开始迭代
    for C_=1:C
        fprintf('迭代第%d 次\n',C_);
        %%%%选择操作%%%%
        nn=0;
        while nn<10
            for i = 1:size(popm,1) %遍历所有种群
                len_1(i,1) = myLength(A1,A2,popm(i,:));
                jc = rand*0.2;
                for j = 1:size(popm,1)
                    if fitness(j,1) > jc
                        nn = nn+1;
                        popm_sel(nn,:) = popm(j,:);
                        break;
                    end
                end
            end
        end
        %每次选择都保存最优的种群
        popm_sel=popm_sel(1:nn,:);
        [len_m len_index]=min(len_1);
        popm_sel=[popm_sel;popm(len_index,:)];

        %交叉操作
        nnper = randperm(nn);
        idx1=nnper(1);
        idx2=nnper(2);

        A = popm_sel(idx1,:);
        B = popm_sel(idx2,:);

        for i=1:nn*Pc
            [A,B]=cross(A,sz);
            popm_sel(idx1,:)=A;
            popm_sel=[popm_sel;B];
        end

        %变异操作
        for i=1:nn
            pick=rand;
            while pick==0
                 pick=rand;
```

```
            end
            if pick<=Pmutation
               popm_sel(i,:)=Mutation(popm_sel(i,:),sz);
            end
         end

         %求适应度函数
         NN=size(popm_sel,1); %染色体个数

         len=zeros(NN,1);
         for i=1:NN
             len(i,1)=myLength(A1,A2,popm_sel(i,:));
         end
         maxlen=max(len);
         minlen=min(len);
         distance_min(C_,1)=minlen;
         fitness=fit(len,m,maxlen,minlen);
         popm=[];
         popm=popm_sel;
     end
     res(cnt)=min(distance_min);
end

figure(1)
x=zeros(1,C);
for i=1:50
     x(i)=i;
end
res(cnt)=min(distance_min);
plot(x,res);
xlabel('m');
ylabel('级联失效后剩余的的最大联通集团节点个数');
```

myLength.m 计算目标函数值

```
function len=myLength(A1,A2,atk)
    [a,b]=attack(A1,A2,atk);%调用函数执行攻击
    [~,~,answer] = cal(a,b); %调用函数实现求最大连通分支
    len=answer;
end
```

Mutation.m 变异函数

```
function a=Mutation(A,sz)
    record = zeros(1,sz);
    for i=1:size(A,2)
        record(A(i))=1;
    end
    idx=0;
    replace=randperm(sz);
    x = randperm(size(A,2));
    idx=x(1);
    for i = 1 : sz
        if record(replace(i)) ~= 1
            A(idx)=replace(i);
            break;
        end
    end
    a=A;
end
```

fit.m 适应度函数

```
function fitness=fit(len,m,maxlen,minlen)
   fitness=len;

   for i=1:length(len)
       fitness(i,1)=(1-(len(i,1)-minlen)/(maxlen-minlen));
   end
nd
```

exchange.m 交换函数

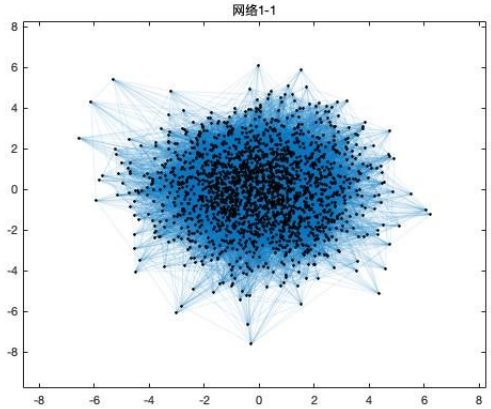
```
function [x,y]=exchange(x,y)
    temp=x;
    x=y;
    y=temp;
end
```

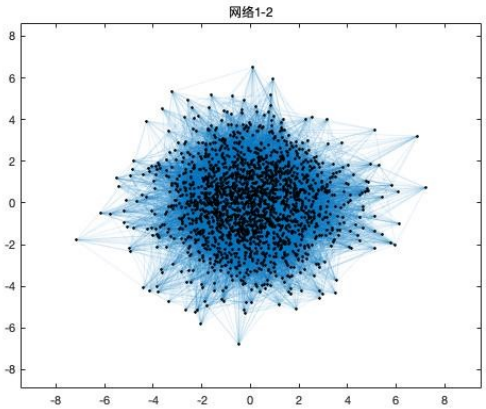
cross.m 交叉函数

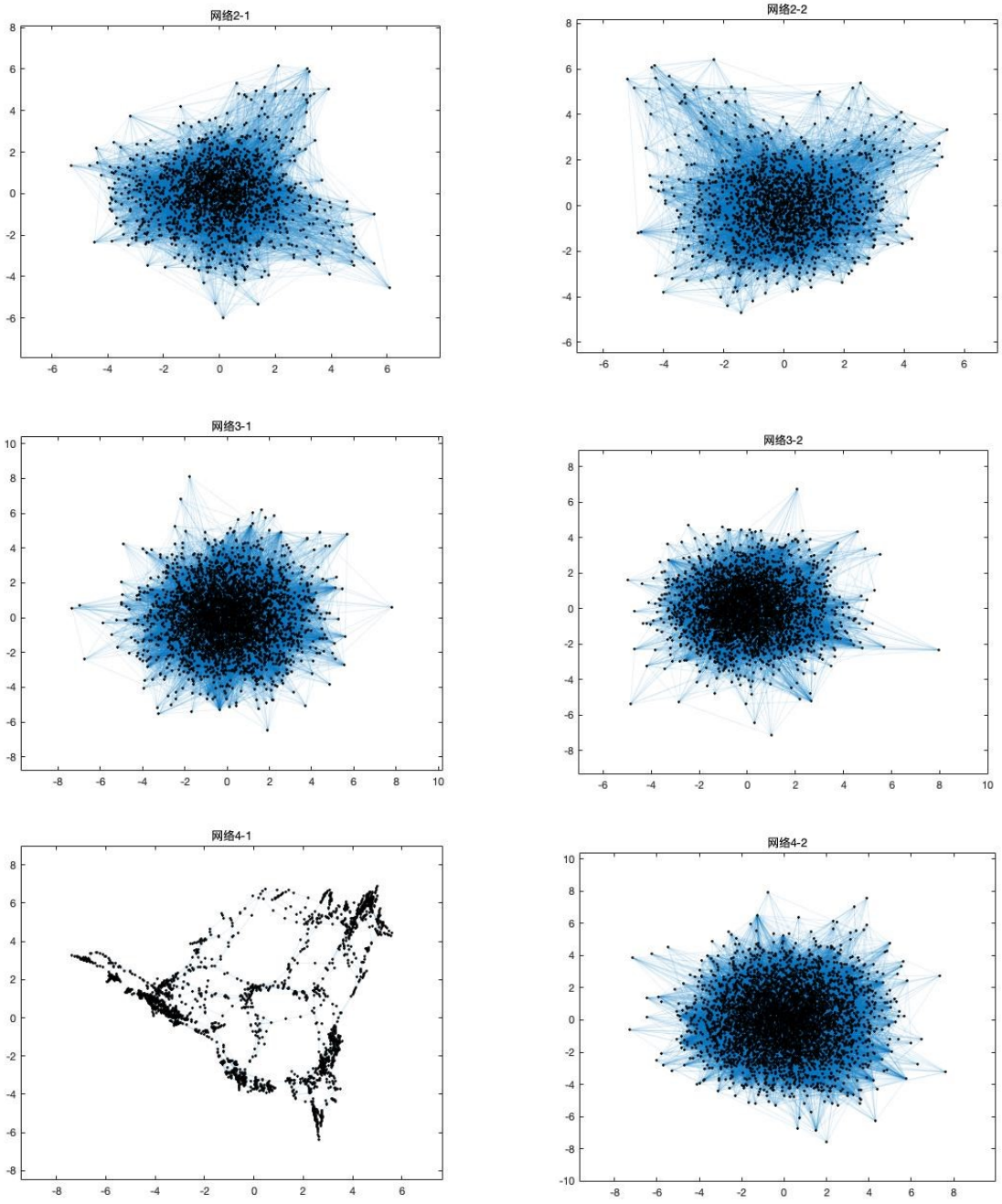
```
function [A,B]=cross(A,sz)

    cnt=size(A,2);
    do=floor(cnt/2);
```

```
    fprintf('cnt : %d\n',cnt);
    fprintf('do : %d\n',do);
%     W=floor(L/de)+8;
    reca=zeros(1,sz);
    recb=zeros(1,sz);
    B = randperm(sz);
    B = B(1:cnt);

    for i=1:cnt
       reca(A(1,i))=1;
    end
    for i=1:cnt
       recb(B(1,i))=1;
    end
    for i=1:do
       fprintf('i : %d   cnt -i +1 : %d\n',i,cnt-i+1);
        if A(1,i) ~= B(1,cnt-i+1) && recb(A(i)) ~= 1 && reca(B,cnt-i+1) ~= 1
            [A(i),B(cnt-i+1)] = exchange(A(i),B(cnt-i+1));
        end
    end

end
```

二、附图部分







附图 1 所给网络可视化图