基于 KSP 算法的图网络内部结构诊断研究

付廷琛 2018202150*

中国人民大学信息学院

2020 年 7 月

摘要

本研究实现和拓展了 Network Tomography 算法。借助于 Yen's algorithm 实现了寻求两点之间无圈最短路的 k-shortest -path 算法，以及利用最小二乘法估计非满秩矩阵方程的解，在数据集 Domestic Network 上进行重复多次实验，将实验结果进行对比分析，并得到的选择 monitor 节点的一些探索性的方法。最后给图网络的故障勘探和建设提出了一些可行的建议。

*电子邮件: lucas.futingchen@gmail.com

1 问题背景

1.1 问题描述

随着近几年互联网规模呈指数级别的迅速扩大以及网络内部结构的日益复杂多元化，对于大型网络结构的勘探和维护工作的难度挑战越来越大，迫切需要一种可以高效率并且准确的诊断定位技术，可以在发生故障时及时的判断网络内部的层次结构以及延迟率、丢包率等重要信息。由此应运而生的 Network Tomography(网络断层成像技术)，是一种通过在网络的外围设置若干监控装置（又称 monitor），利用这些监控装置进行彼此之间的信息发送传递，由此推断出网络内部的丢失率、延迟分布、流量矩阵等信息的技术。我们希望通过 monitor 之间的信息传递，测量其发送端和接收端之间的延迟时间 Y，从而计算推断出网络之中每一条边的延迟 X，使得满足测量矩阵方程 Y=AX

1.2 文献综述

Teresa Pepe, Marzio Puleri 等人提出了解决该问题的一种基本算法。首先确定若干对可供使用的 monitor 节点作为测量节点，对于每一个测量节点对（m1,m2），通过 Yen ‘s algorithm 的 k-最短路算法找到两个 monitor 之间的所有路径并按照长度由小到大进行排序。依次检查这些待选路径。如果这条路径与所有已经被选择的路径之间都是线性独立的，也就是说，将这条路径加入到依赖矩阵之中会使得依赖矩阵的秩增加，那么将这条路径选中并加入到依赖矩阵，否则舍弃。直至依赖矩阵达到满秩方阵的状态时算法终止。接下来通过求解矩阵方程来获得 X。我的研究就是基于这两人的工作展开的。除此之外，K. Claffy, T. Monk 等人着重研究了依赖矩阵的可识别性，提出了在 A 不满秩的情况下一种处理方法；A. Chen J. Cao 使用傅里叶变换提出了一种新的测量方法并降低了算法的复杂度；Jiaqi Gui, Vahid Shah-Mansouri 使用线性代数的方法对依赖矩阵进行分块处理并对子矩阵进行了进一步的研究。

1.3 我的工作

我使用 python 自己实现了 Yen’s algorithm 的 k-最短路算法，通过确定一系列的 monitor 得到依赖矩阵，然后使用线性代数中的最小二乘法来估计 X。改变 monitor 的个数和选择方式，进行多组重复测量实验，根据实验得到的结果尝试发现和总结测量准确率同测量节点选择之间的关系。本文的第一部分是对问题的简要介绍和对已经工作的回顾，第二部分是 KSP 算法的原理以及实现细节，第三部分是对最小二乘法原理的说明，第四部分是实验结果的说明解释，最后一部分是文章的结论。

2 KSP 算法

KSP（k-shortest-path）问题是指在图中寻找两点之间的 k 条最短路径。依据是否允许路径之中有圈，可以将 KSP 问题的算法分为分为两大类，并有标号法、偏离路径法、改进 Dijkstra 算法等多种算法。Yen’s algorithm 采用的是处理无圈 k 最短路的偏离路径算法，通过路径之间的迭代找到要求的两点之间 k 条最短路径。这里实现的是 Yen’s algorithm 算法。

2.1 算法原理

偏离路径路径算法，核心概念就是偏离点。假设寻求的是从 s 点出发，到达 t 点的最短路径，这 k 条路径之间必然有公共点，也就进而有最长公共子路径。假设路径 p_2 是由 p_1 迭代产生的，p_1 和 p_2 具有的从 s 开始的最长公共路径为 $s \to v_i$，那么就称 v_i 为 l_2 相对于 l_1 的偏离点。

偏离路径算法的整体流程如下：首先利用 Dijkstra 算法求出 s 到 t 之间的最短路径，作为 p_1，首先加入结果路径列表 A 之中。然后遍历 p_1 上的每一个点 v_i，依次作为偏离点分别生成一条由 $v_i \to t$ 的最短路，然后将这条路与 $s \to vi$ 进行拼接，成为一条完整的路径，计算他的总权重并加入到待选路径列表 B 之中。等 p_1 所有顶点都作为偏移点之后，从 B 中选择一条总权重最小的最短路，从 B 中移除这一条路径，并加入到结果路径列表 A 之中，是为 p_2。然后以同样的流程，将 p_2 作为基础路径，对 p_2 上除去 t 之外的所有点进行上述操作，得到 p_3，以此类推……

最终算法结束可能有两种情况：找到了 k 条路径 $p_1, p_2, p_3, ..., p_k$ 后终止; 或者是 s 到 t 之间的路径总数小于 k，那么在找到 s 到 t 的所有路径之后终止。

2.2 实现细节

- 为了保证所找到的路径是无圈的，在以 p_i 为基础路径寻找从 vi 到 t 的路径的过程之中，需要在图中去掉在 p_i 中从 s 到 vi 的顶点，使用 Dijkstra 找到路径之后再将这些点重新加入到图中。
- 为了防止路径的重复，A 中所有路径的边 (v_i, v_j) 都要加入到已使用列表，当 v_i 作为偏离点时，不可以选择 (vi, vj) 边，否则会造成路径的重复，但是当 v_j 作为偏离点时，是可以选择 v_i 的
- 如果某一轮迭代完成时，B 中有多条路径同时具有最小权重，那么应当将节点数最少的加入到 A 之中

- 在某一轮迭代中需要去掉的边应当以三元组的形式，专门放到一个列表之中，防止循环结束还原边的时侯，边的权重丢失。

3 最小二乘法

对于实线性方程组 $Ax = y$，在很多情况下，由于无法找到足够数量的线性独立路径，导致该方程可能无解，在这种情况下仍需要获得一个解，那么一个可行的办法是：寻找一个向量 x，使得 Ax 与 y 最近，换句话说用 Ax 去逼近 y，它们之间的距离 $|Ax-y|$ 越小，逼近的效果越好，最小二乘法就是寻求 x 使得达到最小的一种方法。设 $A \in M_{m\times n}(R), y \in R^m$, 记 $W = \mathcal{R}(A)$，则 W 是 R^m 的子空间. 令 $\hat{y}$ 为 y 在 W 上的正交投影，即：

$$\hat{y} = \mathcal{P}_W y \tag{1}$$

则 $\hat{y} \in W$ 于是线性方程组

$$Ax = \hat{y} \tag{2}$$

有解。根据最佳逼近定理，$\hat{y}$ 是 W 中最接近 y 的向量，因此 $\hat{x}$ 是一个最小二乘解当且仅当他是方程组的解。显然，方程组有唯一解的充要条件是 $r(A) = n$。故当且仅当 $r(A) = n$ 时，$Ax = y$ 的最小二乘解唯一。设 $\hat{x}$ 是一个最小二乘解，则 $A\hat{x} = \hat{y}$。由正交分解定理，得 $y - \hat{y} \perp W$，即

$$y - A\hat{x} \in \mathcal{R}(A)^{\perp} \tag{3}$$

又因为 $\mathcal{R}(A)^{\perp} = \mathcal{N}(A^T), y - A\hat{x} \in \mathcal{N}(A^T)$ 因此有

$$A^T(y - A\hat{x}) = 0 \tag{4}$$

于是得 $A^TA\hat{x} = A^Ty$，这说明 $Ax = y$ 的每一个最小二乘解都满足线性方程组

$$A^TAx = A^Ty \tag{5}$$

因此，我们可以计算

$$A^TAx = A^Ty \tag{6}$$

作为估计值，也就是 $x = (A^TA)^{-1}A^Ty$. 在具体实现层面，使用 numpy 的 linalg 包中相关函数对最小二乘法进行实现

4 实验

为了寻找最有效率同时最准确的测量点，我进行了若干组实验。在实验的过程之中发现，monitor 的数量一般只需要两组也就是四个。超过两组之后，多余的 monitor 所发现的道路大概率和之前两组 monitor 所发现的道路是线性相关的，不可以加入到依赖矩

阵之中，没有意义。因此，接下来展示的这几组实验全部都是使用了两组四个 monitor。通过 monitor 所找到的依赖矩阵非常难达到 56×56 的满秩方阵状态，他们的秩往往只能达到 49、50 左右。由于不可逆，所以我们通过上述的最小二乘法得到估计值, 具有一定的误差。在每一组实验中，我统计了估计值和真实值之间的残差，并计算相对误差。

4.1 第一组实验

选择的两组 monitor 分别是：SNFN 同 PHLA 之间通信，CHCG 同 SNAN 之间通信。将第一组实验的结果作为 baseline 来使用比较。得到矩阵秩为 49，做残差图和相对误差占比图如下：

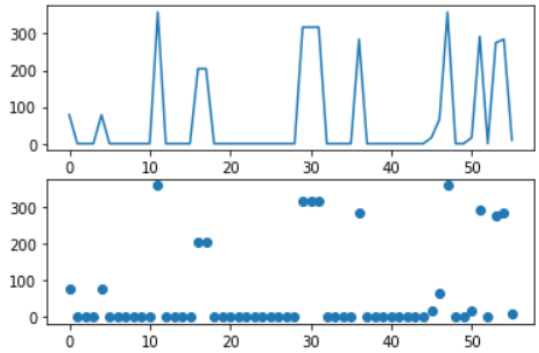

图 1: 第一组实验残差分布

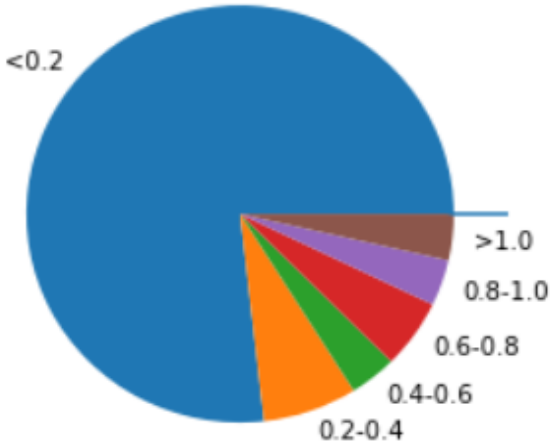

图 2: 第一组实验相对误差占比

4.2 第二组实验

第二次实验选择的 monitor 是：STTL 同 ORLD 之间进行通信，NY54 同 LA03 之间进行通信。得到的矩阵秩为 50，同样做图如下： 可以看到，虽然矩阵的秩仅仅增加了

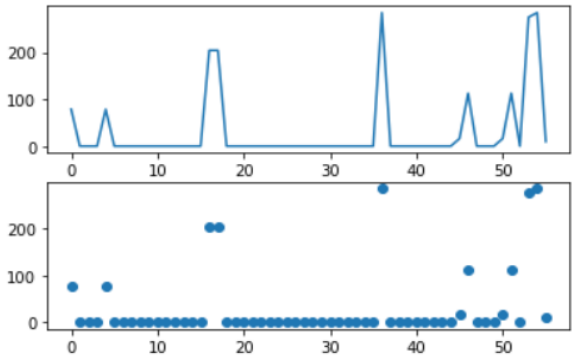

图 3: 第二组实验残差分布

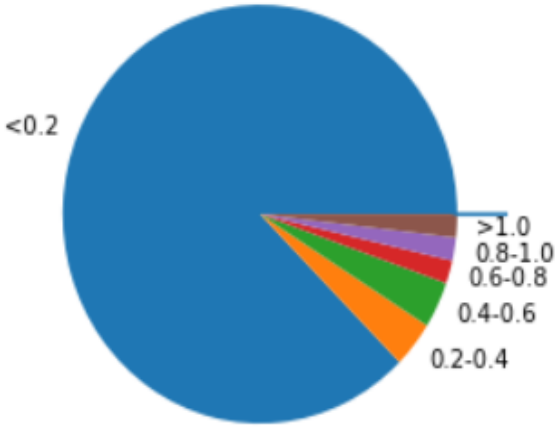

图 4: 第二组实验相对误差占比

1，但是效果有很明显的提升。第一组和第二组的 monitor 选取都是实验性的，大致是观察分布图，从中选择位于边缘上，同时度数相对而言比较大的顶点。

4.3 第三次实验————基于 Betweenness Centrality

第三次实验对于 monitor 的选择尝试了新的方法，首先计算了每一个图中所有顶点的 Betweenness Centrality，其计算公式如下：

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \times cc = \frac{2}{(n-1)(n-2)} \tag{7}$$

其中，$\sigma(s,t|v)$ 是有 v 出现的，从 s 到 t 的最短路径的数量，$\sigma(s,t)$ 是从 s 到 t 的最短路径的总数量。这种方法被称为基于路径的中心度的度量。对各个节点的 Betweenness Centrality 计算结果如下：其中，Betweenness Centrality 最大的四个点是 DLLS，CHCG，

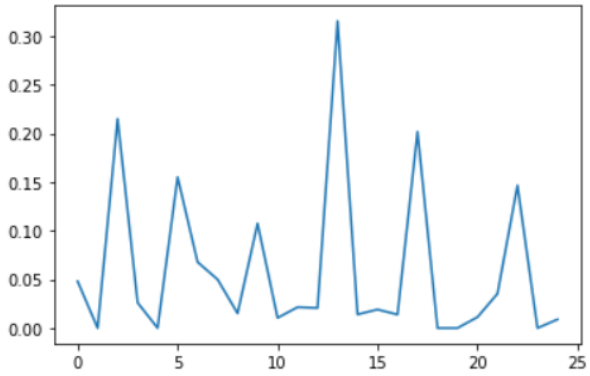

图 5: 各顶点的 Betweenness Centrality

SNFN，ATLN，Betweenness Centrality 最小的四个点是 SCRM，PTLD，SNDG，RLGH。在本组实验中尝试选择 Betweenness Centrality 最小的点作为 monitor，因为他们处于相对边缘化的位置，与他们所关联的边很少被选择。因此如果选择将他们作为路径的起点或者终点，就会不得不使用这些很少被路径覆盖的边，可能对于边的覆盖情况会更加全面，从而有比较好的效果。

作图如下：

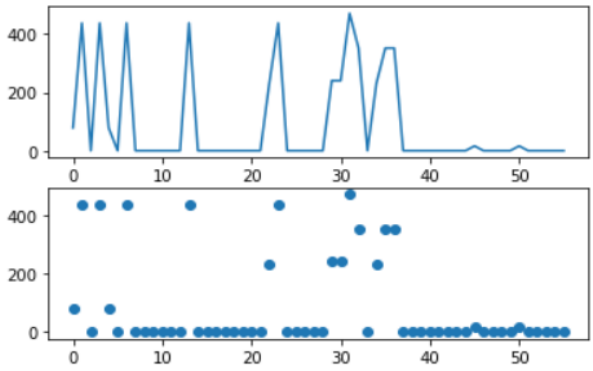

图 6: 第三组实验残差分布

实现效果并不理想

4.4 第四组实验————基于 Betweenness Centrality

由于第三组实验的效果并不是很理想，所以在这次实验之中反其道而行之，尝试选择 Between Centrality 最大的点作为 monitor，最后选择了 CHCG 与 SNFN 通信，DLLS 与 ATLN 进行通信。效果如下

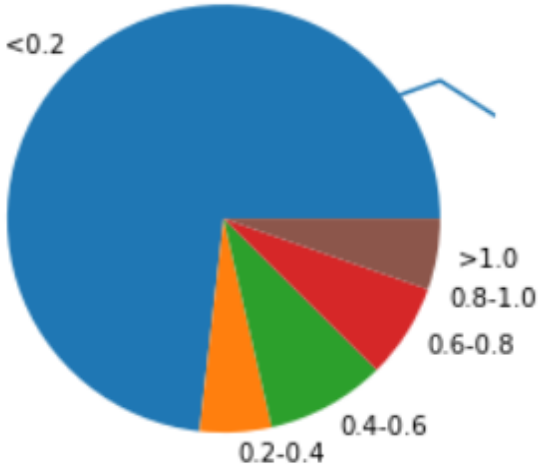

图 7: 第三组实验相对误差占比分布

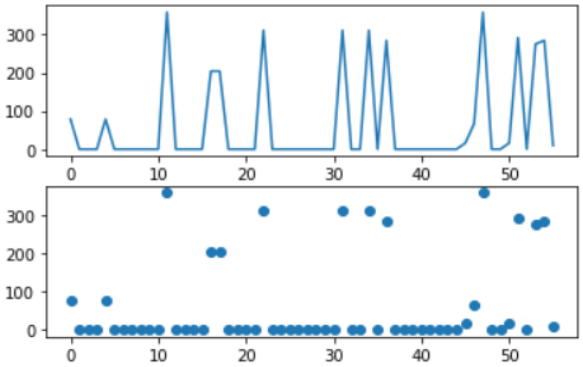

图 8: 第四组实验残差分布

我们发现，无论是选择 Betweenness Centrality 较小的点还是较大的点，实验效果都没有明显的改进。唯一的区别是相对于实验三而言相对误差的分布相对而言更加集中于 0.2-0.4，高相对误差的估计点数量减少了。

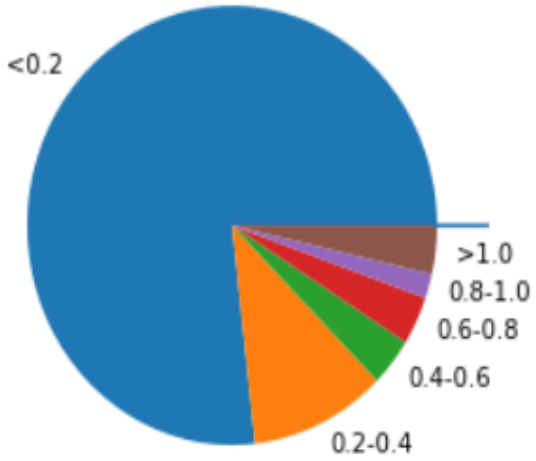

图 9: 第四组实验相对误差占比分布

4.5 第五组实验————基于 Degree Centrality

这次使用 Degree Centrality 作为 monitor 节点选择的准则。Degree Centrality 的定义如下：

$$C_D(v) = \frac{deg(v)}{n-1} \tag{8}$$

将各个顶点的计算结果展示如下图所示：

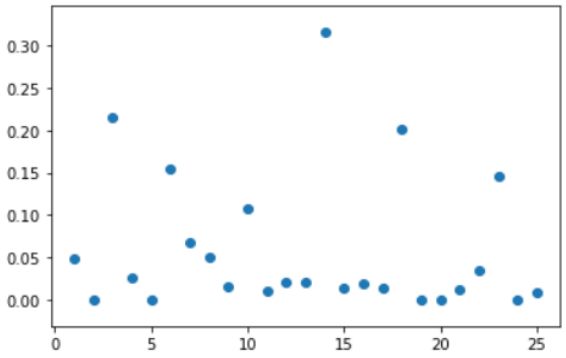

图 10: 各顶点的 Degree Centrality

这次我们优先选择的是 Degree Centrality 最大的顶点。由此得到的 monitor 是：DLLS 同 CHCG 通信，SNFN 同 LA03 进行通信。计算结果如下图所示：

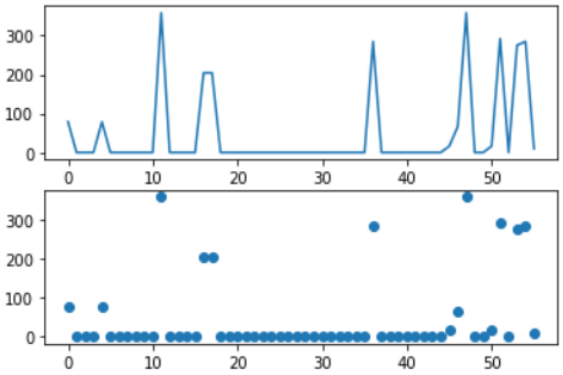

图 11: 第五组实验残差分布

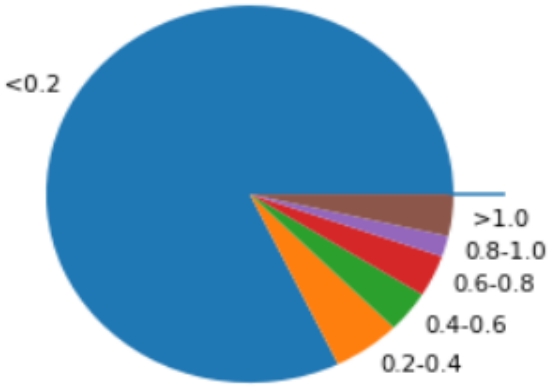

图 12: 第五组实验相对误差占比分布

4.6 第六组实验————Hybrid Method

综合前面几组实验，我们可以发现，除了第三组之外，第一、二、四、五组实验的计算结果中，计算错误的边重合率是非常高的。比如说 16、17、36、53、54 这几条边在上面几组实验之中往往都会出现很大的误差，严重偏离了真实值。而只有第三组是例外的。其实这不难理解，因为除去第三组之外，都是选择了中心度较高，与其他节点关联密切的边，只有第三组选择的是比较“偏僻”的边。由此我们得到启发，是否可以将 Degree Centrality 和 Betweenness Centrality 结合起来，形成一种 Hybrid 方法，从而同时具有两种方法的优点。为此，我们进行实验得到的结果如下：
可以看到，实验效果非常理想，远远超过其他几组的表现，说明我们的推断是合理的。

4.7 实验总结

将这六组实验得到的相对误差分布对比展示如下：
由此我们看到，第六组的实现效果最为理想，使用的是 hybrid 方法。其次是第二组的实验效果最为理想，选择的 monitor 是 STTL 同 ORLD 之间进行通信，NY54 同 LA03

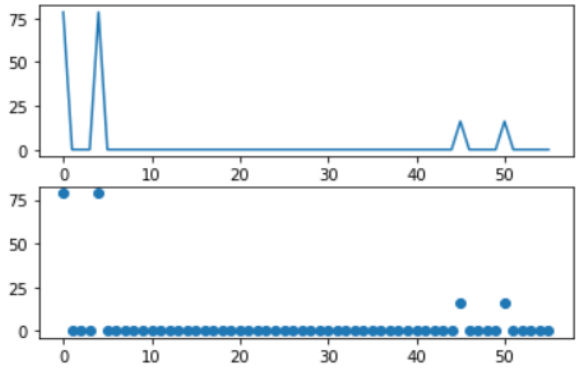

图 13: 第六组实验残差分布

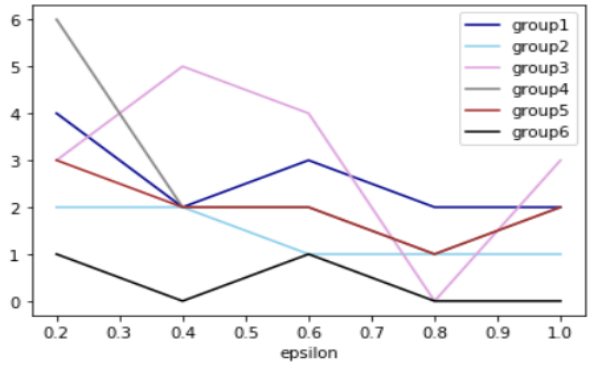

图 14: 各组相对误差分布情况

之间进行通信。由此我们可以得到一个 monitor 选取准则：结合使用 Degree Centrality 较高的点和 Betweenness Centrality 较低的点作为我们的 monitor。前者用于寻找和覆盖尽可能多的路径，后者用于对一些较为“偏僻”的路径增加他们的测量准确率。当然，上述 monitor 选择的准则仅仅是是由一个书局街 domestic network 得到的，为了更加具有可信性和普适性，还许多对多个数据集进行进一步的研究和实验才能得到。

5 结论

本次实验使用 KSP 算法，通过最小二乘法的近似方式，得到了对图中 55 条 link delay 的估计值。在进行具体的数值对比的过程之中，发现了一个独特的现象。估计结果与真实值之间的关系只有两类：第一类，估计值非常接近真实值，同真实值几乎没有差异，甚至可以做到小数点后六位精确；第二类：估计值同真实值的差异超过 0.1，呈现明显的偏差。由此可以看出图网络的自身的拓扑结构对于网络测量具有重要的影响。除此之外，跟据多次实验结果，发现并提出了综合 monitor 的选择准则，即综合平衡选择 Degree Centrality 较高的点和 Betweenness Centrality 较低的点作为我们的 monitor。在这种情况下实验效果最好。此外，我发现，RLGH、SNDG、PHNX 三个顶点及其相关联边的测量存在较大误差，建议在网络故障的情况下优先考虑检查排除这几个点。在条件允许的情况下，在日后的网络建设和完善过程之中要加强这几个点同其他点之间的联系，以增强图网络的 Robustness