

通讯网络的设计与抗毁性研究

摘要

近几年,军事信息化的建设对于一个国家军队核心战斗力的提升有着重要作用,提高军队信息化作战能力已成为必然趋势。信息战作为一种方式将逐渐步入未来战场,网络通讯的职能需求也随之越来越高。

本文针对通讯网络的设计与抗毁性进行了研究。假设因军事战备需要,在全国范围内建立一个包含 139 个城市节点的有线通讯网络,对该通讯网络进行研讨。首先对通讯网络内的城市节点建立了最优路径的数学模型。其次研究了通信线路最短连接方案、城市节点被摧毁后的备战节点的选择与连接、最优通讯网络连接图的设计方案,并通过 MATLAB 编程进行了模拟。最后根据分析结果,从连通性能、经济性能等角度,得出对于军备通讯网络设计与提升抗毁性的合理化建议。

针对问题一,结合所给城市的分布情况,在 MATLAB 中构造最小生成树模拟出通信线路的连接方案,建立城市网络连接模型,得出通信线路总长度最短的连接方案。

针对问题二,我们对通讯网络的抗毁性进行研究,确定当某些城市节点被摧毁后,建立的备战节点可以恢复整个通讯网络的正常运行。首先查询中国全部城市的地理位置,做为备战节点的候选点,把地理坐标转换为笛卡尔坐标,构建中国全部城市地理坐标的模型,以被毁城市节点为圆心及其到最远目标点为半径所构造的圆为最优备选区域,在该范围内进行筛选,得出备战节点的数量和地理坐标,以及修复路线的连接方式。

针对问题三,通过 MATLAB 得出被摧毁城市节点的“地理中心点”,以此建立模型,按照问题二的方法,详细研究了建立一个备战点或多个备战点的最优备战情况;另外为确定通讯网络的连通性能,本文引用网络的连通系数 C 作为指标,去度量其连通性能。

针对问题四,在问题一构造的模型基础上进行考虑。由鲁棒控制和连通系数 C 可知,创造回路可以提高网络的抗毁性且网络连通分支数越少越好。综合考虑网络的经济性和抗攻击性,我们在原有的基础上进行了优化,设计的方案在保证网络的高抗毁性的同时仅填加了少许网络线路,并可以使其在随机摧毁 10% 网络通讯节点的情况下连通性能仍然高于问题一的模型。

关键词: 最小生成树; 地理中心点; MATLAB; 连通性; 抗毁性研究

一、问题重述

1.1 问题的背景

当今世界，军事现代化、信息化的进程逐步加快，军事通讯网络的成熟已经成为一个国家军队实力的直接标志。现为军事演习的需要，在全国范围内构建一个有线通信网络，其选择了 139 个城市作为节点，将专用网络转接设备安置在每一个城市内，假设每两个城市之间的通信线路以地球上经过这两个城市所在节点位置大圆的劣弧长方式连接，即这两个城市之间的最短距离，以便在战争时期，实现最快最准确地传递军事信息的目的。

1.2 问题的提出

利用数学模型建立模型模拟战争时期通讯网络的建设，解决以下问题：

(1) 求出连接中国 139 个城市节点的通信线路使其总长度达到最短的网络连接方案。

(2) 某城市的通讯网络节点在战争期间遭到破坏，将会影响到整个通讯网络的连通性，假设在节点被攻击的过程中，没有办法对其进行即时修复，这将会导致整个通讯网络被分解成许多个连通分支并使其相互独立。为避免延误战机，需启用事先建设在 139 个城市以外的任意数量、地点的备战节点，并且战备节点将继承距离它最近的被破坏的城市节点的网络职能，使整个通讯网络恢复正常运行。现求出北京市、武汉市、上海市这几个城市节点受到攻击时，战备城市节点的数量、地理坐标和连接路线的修复方式。

(3) 假设包括武汉市、黄石市、岳阳市、沙市、宜昌市、信阳市、南昌市、九江市、安庆市的九个城市节点在战争期间同时被破坏，求出网络恢复所需要启用的战备城市节点的数量、地理坐标，同时指出某指标去衡量这个网络的连通性能。

(4) 在问题一搭建网络通信的前提下，对于一部分重要城市，很有必要去构建高连通性，强抗攻击性的网络通讯。在斟酌网络通讯的抗攻击性和建设网络通讯的经济性的基础上，对其提出最优的设计方案。并在该方案中，模拟出 10% 的随机通讯节点被破坏后，其通讯网络的连通性。

二、模型假设

- 1、以圆心到最远目标点为半径所构造的圆为最优备选区域，其可以兼顾所有目标点。
- 2、把地理坐标转换为笛卡尔坐标所做出的图与实际情况一致。
- 3、所生成的任一随机摧毁模型均具有足够的代表性，可以表示所有随机模型的连通性。

三、符号说明

3.1 符号说明

表 1 符号说明

符号	符号说明
R_E	地球半径
R	距离半径
P_i	圆周率
$RadLat$	弧度
Lat	东经
Lng	北纬
L	距离
L_{P_n}	战备节点到 A 市距离
L_{B_n}	战备节点到 B_n 市距离
M, N	备战节点城市待选集合
C	连通系数
H	网络连通分支
G	网络节点总数目
G_i	第 i 个连通分支中节点数目

四、问题的分析

4.1 问题一的分析

在问题一中，要求给出连接 139 个通讯城市节点的线路连接方式，并使得其线路总长度最短。由题干已知城市节点为中国国土上给定的一些节点，同时已知 139 个城市节点的地理坐标 (lng, lat) ，其中 lng 为东经， lat 代表北纬，此处可以用连通图体现这 139 个城市节点，以及 139 个城市节点之间的专用网络连接设备的通信线路，其中连通图的顶点代表城市节点，边代表两个城市节点间的通信线路，定义边的权值代表城市节点之间通信线路的长度。对于 139 个城市节点的连通图可构建很多不一样的生成树，各边权值之和最小的生成树，最小生成树^[1]即为问题一所求的最短网络连接方案。

4.2 问题二的分析

通过问题二的叙述，通讯网络中城市节点被摧毁以后，备战节点将被启用并继承其网络职能。该问题是在地球球面上以经纬度进行讨论分析，为更加直观的进行研究，我们首先将地球投影到平面上，将经纬度转换为平面直角坐标^[2]，然后将一个城市节点作为研究对象，在该城市被摧毁以后，以其为圆心，摧毁城市与其最远连接城市距离为半径，确定一个备战节点的选择范围，在该范围内对中国城市进行筛选，求出备战节点的

数量和地理坐标。

根据问题一求出的最短网络连接方案，将问题二中北京市，武汉市，上海市三个地理坐标带入讨论，求出其被摧毁时备战节点的数量和地理坐标。

4.3 问题三的分析

由问题三知，在全部 139 个城市节点中，同时有九个城市节点被摧毁，根据问题一建立的最短网络连接方案模型，可将九个摧毁城市节点分为两大部分，我们对以下情况进行讨论：

(1)在给定摧毁城市节点的部分区域内，充分考虑经济成本，只建立一个距离被摧毁城市节点最近的备战节点。可通过 MATLAB 进行“地理中心点”^[3]的运算，求得被摧毁城市节点的地理中心点。

(1.1)以地理中心点为圆心，地理中心点与最远连接城市节点距离为半径，确定一个备战节点的选择范围，在该范围内对中国城市进行筛选，筛选结果中与被摧毁城市节点距离和最短的地理坐标点即为备战节点。

(1.2)以地理中心点为圆心，地理中心点与其最远连接城市节点距离为半径，确定一个备战节点的选择范围，在该范围内对中国城市进行筛选，筛选结果中与连接的城市节点距离和最短的地理坐标点即为备战节点。

(1.3)以地理中心点为圆心，地理中心点与其最远连接城市节点距离为半径，确定一个备战节点的选择范围，在该范围内对中国城市进行筛选，筛选结果中与被摧毁城市节点和它连接的城市节点距离总和最短的地理坐标点即为备战节点。

(2)在给定摧毁城市节点的部分区域内，综合考虑经济成本和网络连通性，需建立与被摧毁城市节点数量相同的备战节点。首先假设九个被摧毁城市中只有某一个城市节点被摧毁，根据问题二的思路，可以求出备战点；之后假设九个点中的另外某点，求出备战点……最终，九个点都被摧毁一遍，选出九个备战点。具体分析：

(2.1)以被摧毁城市节点为研究对象，以其为圆心，摧毁城市与其最远连接城市距离为半径，确定一个备战节点的选择范围，在距离被摧毁城市节点最近的地理位置点建立备战节点，

(2.2)以被摧毁城市节点为研究对象，以其为圆心，摧毁城市与其最远连接城市距离为半径，确定一个备战节点的选择范围，在该范围内找与连接城市距离和最短的地理坐标点即为备战节点。

(2.3)以被摧毁城市节点为研究对象，以其为圆心，摧毁城市与其最远连接城市距离为半径，确定一个备战节点的选择范围，在该范围内找与摧毁城市和其连接城市距离和最短的地理坐标点即为备战节点。

(3)为确定通讯网络的连通性能^[4]，我们通过对其进行抗毁性^[5]的研究进而得出结论。本文我们采取计算网络的连通系数 C ^[5]，衡量出其性能的指标。若连通系数 C 越小，网络连通性越低；若连通系数 C 越大，网络连通性越高。

4.3 问题四的分析

(1)由于问题四是在问题一所构造的基础上进行考虑，所以不需要备战点。根据鲁棒控制^[6]可以知道创造回路可以提高网络的抗毁性；通过连通系数 C 的表达式可以知道网络连通分支数越少越好。因为同时需要考虑经济性，所以我们所要构造的环路需要尽量的大，且所添加的线路需要尽量少的、短。而问题三中所给的性能指标，可以演算出所选方案是否合适。

(2)根据问题中所给的条件，我们创造出被摧毁后的模型，通过分析问题一所构造的模型和被摧毁前、后的模型三者的连通系数 C ，进而求得设计后的模型连通性能。

五、问题的建立与求解

5.1 问题一

5.1.1 模型的建立

将所给的 139 个城市的经纬度数据通过以下公式 (5-1) 转变为弧度, 再计算出两城市的距离作为权值, 具体方程:

$$\begin{cases} \text{radLat1} = \text{lat1} * \pi / 180.0 \\ \text{radLat2} = \text{lat2} * \pi / 180.0 \\ a = \text{radLat1} - \text{radLat2} \\ b = (\text{lng1} - \text{lng2}) * \pi / 180.0 \\ L = 2 * \text{asin}(\sqrt{\sin^2(a/2) + \cos(\text{radLat1}) * \cos(\text{radLat2}) * \sin^2(b/2)}) * R_E \end{cases} \quad (5-1)$$

利用数据, 用 MATLAB 生成连通图的邻接矩阵, 再通过普里姆算法构造出最小生成树。

5.1.2 模型的求解

通过 MATLAB 绘制出生成的最小生成树:

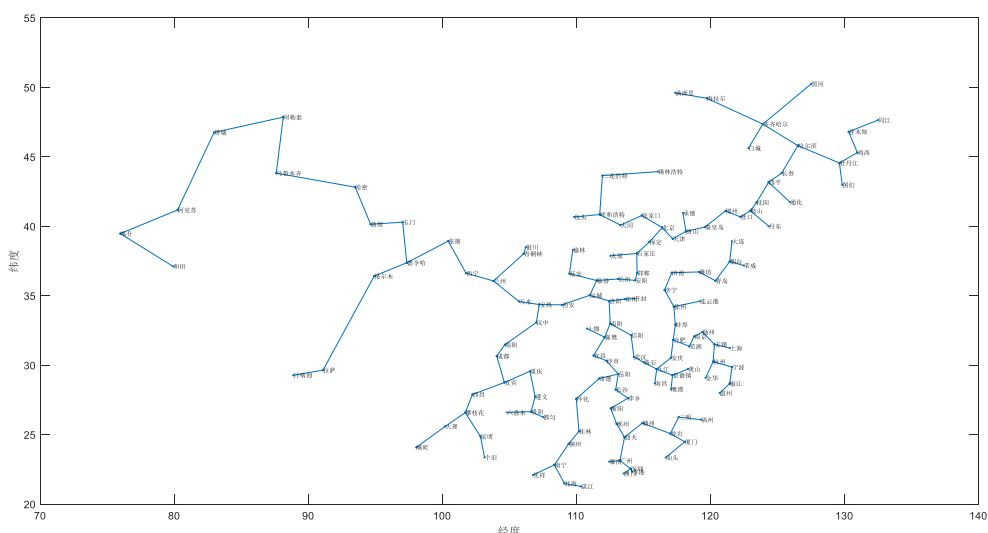


图 1 总长度最短的网络连接图

图 1 为连接 139 个城市节点通信线路总长度最短的网路连接方案。

5.2 问题二

5.2.1 模型的建立

A 市地理坐标 (lng, lat) , 与 A 市连接的城市有 B_1 市, B_2 市, ..., B_n 市 ($n < 139$), 记 A 市距其连接城市 B_1 市, B_2 市, ..., B_n 市距离为 R_1, R_2, \dots, R_n (单位/千米), 取 $R_i = \max\{R_1, R_2, \dots, R_n\}$ 为半径做圆, 得出备战节点选择的最优备选区域 S, 通过 MATLAB 在附件中国城市经纬度中进行筛选:

$$(\text{Lng} - \text{Lng}_i)^2 + (\text{Lat} - \text{Lat}_i)^2 \leq R_i^2 \quad (5-2)$$

从全国城市中得到一个满足最优备选区域 S 的城市集合 $M = \{P_1, P_2, \dots, P_m\}$ 。设集

合 M 中的城市 P_j 地理坐标为 (X_m, Y_m) 我们分为三种情况进行考虑:

$$\begin{cases} \text{radLat1} = \text{lat1} * \pi / 180.0 \\ \text{radLat2} = \text{lat2} * \pi / 180.0 \\ a = \text{radLat1} - \text{radLat2} \\ b = (\text{lng1} - \text{lng2}) * \pi / 180.0 \\ L = 2 * \text{asin}(\sqrt{\sin^2(a/2) + \cos(\text{radLat1}) * \cos(\text{radLat2}) * \sin^2(b/2)}) * R_E \end{cases} \quad (5-3)$$

(1) 与被摧毁 A 市距离 L_1 最短, 连通性较强

$$L_1 = \min\{L_{P_1}, L_{P_2}, \dots, L_{P_n}\} \quad (5-4)$$

(2) 与 A 市连接的城市 B_1 市, B_2 市, ..., B_n 市 ($n < 139$) 的距离和 L_2 最短, 经济成本低

$$L_2 = \min\{L_{B_1}, L_{B_2}, \dots, L_{B_n}\} \quad (5-5)$$

(3) 与被摧毁 A 市和城市 B_1 市, B_2 市, ..., B_n 市 ($n < 139$) 的距离和 L_3 最短, 综合考虑连通性和经济成本

$$L_3 = \min\{L_{P_1} + L_{B_1}, L_{P_2} + L_{B_2}, \dots, L_{P_n} + L_{B_n}\} \quad (5-6)$$

5.2.2 模型的求解

通过问题一可知, 在该通讯网络连通图中, 北京市与保定市、天津市、张家口市连接, 武汉市与信阳市、黄石市连接, 上海市与无锡市连接。

(1) 对于北京市, 在通讯网络与它连接的城市中, 北京市距离张家口市最远, 在其被摧毁以后, 以北京市为圆心, 北京市距离张家口市的长度为半径做圆, 在该范围内对中国城市经纬度进行筛选, 得到符合备战节点选择的集合 $M_1 = \{\text{遵化, 三河, 丰南, 廊坊, 涿州, 高碑店, 霸州}\}$, 对 M_1 中七个城市分三种情况进行考虑:

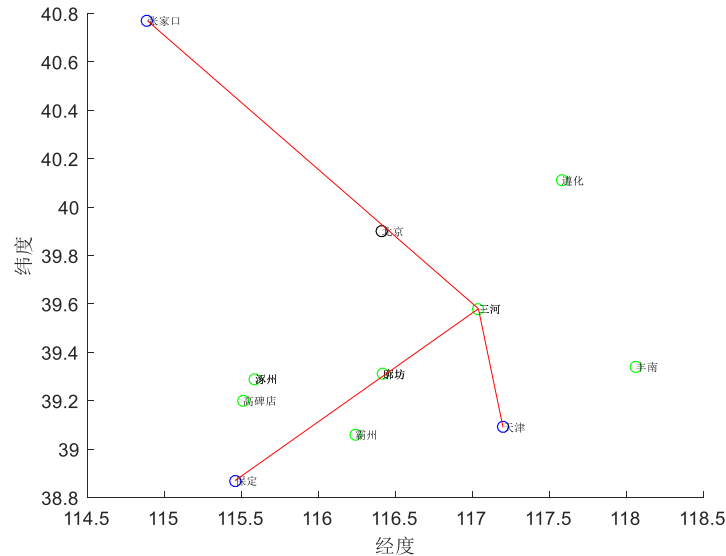


图 2 情况一 (北京)

(1.1) 在集合范围 $M_1 = \{\text{遵化, 三河, 丰南, 廊坊, 涿州, 高碑店, 霸州}\}$ 中的城市地理坐标与北京市地理坐标距离进行比较, 通过 MATLAB 计算作图得图 2, 由图 2 可直观体现出, M_1 中与北京市距离最短的城市为三河市, 故三河市可作为备战节点。

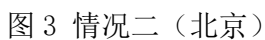


图 4 情况三（北京）

(2)武汉市,在通讯网络与它连接的城市中,武汉市距离信阳市最远,在其被摧毁以后,以武汉市为圆心,武汉市距离信阳市的长度为半径做圆,在该范围内对中国城市经纬度进行筛选,得到符合备战节点选择的集合 $M_2 = \{\text{随州, 广水, 安陆, 应城, 孝感, 仙桃, 洪湖, 咸宁, 潜江, 大冶, 武穴, 瑞昌, 麻城, 鄂州, 黄州, 浦圻}\}$, 对 M_2 中十

六个城市分三种情况进行考虑：

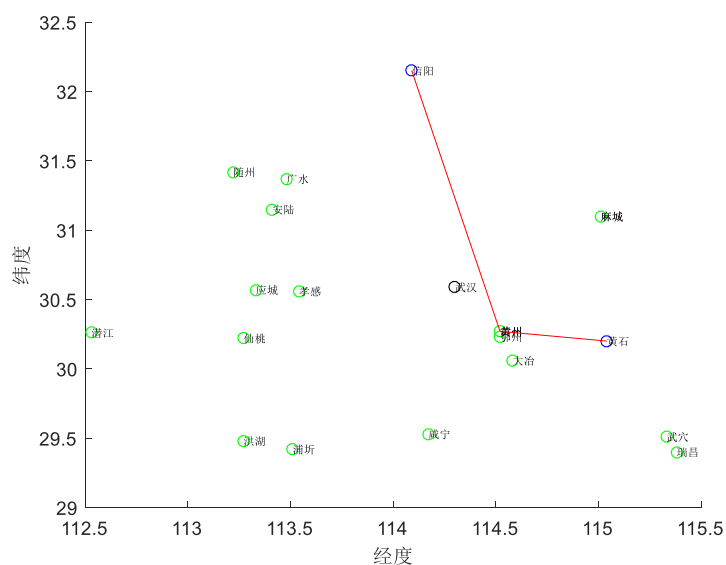


图 5 情况一（武汉）

(2.1)在集合范围 $M_2 = \{\text{随州, 广水, 安陆, 应城, 孝感, 仙桃, 洪湖, 咸宁, 潜江, 大冶, 武穴, 瑞昌, 麻城, 鄂州, 黄州, 蒲圻}\}$ 中的城市地理坐标与武汉市地理坐标距离进行比较, 通过 MTLAB 计算作图得图 5, 由图 5 可直观体现出, M_2 中与武汉市距离最短的城市为黄州市, 故黄州市可作为备战节点。

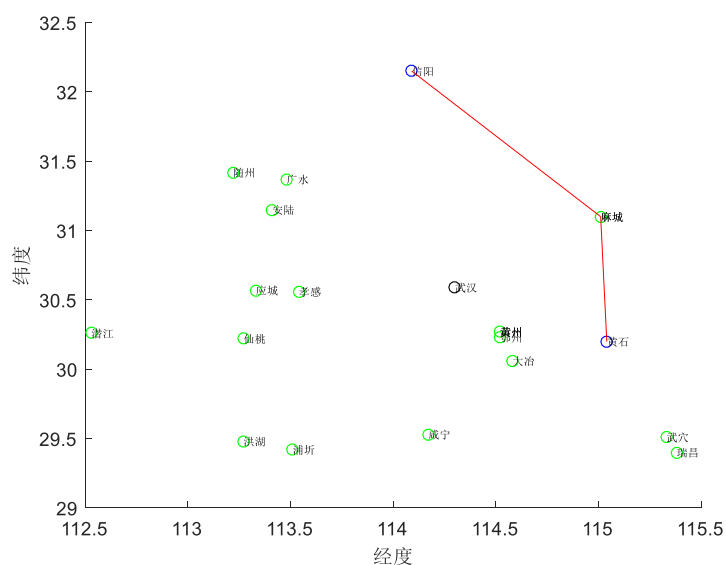


图 6 情况二（武汉）

(2.2)在集合范围 M_2 中的城市地理坐标与信阳市、黄石市两市地理坐标距离和进行比较, 通过 MTLAB 计算作图得图 6, 由图 6 可直观体现出, M_2 中与信阳市、黄石市两市距离和最短的城市为麻城市, 故麻城市可作为备战节点。

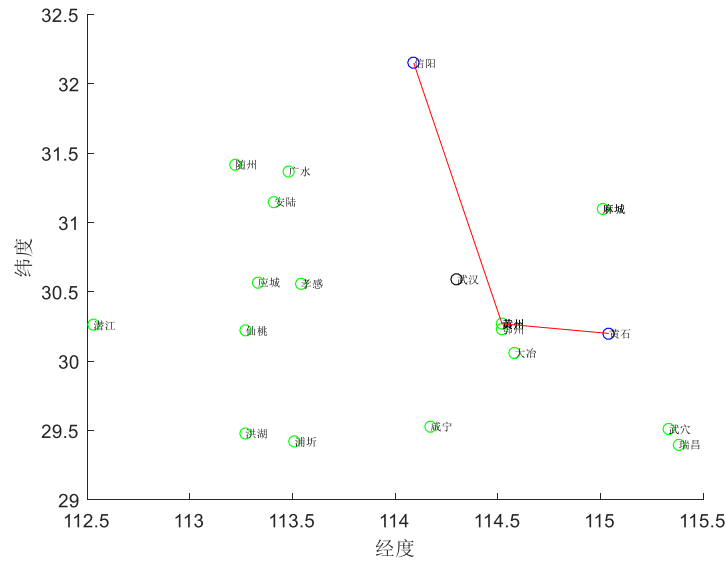


图 7 情况三（武汉）

(2.3) 在集合范围 M_2 中的城市地理坐标与武汉市、信阳市、黄石市三市地理坐标距离和进行比较，通过 MTLAB 计算作图得图 7，由图 7 可直观体现出， M_2 中与武汉市、信阳市、黄石市三市距离和最短的城市为黄州市，故黄州市可作为备战节点。

(3) 对于上海市，只有无锡市与其连接，在其被摧毁以后，以上海市为圆心，上海市距离无锡市的长度为半径做圆，在该范围内对中国城市经纬度进行筛选，得到符合备战节点选择的集合 $M_3 = \{\text{通州, 海门, 太仓, 启乐, 常熟, 张家港, 苏州, 吴江, 昆山, 平湖}\}$ ，对 M_3 中十个城市分三种情况进行考虑：

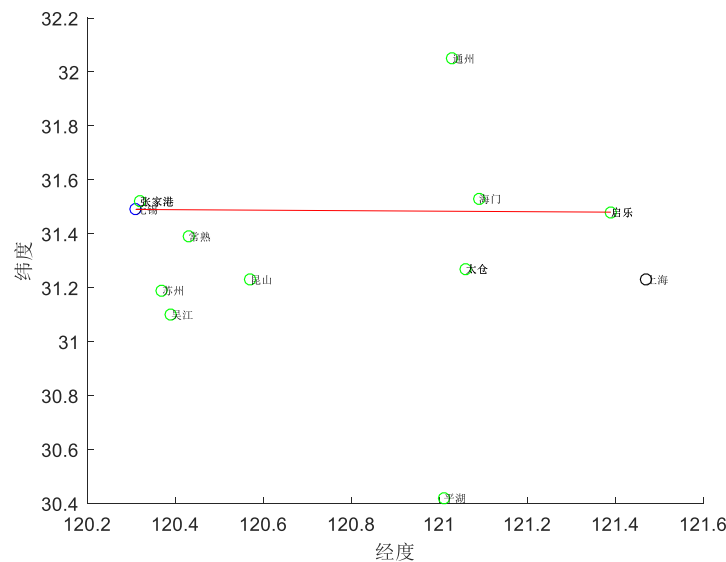


图 8 情况一（上海）

(3.1) 在集合范围 $M_3 = \{\text{通州, 海门, 太仓, 启乐, 常熟, 张家港, 苏州, 吴江, 昆山, 平湖}\}$ 中的城市地理坐标与上海市地理坐标距离进行比较，通过 MTLAB 计算作图得图 8，由图 8 可直观体现出， M_3 中与上海市距离最短的城市为启乐市，故启乐市可作为备战节点。

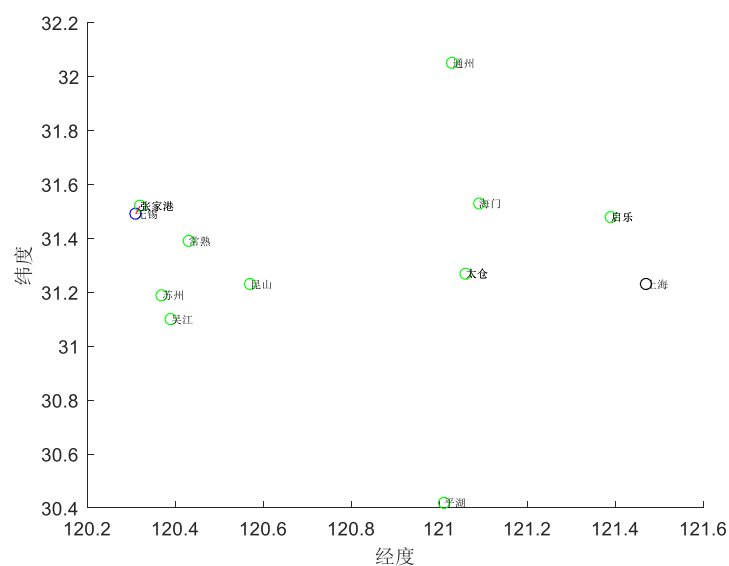


图 9 情况二（上海）

(3.2)在集合范围 M_3 中的城市地理坐标与与无锡市地理坐标距离和进行比较，通过 **MTLAB** 计算作图得图 9，由图 9 可直观体现， M_3 中与无锡市距离最短的城市为张家港市，故张家港市可作为备战节点。

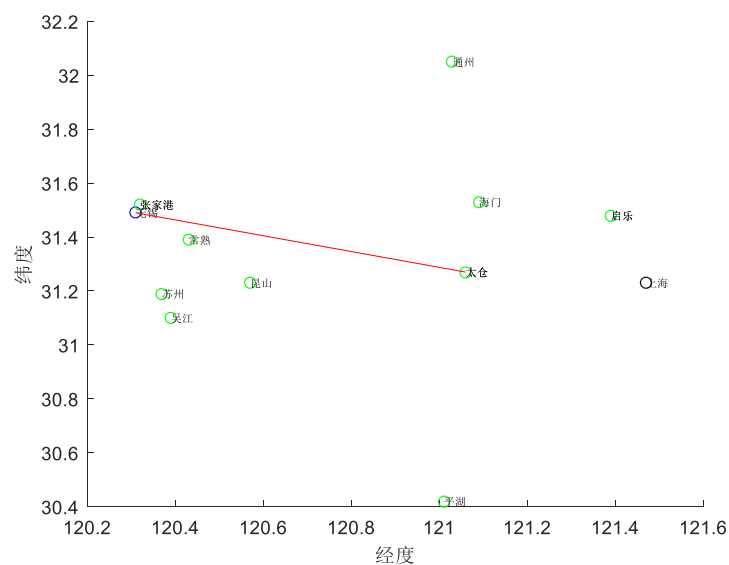


图 10 情况三（上海）

(3.3)在集合范围 M_3 中的城市地理坐标与上海市、无锡市两市地理坐标距离和进行比较，通过 **MTLAB** 计算作图得图十，由图 10 可直观体现， M_3 中与上海市、无锡市两市距离和最短的城市为太仓市，故太仓市可作为备战节点。

5.3 问题三

5.3.1 模型的建立

在全部 139 个城市节点中，同时有九个城市节点被摧毁，根据问题一建立的最短网络连接方案模型，截取出九个被摧毁的城市节点的连接图：

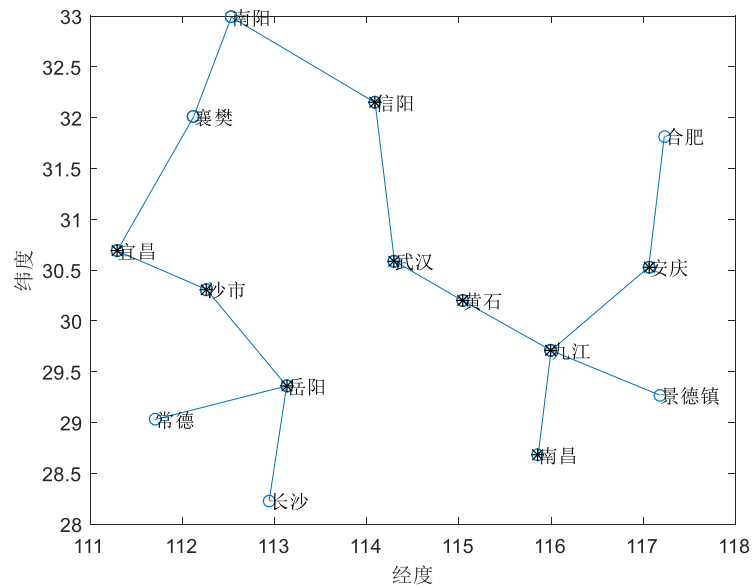


图 11 被摧毁城市节点的部分连接图

如图 11，武汉市、黄石市、岳阳市、沙市、宜昌市、信阳市、南昌市、九江市、安庆市这九个城市被摧毁。

$$\begin{cases}
 \text{RadLat} = \text{Lat} * \pi / 180.0 \\
 \text{RadLng} = \text{Lng} * \pi / 180.0 \\
 X = \cos(\text{RadLat}) * \cos(\text{RadLng}) \\
 Y = \cos(\text{RadLat}) * \sin(\text{RadLng}) \\
 Z = \sin(\text{RadLat}) \\
 x = \frac{1}{n} \sum_{i=1}^n X_i; y = \frac{1}{n} \sum_{i=1}^n Y_i; z = \frac{1}{n} \sum_{i=1}^n Z_i \\
 \text{Lng1} = \text{atan2}(y, x) * 180 / \pi \\
 \text{Hyp} = \sqrt{(x * x + y * y)} \\
 \text{Lat1} = \text{atan2}(z, \text{Hyp}) * 180 / \pi
 \end{cases} \quad (5-7)$$

(1)在给定摧毁城市节点的部分区域内，充分考虑经济成本，只建立一个备战节点。可通过 MTLAB 进行运算公式 (5-7) 求得被摧毁城市节点的“地理中心点”，以其为圆心，地理中心点与最远连接城市节点距离为半径，确定一个备战节点的选择范围 N ，在 N 中对中国城市进行筛选。

筛选结果可分为三类备战节点：与被摧毁城市节点距离和最短的地理坐标点为备战节点 1；与连接的城市节点距离和最短的地理坐标点为备战节点 2；与被摧毁城市节点和它连接城市节点距离总和最短的地理坐标点为备战节点 3。

(2)在给定摧毁城市节点的部分区域内，综合考虑经济成本和网络连通性，建立与被摧毁城市节点数量相同的备战节点。根据问题二的思路，首先假设只有一个城市节点被摧毁，以其为圆心，摧毁城市与其最远连接城市距离为半径，确定一个备战节点的选择

范围 N' ，在 N' 中对中国城市进行筛选。

这样会出现三类备战节点：与被摧毁城市节点距离和最短的地理坐标点为备战节点 4；与连接的城市节点距离和最短的地理坐标点为备战节点 5；与被摧毁城市节点和它连接城市节点距离总和最短的地理坐标点为备战节点 6。

5.3.2 模型的求解

(1) 建立一个备战节点时，首先找出被摧毁九个城市的地理中心点，以其点为圆心，地理中心点与最远连接城市节点距离为半径，确定一个备战节点的选择范围 N_1 ，在该范围内对全国城市进行筛选。

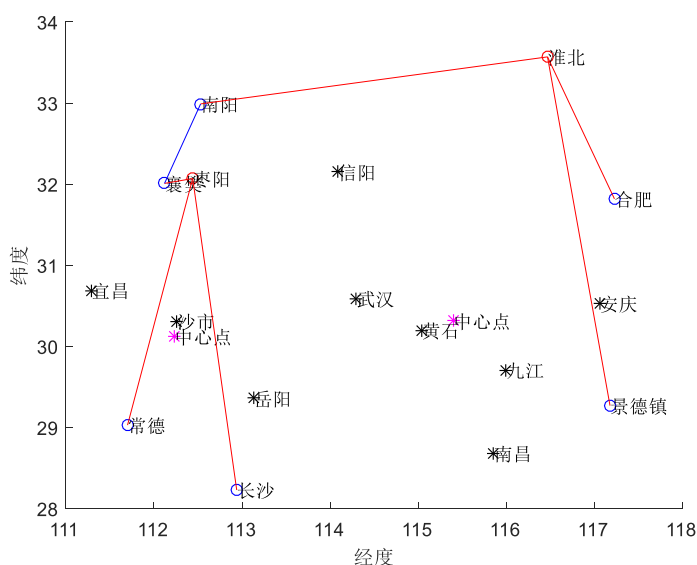


图 12 战备节点 1

(1.1) 在集合范围 N_1 中的城市节点与被摧毁的九座城市地理坐标距离总和进行比较，通过 MATLAB 计算作图得图 12，由图 12 可知，红色标注城市枣阳市和淮北市到被摧毁的九座城市距离总和最短，即为战备节点 1。

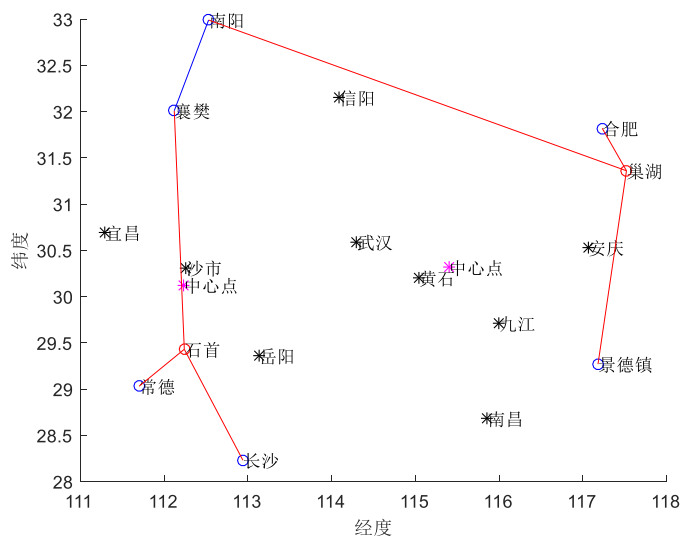


图 13 战备节点 2

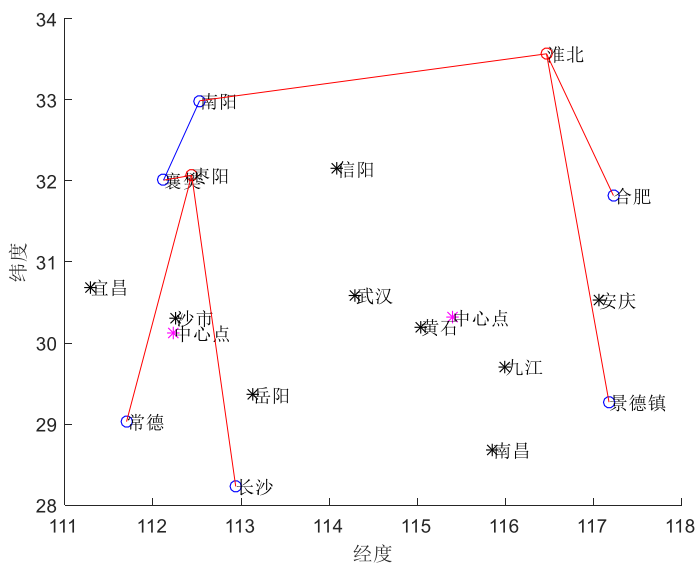


图 14 战备节点 3

(1.3) 在集合范围 N_1 中的城市节点与被摧毁城市节点和它连接城市节点距离总和进行比较, 通过 **MTLAB** 计算作图得图 14, 由图 14 可知, 红色标注城市枣阳市和淮北市与被摧毁城市节点和它连接城市节点距离总和最短, 即为战备节点 3。可以发现备战节点 3 与备战节点 1 为同一城市节点。

(2)建立与被摧毁城市节点数量相同的备战节点时,我们首先假设只有一个城市节点被摧毁,以被摧毁城市节点为研究对象,以其为圆心,摧毁城市与其最远连接城市距离为半径,确定一个备战节点的选择范围 N_0 , 在该范围内对全国城市进行筛选。

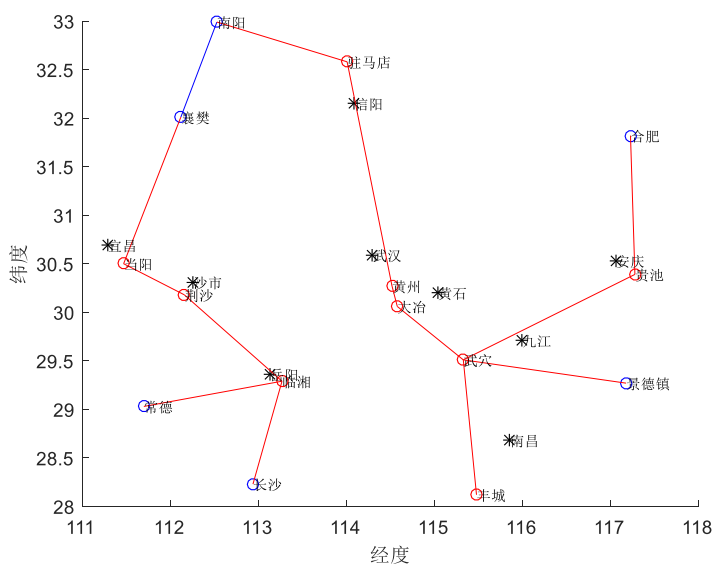


图 15 战备节点 4

(2.1) 在集合范围 N_2 中的城市节点与被摧毁的九座城市地理坐标距离总和进行比较, 通过 MTLAB 计算作图得图 15, 由图 15 可知, 红色标注城市驻马店市、黄州市、武穴市等九个城市与被摧毁城市节点距离总和最短, 即为战备节点 4。

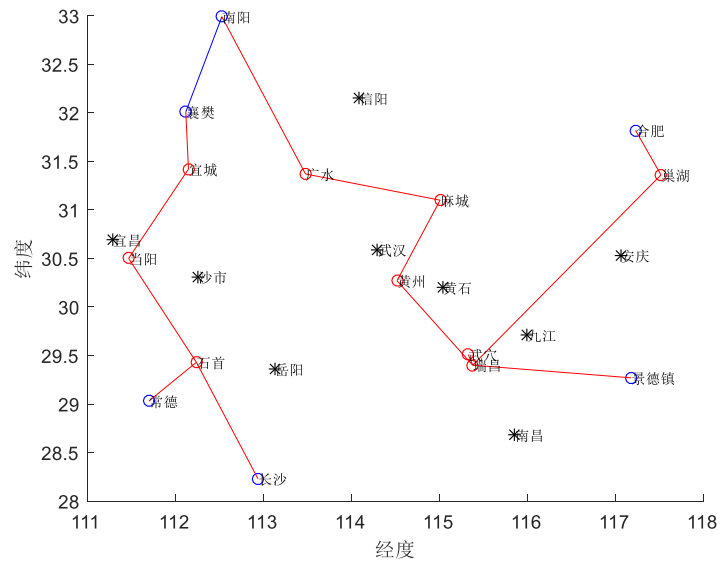


图 16 战备节点 5

(2.2) 在集合范围 N_2 中的城市节点与其连接的城市节点地理坐标距离总和进行比较, 通过 MTLAB 计算作图得图 16, 由图 16 可知, 红色标注城市广水市、麻城市、武穴市等九个城市与其连接的城市节点距离总和最短的地理坐标点, 即为备战节点 5。

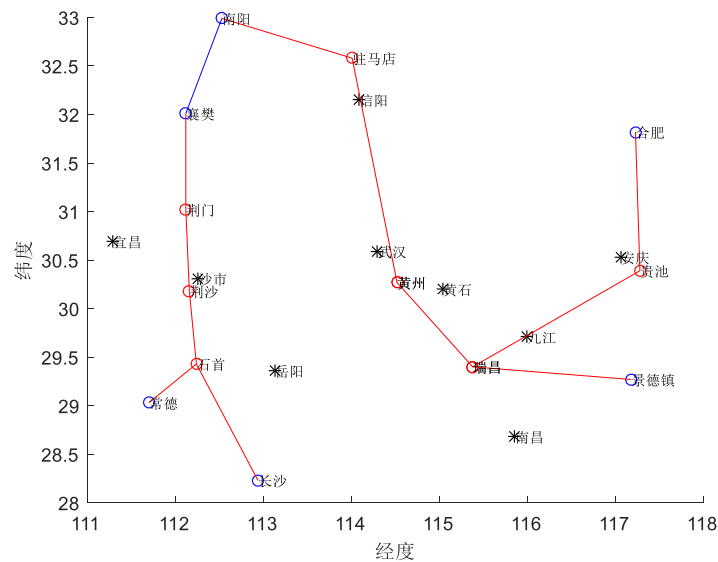


图 17 战备节点 6

(2.3) 在集合范围 N_2 中的城市节点与被摧毁城市节点和它连接城市节点地理坐标距离总和进行比较, 通过 MTLAB 计算作图得图 17, 由图 17 可知, 红色标注城市驻马店市、石首市、荆门市等七个城市与其连接的城市节点距离总和最短的地理坐标点, 即为备战节点 6。其中武汉市和黄石市的战备节点重合, 南昌市和九江市的战备节点重合。

(3) 通讯网络中, 网络连通性能对于整个网络具有信息传递的及时性、效率性等性

能具有重要影响，我们对其指标根据网络的连通系数进行研究：

$$C = \frac{1}{H \sum_{i=1}^H *(G_i / G) * l_i} \quad (5-8)$$

我们可以通过网络连通系数 C 衡量所设计的网络连通性能的优劣。当在某网络中的连通系数 C 越小，所设计的网络其连通性越低；连通系数越大，所设计的网络其连通性越高。

5.4 问题四

5.4.1 模型的建立

(1)根据鲁棒控制、网络的连通系数 C 、抗毁性可知，我们所需要设计的网络需要建立尽量少的回路。对问题一所得模型添加边来建立回路，考虑到经济性同时，我们所添加的边需要尽量少的、短。

通过计算我们可以得到连接和田市，日喀则市，拉萨市，畹町市，个旧市，凭祥市，湛江市，澳门市，福州市，温州市，大连市，营口市，锡林浩特市，满洲里这十四个城市时，我们所需要建设的线路相对较短，所生成的网络连通分支数相对较少。

(2)在上述的基础上，通过对设计好的模型进行随机选取，摧毁 10%的城市通讯节点，计算其网络的连通系数 C_3 ，将其与问题一基础上的模型的网络连通系数 C_1 和问题四中设计的模型的网络连通系数 C_2 进行比较，从而验证设计的模型其网络性能高低和判断设计的模型摧毁后网络性能。

5.4.2 模型的求解

(1)通过 MATLAB 绘制出所设计的网络连通图：

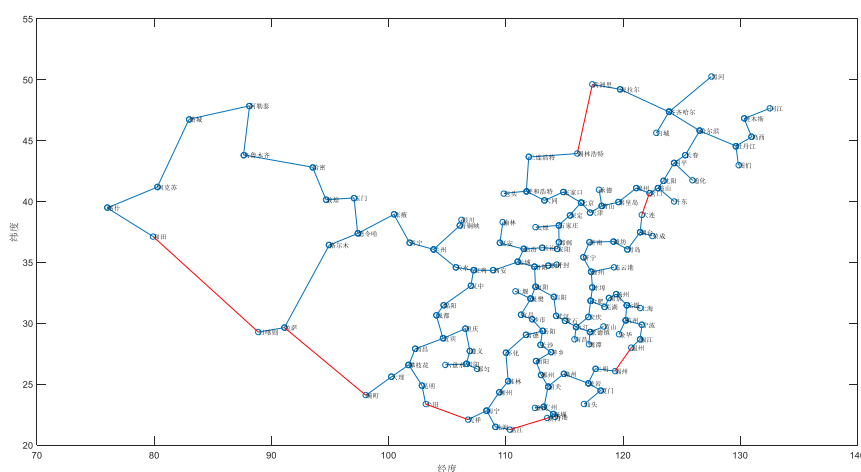


图 18 最优设计方案

图 18 为综合考虑后计算所得最优设计方案。

(2)模拟摧毁一部分后网络的连通性能

在该设计方案中，模拟 10%的随机通讯节点被摧毁后，通讯网络的连通性。

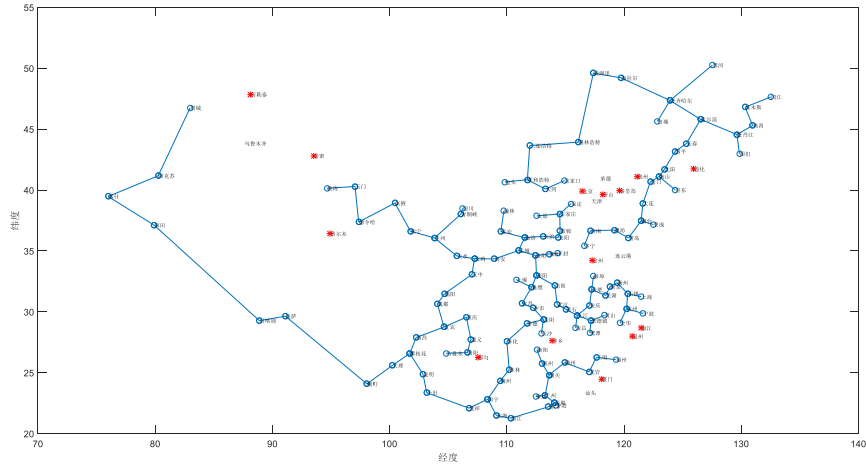


图 19 随机摧毁 10%后网路连接方案

如图 19 所示，随机摧毁 10%的通讯节点以后，通过计算问题一、本题设计的优化模型、随机摧毁后的优化模型的网络连通系数，对其进行比较，判断网络的连通性能。

网络的连通系数：

$$C = \frac{1}{H \sum_{i=1}^H *(G_i / G) * l_i} \quad (5-9)$$

通过计算可得

$$C_1 = 0.000128409660418922$$

$$C_2 = 0.000130369242452242$$

$$C_3 = 0.000132501223017047$$

通过比较网络连通系数，易得 $C_2 > C_1$ ， $C_3 > C_1$ ，而网络连通系数越大连通性能越强所以所设计方案成立。

六、模型的评价

6.1 模型的优点

1、通过 MTALAB 建模可以全面考虑各种影响因素，并且所建立模型直观性更强，便于观察与描述。

2、在构造最小生成树的 prim 算法中，时间复杂度不仅仅依赖于排序算法，而且主要与点的个数有关，更加适用于密集图。

3、鲁棒控制尤其适用于那些不确定因素变化范围大，稳定裕度小的系统，符合我们随机摧毁模型的建立。

6.2 模型的缺点

1、我们所构建的模型其最小生成树的节点传输能力有限，距离较远的某些节点必须通过多跳的通信方式，才可以进行军事信息的交换，所以必须保证我们所设计的网络在物理链路上的连通性要足够好。

2、对于网络连通系数 C 的计算通过定义法计算十分困难，并且不利于应用计算机程序来解决

参考文献

- [1] 叶军伟. 普里姆算法和克鲁斯卡尔算法构造最小生成树[J]. 河南科技, 2010(6):22-22.
- [2] 韩忠民. 知经纬度计算两点精确距离[J]. 科技传播, 2011(11):211+233.
- [3] Calculation Methods [2019.5.3],
http://www.geomidpoint.com/calculation.html?tdsourcetag=s_pcqq_aiomsg
- [4] 李玉军. (0). 无线网络连通性及路由关键技术研究. (Doctoral dissertation, 电子科技大学).
- [5] 吴俊, 谭跃进. 复杂网络抗毁性测度研究[J]. 系统工程学报, 2005, 20(2):128-131.
- [6] 杨盐生. (2000). 不确定系统的鲁棒控制及其在船舶运动控制中的应用. (Doctoral dissertation, 大连海事大学).

附录

附录 1

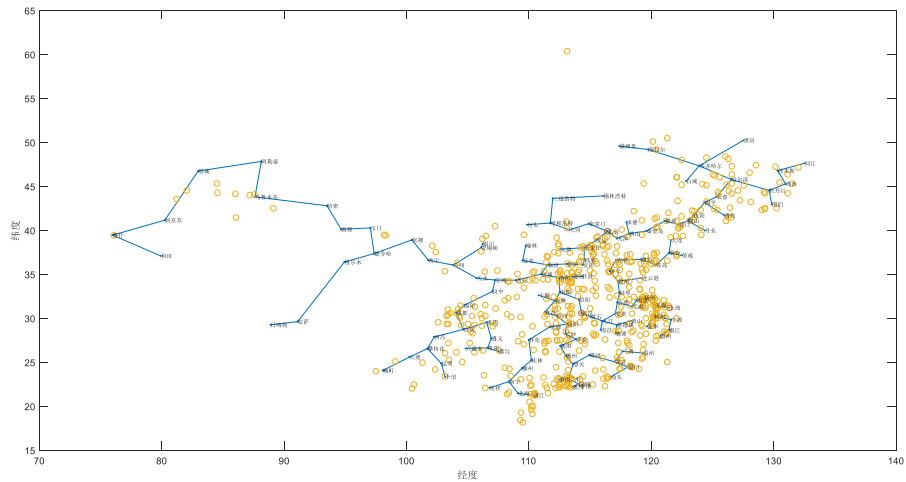


图 1 插入中国所有城市在地图中的地理位置显示

附录 2

matlab 编写的求解问题一、二、三、四的源代码

```
%Problem_1.m
clear
clc
load('location.mat');
d=Distance{lng,lat);
P = zeros(1, 139);
P(1,1) = 1;
V = 1:139;
V_P = V - P;
link = zeros(138,2);
k=1;
Sum1=0;
c=zeros(139);
while k<139
    p = P(P~=0);
    v = V_P(V_P~=0);
    pv = min(min(d(p,v)));
    [x, y] = find(d==pv);
    for i=1:length(x)
        if any(P==x(i)) && any(V_P==y(i))
            P(1,y(i)) = y(i);
            V_P = V - P;
            link(k, :) = [x(i), y(i)];
            Sum1=Sum1+d(x(i), y(i));
            c(x(i), y(i))=d(x(i), y(i));
            k = k+1;
            break;
        end
    end
end
```

```

        end
    end
    r=[lng,lat];
    figure;
    gplot(c,r,'- o')
    text(lng,lat,cityName,'fontsize',7,'fontweight','bold');%6
    xlabel('经度');
    ylabel('纬度');

%Problem_2.m
clear
clc
load('Problem_1.mat');
gplot(c,r,'- .')
hold;
scatter([lng(105),lng(53),lng(56)],[lat(105),lat(53),lat(56)]);
scatter(insLng,insLat);
hold off;
text(lng,lat,cityName,'fontsize',6,'fontweight','bold');
xlabel('经度');
ylabel('纬度');

%Problem_2_Beijing.m
clear
clc
load('Problem_2.mat');
firedIndex=105;%北京节点索引
connIndex=[find(c(:,firedIndex)),find(c(firedIndex,:))]; %连接点的索引
%连接点的经纬度
coLng=lng(connIndex);
coLat=lat(connIndex);
R=max(max(c(firedIndex,find(c(firedIndex,:))))),max(c(find(c(:,firedIndex)),firedIndex)))
;%求得与被炸点之前连接节点的最远距离
[insIndex,insP1,insP2,insP3]=Alterna(lng(firedIndex),lat(firedIndex),insLng,insLat,R,coLng,coLat);
%备选节点坐标
insX=insLng(insIndex);
insY=insLat(insIndex);

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCityName(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY

```

```

;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP1(1);coLng];
lIndexY1=[insP1(2);coLat];
for i=2:length(lIndexX1)
    pointX1=[lIndexX1(i),lIndexX1(1)];
    pointY1=[lIndexY1(i),lIndexY1(1)];
    line(pointX1,pointY1,'color','r');
end
hold off;
xlabel('经度');
ylabel('纬度');

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCity
Name(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY
;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP2(1);coLng];
lIndexY1=[insP2(2);coLat];
for i=2:length(lIndexX1)
    pointX1=[lIndexX1(i),lIndexX1(1)];
    pointY1=[lIndexY1(i),lIndexY1(1)];
    line(pointX1,pointY1,'color','r');
end
hold off;
xlabel('经度');
ylabel('纬度');

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCity
Name(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY
;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP3(1);coLng];
lIndexY1=[insP3(2);coLat];
for i=2:length(lIndexX1)

```

```

        pointX1=[lIndexX1(i),lIndexX1(1)];
        pointY1=[lIndexY1(i),lIndexY1(1)];
        line(pointX1,pointY1,'color','r');
    end
    hold off;
    xlabel('经度');
    ylabel('纬度');

%Problem_2_Wuhan.m
clear
clc
load('Problem_2.mat');
firedIndex=53;%武汉节点索引
connIndex=[find(c(:,firedIndex)),find(c(firedIndex,:))]; %连接点的索引
%连接点的经纬度
coLng=lng(connIndex);
coLat=lat(connIndex);
R=max(max(c(firedIndex,find(c(firedIndex,:)))),max(c(find(c(:,firedIndex)),firedIndex)))
;%求得与被炸点之前连接节点的最远距离
[insIndex,insP1,insP2,insP3]=Alterna(lng(firedIndex),lat(firedIndex),insLng,insLat,R,c
oLng,coLat);
%备选节点坐标
insX=insLng(insIndex);
insY=insLat(insIndex);

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCity
Name(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY
;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP1(1);coLng];
lIndexY1=[insP1(2);coLat];
for i=2:length(lIndexX1)
    pointX1=[lIndexX1(i),lIndexX1(1)];
    pointY1=[lIndexY1(i),lIndexY1(1)];
    line(pointX1,pointY1,'color','r');
end
hold off;
xlabel('经度');
ylabel('纬度');

figure;

```

```

scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCity
Name(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY
;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP2(1);coLng];
lIndexY1=[insP2(2);coLat];
for i=2:length(lIndexX1)
    pointX1=[lIndexX1(i),lIndexX1(1)];
    pointY1=[lIndexY1(i),lIndexY1(1)];
    line(pointX1,pointY1,'color','r');
end
hold off;
xlabel('经度');
ylabel('纬度');

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCity
Name(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY
;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP3(1);coLng];
lIndexY1=[insP3(2);coLat];
for i=2:length(lIndexX1)
    pointX1=[lIndexX1(i),lIndexX1(1)];
    pointY1=[lIndexY1(i),lIndexY1(1)];
    line(pointX1,pointY1,'color','r');
end
hold off;
xlabel('经度');
ylabel('纬度');

%Problem_2_Shanghai.m
clear
clc
load('Problem_2.mat');
firedIndex=56;%上海节点索引

```

```

connIndex=[find(c(:,firedIndex)),find(c(firedIndex,:))]; %连接点的索引
%连接点的经纬度
coLng=lng(connIndex);
coLat=lat(connIndex);
R=c(find(c(:,firedIndex)),firedIndex) %求得与被炸点之前连接节点的最远距离
[insIndex,insP1,insP2,insP3]=Alterna(lng(firedIndex),lat(firedIndex),insLng,insLat,R,coLng,coLat);
%备选节点坐标
insX=insLng(insIndex);
insY=insLat(insIndex);

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCityName(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP1(1);coLng];
lIndexY1=[insP1(2);coLat];
for i=2:length(lIndexX1)
    pointX1=[lIndexX1(i),lIndexX1(1)];
    pointY1=[lIndexY1(i),lIndexY1(1)];
    line(pointX1,pointY1,'color','r');
end
hold off;
xlabel('经度');
ylabel('纬度');

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCityName(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP2(1);coLng];
lIndexY1=[insP2(2);coLat];
for i=2:length(lIndexX1)

```

```

        pointX1=[lIndexX1(i),lIndexX1(1)];
        pointY1=[lIndexY1(i),lIndexY1(1)];
        line(pointX1,pointY1,'color','r');
    end
    hold off;
    xlabel('经度');
    ylabel('纬度');

figure;
scatter(lng(firedIndex),lat(firedIndex),'k');%被炸点
hold;
scatter(coLng,coLat,'b');%连接点
scatter(insX,insY,'g');%备选点
insName=[cityName(firedIndex);cityName(connIndex);insCityName(insIndex);insCity
Name(insP1(3));insCityName(insP2(3));insCityName(insP3(3))];

%标注
text([lng(firedIndex);coLng;insX;insP1(1);insP2(1);insP3(1)],[lat(firedIndex);coLat;insY
;insP1(2);insP2(2);insP3(2)],insName,'fontsize',6,'fontweight','bold');
lIndexX1=[insP3(1);coLng];
lIndexY1=[insP3(2);coLat];
for i=2:length(lIndexX1)
    pointX1=[lIndexX1(i),lIndexX1(1)];
    pointY1=[lIndexY1(i),lIndexY1(1)];
    line(pointX1,pointY1,'color','r');
end
hold off;
xlabel('经度');
ylabel('纬度');

%Problem_3.m
clear
clc
load('Problem_2.mat');
%邻接矩阵
pThrMat=zeros(15);
pThrMat(1,2)=1;
pThrMat(1,6)=1;
pThrMat(2,8)=1;
pThrMat(3,12)=1;
pThrMat(3,13)=1;
pThrMat(3,4)=1;
pThrMat(4,5)=1;
pThrMat(5,14)=1;
pThrMat(6,15)=1;
pThrMat(7,8)=1;
pThrMat(8,9)=1;
pThrMat(8,11)=1;

```



```

pThrMat(9,10)=1;
pThrMat(14,15)=1;
%武汉,黄石,岳阳,沙市,宜昌,信阳,南昌,九江,安庆 ||| 合肥 景德镇 长沙 常德 襄樊 南阳
%53,49,43,51,55,63,37,46,52,60,42,34,39,61,67 城市索引
pThrMatIndex=[53,49,43,51,55,63,37,46,52,60,42,34,39,61,67];
pThrMatCo=[lng(pThrMatIndex),lat(pThrMatIndex)];
gplot(pThrMat,pThrMatCo,'- o');
text(pThrMatCo(:,1),pThrMatCo(:,2),cityName(pThrMatIndex),'fontsize',10,'fontweight'
,'bold');
%被炸点黑色
hold
scatter(pThrMatCo(1:9,1),pThrMatCo(1:9,2),'k','*');
hold off;
xlabel('经度');
ylabel('纬度');

%Problem_3_1.m
clear
clc
load('Problem_3.mat');

cityListIndex1=[53,49,63,37,46,52];%被炸城市节点索引集合 1
cityListIndex2=[43,51,55];%被炸城市节点索引集合 2
centerP1=Center(cityListIndex1,lat,lng);%被炸城市节点集合的中心点 1
centerP2=Center(cityListIndex2,lat,lng);%被炸城市节点集合的中心点 2
%被炸点集合的坐标
fList1=[lng(cityListIndex1),lat(cityListIndex1)];
fList2=[lng(cityListIndex2),lat(cityListIndex2)];
%part1 的连接点
coLng1=lng([67,60,42]);
coLat1=lat([67,60,42]);
%part2 的连接点
coLng2=lng([34,39,61]);
coLat2=lat([34,39,61]);

R1=max([GetDistance(lng(67),lat(67),centerP1(1),centerP1(2)),GetDistance(lng(60),l
at(60),centerP1(1),centerP1(2)),GetDistance(lng(42),lat(42),centerP1(1),centerP1(2))
]);
R2=max([GetDistance(lng(34),lat(34),centerP2(1),centerP2(2)),GetDistance(lng(39),l
at(39),centerP2(1),centerP2(2)),GetDistance(lng(61),lat(61),centerP2(1),centerP2(2))
]);
[insIndex1,insP11,insP21,insP31]=AlternaP3(fList1,centerP1(1),centerP1(2),insLng,ins
Lat,R1,coLng1,coLat1);
[insIndex2,insP12,insP22,insP32]=AlternaP3(fList2,centerP2(1),centerP2(2),insLng,i
nsLat,R2,coLng2,coLat2);

figure;

```

```

scatter(pThrMatCo(1:9,1),pThrMatCo(1:9,2),'k','*');%被炸点
hold;
scatter([insP11(1),insP12(1)],[insP11(2),insP12(2)],'r');%备战点
scatter([coLng1;coLng2],[coLat1;coLat2],'b');%连接点
scatter([centerP1(1),centerP2(1)],[centerP1(2),centerP2(2)],'m','*')%中心点
text([centerP1(1),centerP2(1)],[centerP1(2),centerP2(2)],'中心点');
text([lng(pThrMatIndex);insP11(1);insP12(1)],[lat(pThrMatIndex);insP11(2);insP12(2)],
[cityName(pThrMatIndex);insCityName(insP11(3));insCityName(insP12(3))],'fontsize',
10,'fontweight','bold');
for i=1:length(coLng1)
    lx1=[coLng1(i);insP11(1)];
    ly1=[coLat1(i);insP11(2)];
    lx2=[coLng2(i);insP12(1)];
    ly2=[coLat2(i);insP12(2)];
    line(lx1,ly1,'color','r');
    line(lx2,ly2,'color','r');
end
line([lng(61),lng(67)],[lat(61),lat(67)],'color','b');
hold off;
xlabel('经度');
ylabel('纬度');

%
figure;
scatter(pThrMatCo(1:9,1),pThrMatCo(1:9,2),'k','*');%被炸点
hold;
scatter([insP21(1),insP22(1)],[insP21(2),insP22(2)],'r');%备战点
scatter([coLng1;coLng2],[coLat1;coLat2],'b');%连接点
scatter([centerP1(1),centerP2(1)],[centerP1(2),centerP2(2)],'m','*')%中心点
text([centerP1(1),centerP2(1)],[centerP1(2),centerP2(2)],'中心点');
text([lng(pThrMatIndex);insP21(1);insP22(1)],[lat(pThrMatIndex);insP21(2);insP22(2)],
[cityName(pThrMatIndex);insCityName(insP21(3));insCityName(insP22(3))],'fontsize',
10,'fontweight','bold');
for i=1:length(coLng1)
    lx1=[coLng1(i);insP21(1)];
    ly1=[coLat1(i);insP21(2)];
    lx2=[coLng2(i);insP22(1)];
    ly2=[coLat2(i);insP22(2)];
    line(lx1,ly1,'color','r');
    line(lx2,ly2,'color','r');
end
line([lng(61),lng(67)],[lat(61),lat(67)],'color','b');
hold off;
xlabel('经度');
ylabel('纬度');

%
figure;

```

```

scatter(pThrMatCo(1:9,1),pThrMatCo(1:9,2),'k','*');%被炸点
hold;
scatter([insP31(1),insP32(1)],[insP31(2),insP32(2)],'r');%备战点
scatter([coLng1,coLng2],[coLat1,coLat2],'b');%连接点
scatter([centerP1(1),centerP2(1)],[centerP1(2),centerP2(2)],'m','*')%中心点
text([centerP1(1),centerP2(1)],[centerP1(2),centerP2(2)],'中心点');
text([lng(pThrMatIndex);insP31(1);insP32(1)],[lat(pThrMatIndex);insP31(2);insP32(2)]
,[cityName(pThrMatIndex);insCityName(insP31(3));insCityName(insP32(3))],'fontsize'
,10,'fontweight','bold');
for i=1:length(coLng1)
    lx1=[coLng1(i);insP31(1)];
    ly1=[coLat1(i);insP31(2)];
    lx2=[coLng2(i);insP32(1)];
    ly2=[coLat2(i);insP32(2)];
    line(lx1,ly1,'color','r');
    line(lx2,ly2,'color','r');
end
line([lng(61),lng(67)],[lat(61),lat(67)],'color','b');
hold off;
xlabel('经度');
ylabel('纬度');

```

%Problem_3_2.m

clear

clc

load('Problem_3_1.mat');

%返回参数列表%返回参数 维度难处理成 object,无法用循环解决

%被炸节点集合索引，按分布顺序

reserveListIndex=[cityListIndex1,cityListIndex2];

```

RR=max(max(c(reserveListIndex(1),find(c(reserveListIndex(1),:)))),max(c(find(c(:,res
erveListIndex(1))),reserveListIndex(1))));

```

```

connIndexP3=[find(c(:,reserveListIndex(1))),find(c(reserveListIndex(1),:))];

```

```

coLngP3=lng(connIndexP3);

```

```

coLatP3=lat(connIndexP3);

```

```

[reserveListAlterIndex1,reserveList11,reserveList12,reserveList13]=Alterna(lng(reser
veListIndex(1)),lat(reserveListIndex(1)),insLng,insLat,RR,coLngP3,coLatP3);

```

%

```

RR=max(max(c(reserveListIndex(2),find(c(reserveListIndex(2),:)))),max(c(find(c(:,res
erveListIndex(2))),reserveListIndex(2))));

```

```

connIndexP3=[find(c(:,reserveListIndex(2))),find(c(reserveListIndex(2),:))];

```

```

coLngP3=lng(connIndexP3);

```

```

coLatP3=lat(connIndexP3);

```

```

[reserveListAlterIndex2,reserveList21,reserveList22,reserveList23]=Alterna(lng(reser
veListIndex(2)),lat(reserveListIndex(2)),insLng,insLat,RR,coLngP3,coLatP3);

```

%

```

RR=max(max(c(reserveListIndex(3),find(c(reserveListIndex(3),:)))),max(c(find(c(:,res
erveListIndex(3))),reserveListIndex(3))));

```

```

connIndexP3=[find(c(:,reserveListIndex(3))),find(c(reserveListIndex(3),:))];

```

```

coLngP3=lng(connIndexP3);
coLatP3=lat(connIndexP3);
[reserveListAlterIndex3,reserveList31,reserveList32,reserveList33]=Alterna(lng(reserveListIndex(3)),lat(reserveListIndex(3)),insLng,insLat,RR,coLngP3,coLatP3);
%
RR=115.4634;%特殊点
connIndexP3=[find(c(:,reserveListIndex(4))),find(c(reserveListIndex(4),:))];
coLngP3=lng(connIndexP3);
coLatP3=lat(connIndexP3);
[reserveListAlterIndex4,reserveList41,reserveList42,reserveList43]=Alterna(lng(reserveListIndex(4)),lat(reserveListIndex(4)),insLng,insLat,RR,coLngP3,coLatP3);
%
RR=max(max(c(reserveListIndex(5),find(c(reserveListIndex(5),:)))),max(c(find(c(:,reserveListIndex(5))),reserveListIndex(5))));
connIndexP3=[find(c(:,reserveListIndex(5))),find(c(reserveListIndex(5),:))];
coLngP3=lng(connIndexP3);
coLatP3=lat(connIndexP3);
[reserveListAlterIndex5,reserveList51,reserveList52,reserveList53]=Alterna(lng(reserveListIndex(5)),lat(reserveListIndex(5)),insLng,insLat,RR,coLngP3,coLatP3);
%
RR=max(max(c(reserveListIndex(6),find(c(reserveListIndex(6),:)))),max(c(find(c(:,reserveListIndex(6))),reserveListIndex(6))));
connIndexP3=[find(c(:,reserveListIndex(6))),find(c(reserveListIndex(6),:))];
coLngP3=lng(connIndexP3);
coLatP3=lat(connIndexP3);
[reserveListAlterIndex6,reserveList61,reserveList62,reserveList63]=Alterna(lng(reserveListIndex(6)),lat(reserveListIndex(6)),insLng,insLat,RR,coLngP3,coLatP3);
%
RR=max(max(c(reserveListIndex(7),find(c(reserveListIndex(7),:)))),max(c(find(c(:,reserveListIndex(7))),reserveListIndex(7))));
connIndexP3=[find(c(:,reserveListIndex(7))),find(c(reserveListIndex(7),:))];
coLngP3=lng(connIndexP3);
coLatP3=lat(connIndexP3);
[reserveListAlterIndex7,reserveList71,reserveList72,reserveList73]=Alterna(lng(reserveListIndex(7)),lat(reserveListIndex(7)),insLng,insLat,RR,coLngP3,coLatP3);
%
RR=max(max(c(reserveListIndex(8),find(c(reserveListIndex(8),:)))),max(c(find(c(:,reserveListIndex(8))),reserveListIndex(8))));
connIndexP3=[find(c(:,reserveListIndex(8))),find(c(reserveListIndex(8),:))];
coLngP3=lng(connIndexP3);
coLatP3=lat(connIndexP3);
[reserveListAlterIndex8,reserveList81,reserveList82,reserveList83]=Alterna(lng(reserveListIndex(8)),lat(reserveListIndex(8)),insLng,insLat,RR,coLngP3,coLatP3);
%
RR=max(max(c(reserveListIndex(9),find(c(reserveListIndex(9),:)))),max(c(find(c(:,reserveListIndex(9))),reserveListIndex(9))));
connIndexP3=[find(c(:,reserveListIndex(9))),find(c(reserveListIndex(9),:))];
coLngP3=lng(connIndexP3);
coLatP3=lat(connIndexP3);
[reserveListAlterIndex9,reserveList91,reserveList92,reserveList93]=Alterna(lng(reserveListIndex(9)),lat(reserveListIndex(9)),insLng,insLat,RR,coLngP3,coLatP3);

```

```

veListIndex(9)),lat(reserveListIndex(9)),insLng,insLat,RR,coLngP3,coLatP3);

%画图
figure;
scatter(pThrMatCo(1:9,1),pThrMatCo(1:9,2),'k','*');%被炸点
hold;
scatter([coLng1;coLng2],[coLat1;coLat2],'b');%连接点
scatter([reserveList11(1),reserveList21(1),reserveList31(1),reserveList41(1),reserveList51(1),reserveList61(1),reserveList71(1),reserveList81(1),reserveList91(1)],[reserveList11(2),reserveList21(2),reserveList31(2),reserveList41(2),reserveList51(2),reserveList61(2),reserveList71(2),reserveList81(2),reserveList91(2)],'r');
%不改索引的情况下 直接连接
line([lng(61),lng(67)],[lat(61),lat(67)],'color','b');
line([lng(67),reserveList31(1)],[lat(67),reserveList31(2)],'color','r');
line([reserveList11(1),reserveList31(1)],[reserveList11(2),reserveList31(2)],'color','r');
line([reserveList11(1),reserveList21(1)],[reserveList11(2),reserveList21(2)],'color','r');
line([reserveList51(1),reserveList21(1)],[reserveList51(2),reserveList21(2)],'color','r');
line([reserveList51(1),reserveList61(1)],[reserveList51(2),reserveList61(2)],'color','r');
line([reserveList51(1),reserveList41(1)],[reserveList51(2),reserveList41(2)],'color','r');
line([reserveList51(1),lng(42)],[reserveList51(2),lat(42)],'color','r');
line([reserveList61(1),lng(60)],[reserveList61(2),lat(60)],'color','r');
%下一部分
line([reserveList91(1),lng(61)],[reserveList91(2),lat(61)],'color','r');
line([reserveList91(1),reserveList81(1)],[reserveList91(2),reserveList81(2)],'color','r');
line([reserveList71(1),reserveList81(1)],[reserveList71(2),reserveList81(2)],'color','r');
line([reserveList71(1),lng(39)],[reserveList71(2),lat(39)],'color','r');
line([reserveList71(1),lng(34)],[reserveList71(2),lat(34)],'color','r');
text([lng(pThrMatIndex);reserveList11(1);reserveList21(1);reserveList31(1);reserveList41(1);reserveList51(1);reserveList61(1);reserveList71(1);reserveList81(1);reserveList91(1)],[lat(pThrMatIndex);reserveList11(2);reserveList21(2);reserveList31(2);reserveList41(2);reserveList51(2);reserveList61(2);reserveList71(2);reserveList81(2);reserveList91(2)],[cityName(pThrMatIndex);insCityName(reserveList11(3));insCityName(reserveList21(3));insCityName(reserveList31(3));insCityName(reserveList41(3));insCityName(reserveList51(3));insCityName(reserveList61(3));insCityName(reserveList71(3));insCityName(reserveList81(3));insCityName(reserveList91(3))],'fontsize',8,'fontweight','bold');
hold off;
xlabel('经度');
ylabel('纬度');

%
figure;
scatter(pThrMatCo(1:9,1),pThrMatCo(1:9,2),'k','*');%被炸点
hold;
scatter([coLng1;coLng2],[coLat1;coLat2],'b');%连接点
scatter([reserveList12(1),reserveList22(1),reserveList32(1),reserveList42(1),reserveList52(1),reserveList62(1),reserveList72(1),reserveList82(1),reserveList92(1)],[reserveList12(2),reserveList22(2),reserveList32(2),reserveList42(2),reserveList52(2),reserveList62(2),reserveList72(2),reserveList82(2),reserveList92(2)],'r');

```

%不改索引的情况下 直接连接

```
line([lng(61),lng(67)],[lat(61),lat(67)],'color','b');
line([lng(67),reserveList32(1)],[lat(67),reserveList32(2)],'color','r');
line([reserveList12(1),reserveList32(1)],[reserveList12(2),reserveList32(2)],'color','r');
line([reserveList12(1),reserveList22(1)],[reserveList12(2),reserveList22(2)],'color','r');
line([reserveList52(1),reserveList22(1)],[reserveList52(2),reserveList22(2)],'color','r');
line([reserveList52(1),reserveList62(1)],[reserveList52(2),reserveList62(2)],'color','r');
line([reserveList52(1),reserveList42(1)],[reserveList52(2),reserveList42(2)],'color','r');
line([reserveList52(1),lng(42)],[reserveList52(2),lat(42)],'color','r');
line([reserveList62(1),lng(60)],[reserveList62(2),lat(60)],'color','r');
```

%下一部分

```
line([reserveList92(1),lng(61)],[reserveList92(2),lat(61)],'color','r');
line([reserveList92(1),reserveList82(1)],[reserveList92(2),reserveList82(2)],'color','r');
line([reserveList72(1),reserveList82(1)],[reserveList72(2),reserveList82(2)],'color','r');
line([reserveList72(1),lng(39)],[reserveList72(2),lat(39)],'color','r');
line([reserveList72(1),lng(34)],[reserveList72(2),lat(34)],'color','r');
text([lng(pThrMatIndex);reserveList12(1);reserveList22(1);reserveList32(1);reserveList42(1);reserveList52(1);reserveList62(1);reserveList72(1);reserveList82(1);reserveList92(1)],[lat(pThrMatIndex);reserveList12(2);reserveList22(2);reserveList32(2);reserveList42(2);reserveList52(2);reserveList62(2);reserveList72(2);reserveList82(2);reserveList92(2)],[cityName(pThrMatIndex);insCityName(reserveList12(3));insCityName(reserveList22(3));insCityName(reserveList32(3));insCityName(reserveList42(3));insCityName(reserveList52(3));insCityName(reserveList62(3));insCityName(reserveList72(3));insCityName(reserveList82(3));insCityName(reserveList92(3))],'fontsize',8,'fontweight','bold');
hold off;
xlabel('经度');
ylabel('纬度');
```

%

figure;

scatter(pThrMatCo(1:9,1),pThrMatCo(1:9,2),'k','*');%被炸点

hold;

scatter([coLng1;coLng2],[coLat1;coLat2],'b');%连接点

```
scatter([reserveList13(1),reserveList23(1),reserveList33(1),reserveList43(1),reserveList53(1),reserveList63(1),reserveList73(1),reserveList83(1),reserveList93(1)],[reserveList13(2),reserveList23(2),reserveList33(2),reserveList43(2),reserveList53(2),reserveList63(2),reserveList73(2),reserveList83(2),reserveList93(2)],'r');
```

%不改索引的情况下 直接连接

```
line([lng(61),lng(67)],[lat(61),lat(67)],'color','b');
line([lng(67),reserveList33(1)],[lat(67),reserveList33(2)],'color','r');
line([reserveList13(1),reserveList33(1)],[reserveList13(2),reserveList33(2)],'color','r');
line([reserveList13(1),reserveList23(1)],[reserveList13(2),reserveList23(2)],'color','r');
line([reserveList53(1),reserveList23(1)],[reserveList53(2),reserveList23(2)],'color','r');
line([reserveList53(1),reserveList63(1)],[reserveList53(2),reserveList63(2)],'color','r');
line([reserveList53(1),reserveList43(1)],[reserveList53(2),reserveList43(2)],'color','r');
line([reserveList53(1),lng(42)],[reserveList53(2),lat(42)],'color','r');
line([reserveList63(1),lng(60)],[reserveList63(2),lat(60)],'color','r');
```

%下一部分

```

line([reserveList93(1),lng(61)],[reserveList93(2),lat(61)],'color','r');
line([reserveList93(1),reserveList83(1)],[reserveList93(2),reserveList83(2)],'color','r');
line([reserveList73(1),reserveList83(1)],[reserveList73(2),reserveList83(2)],'color','r');
line([reserveList73(1),lng(39)],[reserveList73(2),lat(39)],'color','r');
line([reserveList73(1),lng(34)],[reserveList73(2),lat(34)],'color','r');
text([lng(pThrMatIndex);reserveList13(1);reserveList23(1);reserveList33(1);reserveList43(1);reserveList53(1);reserveList63(1);reserveList73(1);reserveList83(1);reserveList93(1)],[lat(pThrMatIndex);reserveList13(2);reserveList23(2);reserveList33(2);reserveList43(2);reserveList53(2);reserveList63(2);reserveList73(2);reserveList83(2);reserveList93(2)],[cityName(pThrMatIndex);insCityName(reserveList13(3));insCityName(reserveList23(3));insCityName(reserveList33(3));insCityName(reserveList43(3));insCityName(reserveList53(3));insCityName(reserveList63(3));insCityName(reserveList73(3));insCityName(reserveList83(3));insCityName(reserveList93(3))],'fontsize',8,'fontweight','bold');
hold off;
xlabel('经度');
ylabel('纬度');

%Problem_4.m
%随机摧毁 10%的节点
clear
clc
load('Problem_1.mat');

figure;
gplot(c,r,'- o')
text(lng,lat,cityName,'fontsize',7,'fontweight','bold');%6
hold;
line([lng(90),lng(41)],[lat(90),lat(41)],'color','r');
line([lng(45),lng(12)],[lat(45),lat(12)],'color','r');
line([lng(11),lng(3)],[lat(11),lat(3)],'color','r');
line([lng(1),lng(4)],[lat(1),lat(4)],'color','r');
line([lng(22),lng(33)],[lat(22),lat(33)],'color','r');
line([lng(100),lng(112)],[lat(100),lat(112)],'color','r');
line([lng(127),lng(138)],[lat(127),lat(138)],'color','r');
hold off;
xlabel('经度');
ylabel('纬度');
c(90,41)=GetDistance(lng(90),lat(90),lng(41),lat(41));
c(45,12)=GetDistance(lng(45),lat(45),lng(12),lat(12));
c(11,3)=GetDistance(lng(11),lat(11),lng(3),lat(3));
c(1,4)=GetDistance(lng(1),lat(1),lng(4),lat(4));
c(22,33)=GetDistance(lng(22),lat(22),lng(33),lat(33));
c(100,112)=GetDistance(lng(100),lat(100),lng(112),lat(112));
c(127,138)=GetDistance(lng(127),lat(127),lng(138),lat(138));
z=c;
fPointList=zeros(1);
for i=1:14
    fPointList(i)=round(unifrnd(1,139));
    while(length(unique(fPointList))<i)

```

```

        fPointList(i)=round(unifrnd(1,139));
    end
end
%导入保存数据 Problem_4.mat 还原当时随机点
load('Problem_4.mat');
c(fPointList,:)=0;
c(:,fPointList)=0;
figure;
gplot(c,r,'- o');
text(lng,lat,cityName,'fontsize',6,'fontweight','bold');
hold;
scatter(lng(fPointList),lat(fPointList),'r');
hold off;
xlabel('经度');
ylabel('纬度');
%Alterna.m
%insIndex 在范围内的节点索引
%insP1 1.与被炸点距离最短 lng lat 索引
%insP2 2.与连接点距离最短 参数同上
%insP3 3.被炸和连接距离最短 参数同上
function [insIndex,insP1,insP2,insP3]=Alterna(lng0,lat0,lng,lat,R,coLng,coLat)
insP1=-inf;
insP2=-inf;
insP3=-inf;
insIndex=zeros(1);

count=1;
vlu1=inf;
vlu2=inf;
vlu3=inf;
for i=1:length(lng)
    temp1=GetDistance(lng(i),lat(i),lng0,lat0);
    if(temp1<=R)
        insIndex(count)=i;
        count=count+1;
        temp2=0;
        for j=1:length(coLng)
            %计算备战节点与之前连接节点的距离
            temp2=temp2+GetDistance(coLng(j),coLat(j),lng(i),lat(i));
        end
        temp3=temp1+temp2;
        %最小值 1.与被炸 2.与连接 3.被炸和连接
        if(vlu1>temp1)
            vlu1=temp1;
            insP1=[lng(i),lat(i),i];
        end
        if(vlu2>temp2)
            vlu2=temp2;
            insP2=[lng(i),lat(i),i];
        end
    end
end

```



```

        end
        if(vlu3>temp3)
            vlu3=temp3;
            insP3=[lng(i),lat(i),i];
        end
    end
end
end
%AlternaP3.m
%insIndex 在范围内的节点索引
%insP1 1.与被炸点的集合距离最短 lng lat 索引
%insP2 2.与连接点距离最短 参数同上
%insP3 3.被炸点集合和连接距离最短 参数同上
%lng0 lat0 中心点坐标
%R 中心点与连接点的最长距离
%coLng coLat 连接点坐标
%fList 被炸点集合
function [insIndex,insP1,insP2,insP3]=AlternaP3(fList,lng0,lat0,lng,lat,R,coLng,coLat)
insP1=-inf;
insP2=-inf;
insP3=-inf;
insIndex=zeros(1);

count=1;
vlu1=inf;
vlu2=inf;
vlu3=inf;
for i=1:length(lng)
    temp=GetDistance(lng(i),lat(i),lng0,lat0);
    if(temp<=R)
        insIndex(count)=i;
        count=count+1;
        temp1=0;
        temp2=0;
        for k=1:length(fList)
            temp1=temp1+GetDistance(fList(1),fList(2),lng(i),lat(i));
        end
        for j=1:length(coLng)
            %计算备战节点与之前连接节点的距离
            temp2=temp2+GetDistance(coLng(j),coLat(j),lng(i),lat(i));
        end
        temp3=temp1+temp2;
        %最小值 1.与被炸 2.与连接 3.被炸和连接
        if(vlu1>temp1)
            vlu1=temp1;
            insP1=[lng(i),lat(i),i];
        end
        if(vlu2>temp2)
            vlu2=temp2;

```

```

        insP2=[lng(i),lat(i),i];
    end
    if(vlu3>temp3)
        vlu3=temp3;
        insP3=[lng(i),lat(i),i];
    end
end
end
end
%Center.m
function pThr=Center(cityListIndex,lat,lng)
%将 Lat1 和 Lon1 从度转换为弧度。
latList=lat(cityListIndex)*pi/180;
lngList=lng(cityListIndex)*pi/180;
%将第一个位置的 lat/lon 转换为笛卡尔坐标。
X=cos(latList).*cos(lngList);
Y=cos(latList).*sin(lngList);
Z=sin(latList);
%计算加权平均 x, y 和 z 坐标
x=sum(X)/length(X);
y=sum(Y)/length(Y);
z=sum(Z)/length(Z);
%将平均 x、y、z 坐标转换为纬度和经度。
lngList1 = atan2(y,x);
Hyp = sqrt(x * x + y * y);
latList1 = atan2(z,Hyp);
pThr=[lngList1*180/pi,latList1*180/pi];
end
%Distance.m
function d=Distance(lng,lat)
d=zeros(length(lat));
d(logical(eye(size(d))))=inf;
for i=1:length(lat)-1
    for j=i+1:length(lng)
        d(i,j)=GetDistance(lng(i),lat(i),lng(j),lat(j));
        d(j,i)=d(i,j);
    end
end
end
end
%GetDistance.m
function s=GetDistance(lng1,lat1,lng2,lat2)
radLat1 = lat1 * pi / 180.0;
radLat2 = lat2 * pi / 180.0;
a = radLat1 - radLat2;
b = (lng1 - lng2)* pi / 180.0;
s = 2 * asin(sqrt(power(sin(a/2),2) +
cos(radLat1)*cos(radLat2)*power(sin(b/2),2)));
s = s * 6378.137;
end

```

附录 3

序号	城市	东经	北纬	序号	城市	东经	北纬
1	北流	110. 21	22. 42	54	鹤壁	114. 11	35. 54
2	百色	106. 36	23. 54	55	辉县	113. 47	35. 27
3	防城港	108. 2	21. 37	56	焦作	113. 12	35. 14
4	贵港	109. 36	23. 06	57	济源	112. 35	35. 04
5	桂平	110. 04	23. 22	58	灵宝	110. 52	34. 31
6	河池	108. 03	24. 42	59	林州	113. 49	36. 03
7	合山	108. 52	23. 47	60	漯河	114. 02	33. 33
8	赁祥	106. 44	22. 07	61	平顶山	113. 17	33. 44
9	钦州	108. 37	21. 57	62	濮阳	115. 01	35. 44
10	梧州	111. 2	23. 29	63	沁阳	112. 57	35. 05
11	玉林	110. 09	22. 38	64	汝州	112. 5	34. 09
12	宜州	108. 4	24. 28	65	三门峡	111. 12	34. 47
13	安顺	105. 55	26. 14	66	商丘	115. 38	34. 26
14	毕节	105. 18	27. 18	67	卫辉	114. 03	35. 24
15	赤水	105. 42	28. 34	68	舞钢	113. 3	33. 17
16	凯里	107. 58	26. 35	69	项城	114. 54	33. 26
17	清镇	106. 27	26. 33	70	荥阳	113. 21	34. 46
18	铜仁	109. 12	27. 43	71	新密	113. 22	34. 31
19	兴义	104. 53	25. 05	72	新乡	113. 52	35. 18
20	海口	110. 2	20. 02	73	新郑	113. 43	34. 24
21	儋州	109. 34	19. 31	74	许昌	113. 49	34. 01
22	琼海	110. 28	19. 14	75	偃师	112. 47	34. 43
23	琼山	110. 21	19. 59	76	义马	111. 55	34. 43
24	通什	109. 31	18. 46	77	禹州	113. 28	34. 09
25	安国	115. 2	38. 24	78	周口	114. 38	33. 37
26	霸州	116. 24	39. 06	79	驻马店	114. 01	32. 58
27	泊头	116. 34	38. 04	80	阿城	126. 58	45. 32
28	沧州	116. 52	38. 18	81	安达	125. 18	46. 24
29	定州	115	38. 3	82	北安	126. 31	48. 15
30	丰南	118. 06	39. 34	83	大庆	125. 01	46. 36
31	高碑店	115. 51	39. 2	84	富锦	132. 02	47. 15
32	蒿城	114. 5	38. 02	85	海林	129. 21	44. 35
33	河间	116. 05	38. 26	86	海伦	126. 57	47. 28
34	黄骅	117. 21	38. 21	87	鹤岗	130. 16	47. 2
35	晋州	115. 02	38. 02	88	密山	131. 5	45. 32
36	冀州	115. 33	37. 34	89	讷河	124. 51	48. 29
37	廊坊	116. 42	39. 31	90	宁安	129. 28	44. 21
38	鹿泉	114. 19	38. 04	91	七台河	130. 49	45. 48
39	南宫	115. 23	37. 22	92	双城	126. 15	45. 22
40	任丘	116. 07	38. 42	93	尚志	127. 55	45. 14
41	三河	117. 04	39. 58	94	双鸭山	131. 11	46. 38
42	沙河	114. 3	36. 51	95	绥芬河	131. 11	44. 25
43	深州	115. 32	38. 01	96	绥化	126. 59	46. 38
44	武安	114. 11	36. 42	97	铁力	128. 01	46. 59

序号	城市	东经	北纬	序号	城市	东经	北纬
45	邢台	114.3	37.04	98	五常	127.11	44.55
46	辛集	115.12	37.54	99	五大连池	126.07	48.38
47	新乐	114.41	38.2	100	伊春	128.56	47.42
48	涿州	115.59	39.29	101	肇东	125.58	46.04
49	遵化	117.58	40.11	102	安陆	113.41	31.15
50	长葛	113.47	34.12	103	当阳	111.47	30.5
51	登封	113.02	34.27	104	丹江口	108.3	32.33
52	邓州	112.05	32.42	105	大冶	114.58	30.06
53	巩义	112.58	34.46	106	恩施	109.29	30.16
107	鄂州	114.52	30.23	160	珲春	130.22	42.52
108	广水	113.48	31.37	161	集安	126.11	41.08
109	洪湖	113.27	29.48	162	蛟河	127.21	43.42
110	黄州	114.52	30.27	163	九台	125.51	44.09
111	荆门	112.12	31.02	164	辽源	125.09	42.54
112	荆沙	112.16	30.18	165	临江	126.53	41.49
113	老河口	111.4	32.23	166	龙井	129.26	42.46
114	利川	108.56	30.18	167	梅河口	125.4	42.32
115	麻城	115.01	31.1	168	舒兰	126.57	44.24
116	浦圻	113.51	29.42	169	松原	124.49	45.11
117	潜江	112.53	30.26	170	洮南	122.47	45.2
118	石首	112.24	29.43	171	延吉	129.3	42.54
119	随州	113.22	31.42	172	榆树	126.32	44.49
120	天门	113.1	60.39	173	常熟	120.43	31.39
121	武穴	115.33	29.51	174	常州	119.58	31.47
122	咸宁	114.17	29.53	175	丹阳	119.32	32
123	仙桃	113.27	30.22	176	东台	120.19	32.51
124	孝感	113.54	30.56	177	高邮	119.27	32.47
125	宜城	112.15	31.42	178	海门	121.09	31.53
126	应城	113.33	30.57	179	淮安	119.09	33.3
127	枣阳	112.44	32.07	180	淮阴	119.02	33.36
128	枝城	111.27	30.23	181	江都	119.32	32.26
129	钟祥	112.34	31.1	182	姜堰	120.08	32.34
130	洪江	109.59	27.07	183	江阴	120.17	31.54
131	津市	111.52	29.38	184	靖江	120.17	32.02
132	吉首	109.43	28.18	185	金坛	119.33	31.46
133	耒阳	112.51	26.24	186	昆山	120.57	31.23
134	冷水江	111.26	27.42	187	连去港	119.1	34.36
135	冷水滩	111.35	26.26	188	溧阳	119.29	31.26
136	涟源	111.41	27.41	189	南通	120.51	32.01
137	醴陵	113.3	27.4	190	邳州	117.59	34.19
138	临湘	113.27	29.29	191	启乐	121.39	31.48
139	浏阳	113.37	28.09	192	如皋	120.33	32.23
140	娄底	111.59	27.44	193	宿迁	118.18	33.58
141	汨罗	113.03	28.49	194	苏州	120.37	31.19
142	韶山	112.29	27.54	195	太仓	121.06	31.27

序号	城市	东经	北纬	序号	城市	东经	北纬
143	邵阳	111.28	27.14	196	泰兴	120.01	32.1
144	武冈	110.37	26.43	197	泰州	119.54	32.3
145	湘潭	112.53	27.52	198	通州	121.03	32.05
146	湘乡	112.31	27.44	199	吴江	120.39	31.1
147	益阳	112.2	28.36	200	兴化	119.5	32.56
148	永州	111.37	26.13	201	新沂	118.2	34.22
149	沅江	112.22	28.5	202	盐在	120.08	33.22
150	张家界	110.29	29.08	203	扬中	119.49	32.14
151	株洲	113.09	27.51	204	宜兴	119.49	31.21
152	资兴	113.13	25.58	205	仪征	119.1	32.16
153	白山	126.26	41.56	206	张家港	120.32	31.52
154	大安	124.18	45.3	207	镇江	119.27	32.11
155	德惠	125.42	44.32	208	德兴	117.35	28.57
156	敦化	128.13	43.22	209	丰城	115.48	28.12
157	公主岭	124.49	43.31	210	高安	115.22	28.25
158	和龙	129	42.32	211	吉安	114.58	27.07
159	桦甸	126.44	42.58	212	井冈山	114.1	26.34
213	乐平	117.08	28.58	266	菏泽	115.26	35.14
214	临川	116.21	27.59	267	胶南	119.58	35.53
215	瑞昌	115.38	29.4	268	胶州	120	36.17
216	瑞金	116.01	25.53	269	即墨	120.28	36.22
217	上饶	117.58	25.27	270	莱芜	117.4	36.12
218	新余	114.56	27.48	271	莱西	120.31	36.52
219	宜春	114.23	27.47	272	莱阳	120.42	36.58
220	樟树	115.32	28.03	273	莱州	119.57	37.1
221	北票	120.47	41.48	274	乐陵	117.12	37.44
222	本溪	123.46	41.18	275	聊城	115.57	36.26
223	朝阳	120.27	41.34	276	临清	115.42	36.51
224	大石桥	122.31	40.37	277	临沂	118.2	35.03
225	东港	124.08	39.53	278	龙口	120.21	37.39
226	凤城	124.02	40.28	279	蓬莱	120.45	37.48
227	抚顺	123.54	41.51	280	平度	119.58	36.47
228	阜新	121.39	42.01	281	青州	118.28	36.42
229	盖州	122.21	40.24	282	曲阜	116.58	35.36
230	海城	122.43	40.51	283	日照	119.32	35.23
231	葫芦岛	120.51	40.45	284	乳山	121.31	36.54
232	开原	124.02	42.32	285	寿光	118.44	36.53
233	辽阳	123.12	41.16	286	泰安	117.08	36.11
234	凌海	121.21	41.1	287	滕州	117.09	35.06
235	凌源	119.22	41.14	288	威海	122.07	37.31
236	盘锦	122.03	41.07	289	文登	122.03	37.12
237	普兰店	121.58	39.23	290	新泰	117.45	35.54
238	铁法	123.32	42.28	291	兖州	116.49	35.32
239	铁岭	123.51	42.18	292	禹城	116.39	36.56
240	瓦房店	122	39.37	293	枣庄	117.33	34.52

序号	城市	东经	北纬	序号	城市	东经	北纬
241	兴城	120.41	40.37	294	章丘	117.32	36.43
242	新民	122.49	41.59	295	招远	120.23	37.21
243	庄河	122.58	39.41	296	诸城	119.24	35.59
244	赤峰	118.58	42.17	297	淄博	118.03	36.48
245	东胜	109.59	39.48	298	邹城	116.58	35.24
246	额尔古纳	120.11	50.13	299	高平	112.55	35.48
247	丰镇	113.09	40.27	300	古交	112.09	37.54
248	根河	121.29	50.48	301	河津	110.41	35.35
249	霍林郭勒	119.38	45.32	302	侯马	111.21	35.37
250	集宁	113.06	41.02	303	霍州	111.42	36.34
251	临河	107.22	40.46	304	介休	111.55	37.02
252	通辽	122.16	43.37	305	晋城	112.51	35.3
253	乌兰浩特	122.03	46.03	306	潞城	113.14	36.21
254	乌海	106.48	39.4	307	朔州	112.26	39.19
255	牙克石	120.4	49.17	308	孝义	111.48	37.08
256	扎兰屯	122.47	48	309	忻州	112.43	38.24
257	石嘴山	106.22	39.02	310	阳泉	113.34	37.51
258	吴忠	106.11	37.59	311	永济	110.27	34.52
259	安丘	119.12	36.25	312	原平	112.42	38.43
260	滨州	118.02	37.22	313	榆次	112.43	37.41
261	昌邑	119.24	39.52	314	安康	109.01	32.41
262	德州	116.17	37.26	315	韩城	110.27	35.28
263	东营	118.3	37.27	316	华阴	110.05	34.34
264	肥城	116.46	36.14	317	商州	109.57	33.52
265	高密	119.44	36.22	318	铜川	109.07	35.06
319	渭南	109.3	34.3	372	富阳	119.57	30.03
320	咸阳	108.43	34.2	373	海宁	120.42	30.32
321	兴平	108.29	34.18	374	湖州	120.06	30.52
322	巴中	106.43	31.51	375	建德	119.16	29.29
323	崇州	103.4	30.39	376	江山	118.37	28.45
324	达川	107.29	31.14	377	嘉兴	120.45	30.46
325	德阳	104.22	31.09	378	兰溪	119.28	29.12
326	都江堰	103.37	31.01	379	临海	121.08	28.51
327	峨眉山	103.29	29.36	380	丽水	119.54	28.27
328	涪陵	107.22	29.42	381	龙泉	119.08	28.04
329	广汉	104.15	30.58	382	平湖	121.01	30.42
330	广元	105.51	32.28	383	衢州	118.52	28.58
331	华蓥	106.44	30.26	384	瑞安	120.38	27.48
332	简阳	104.32	30.24	385	上虞	120.52	30.01
333	江油	104.42	31.48	386	绍兴	120.34	30
334	阆中	105.58	31.36	387	台州	121.27	28.41
335	乐山	103.44	29.36	388	桐乡	120.32	30.38
336	泸州	105.24	28.54	389	温岭	121.21	28.22
337	南充	106.04	30.49	390	温州	120.39	28.01
338	内江	105.02	29.36	391	萧山	120.16	30.09

序号	城市	东经	北纬	序号	城市	东经	北纬
339	彭州	103.57	30.59	392	义乌	120.04	29.18
340	邛崃	103.28	30.26	393	乐清	120.58	28.08
341	遂宁	105.33	30.31	394	余杭	120.18	30.26
342	万县	108.21	30.5	395	余姚	121.1	30.02
343	万源	108.03	32.03	396	永康	120.01	29.54
344	雅安	102.59	29.59	397	舟山	122.06	30.01
345	自贡	104.46	29.23	398	诸暨	120.14	29.43
346	资阳	104.38	30.09	399	合川市	106.15	30.02
347	台北市	121.3	25.03	400	江津	106.16	29.18
348	阿图什	76.08	39.42	401	南川	107.05	29.1
349	博乐	82.08	44.57	402	永川	105.53	29.23
350	昌吉	87.18	44.02	403	兰浩特	122.08	46.07
351	阜康	87.58	44.09	404	郑州	113.65	34.77
352	克拉玛依	84.51	45.36	405	三亚	109.50	18.20
353	库尔勒	86.07	41.46	406	海口	110.35	20.02
354	奎屯	84.56	44.27	407	蚌埠	117.21	32.56
355	石河子	86	44.18	408	亳州	115.47	33.52
356	吐鲁番	89.11	42.54	409	巢湖	117.52	31.36
357	伊宁	81.2	43.55	410	滁州	118.18	32.18
358	保山	99.1	25.08	411	阜阳	115.48	32.54
359	楚雄	101.32	25.01	412	贵池	117.28	30.39
360	东川	103.12	26.06	413	淮北	116.47	33.57
361	景洪	100.48	22.01	414	淮南	116.58	32.37
362	开远	103.13	23.43	415	黄山	118.18	29.43
363	曲靖	103.48	25.3	416	界首	115.21	33.15
364	瑞丽	97.5	24	417	马鞍山	118.28	31.43
365	思茅	100.58	22.48	418	明光	117.58	32.47
366	宣威	104.06	26.13	419	宿州	116.58	33.38
367	玉溪	102.32	24.22	420	天长	118.59	32.41
368	昭通	103.42	27.2	421	铜陵	117.48	30.56
369	慈溪	121.15	30.11	422	宣州	118.44	30.57
370	东阳	120.14	29.16	423	长乐	119.31	25.58
371	奉化	121.24	29.39	424	福安	119.39	27.06
425	福清	119.23	25.42	460	鹤山	112.57	22.46
426	建瓯	118.2	27.03	461	河源	114.41	23.43
427	建阳	118.07	27.21	462	花都	113.12	23.23
428	晋江	118.35	24.49	463	化州	110.37	21.39
429	龙海	117.48	24.26	464	惠阳	114.28	22.48
430	南安	118.23	24.57	465	惠州	114.22	23.05
431	南平	118.1	26.38	466	江门	113.04	22.35
432	宁德	119.31	26.39	467	揭阳	116.21	22.32
433	莆田	119.01	24.26	468	开平	112.4	22.22
434	泉州	118.36	24.56	469	乐昌	113.21	25.09
435	三明	117.36	26.13	470	雷州	110.04	20.54
436	邵武	117.29	27.2	471	廉江	110.17	21.37

序号	城市	东经	北纬	序号	城市	东经	北纬
437	石狮	118.38	24.44	472	连州	112.23	24.48
438	武夷山	118.02	27.46	473	罗定	111.33	22.46
439	永安	117.23	25.58	474	茂名	110.53	21.4
440	漳平	117.24	25.17	475	梅州	116.07	24.19
441	漳州	117.39	24.31	476	南海	113.09	23.01
442	白银	104.12	36.33	477	番禺	113.22	22.57
443	嘉峪关	98.14	39.48	478	普宁	116.1	23.18
444	金昌	102.1	38.28	479	清远	113.01	23.42
445	酒泉	98.31	39.44	480	三水	112.52	23.1
446	临夏	103.12	35.37	481	汕尾	115.21	22.47
447	平凉	106.4	35.32	482	顺德	113.15	22.5
448	武威	102.39	37.56	483	四会	112.41	23.21
449	西峰	107.4	35.45	484	台山	112.48	22.15
450	潮阳	116.36	23.16	485	吴川	110.47	21.26
451	潮州	116.38	23.4	486	新会	113.01	22.32
452	澄海	116.46	23.28	487	兴宁	115.43	24.09
453	从化	113.33	23.33	488	阳春	111.48	22.1
454	东莞	113.45	23.02	489	阳江	111.58	21.5
455	恩平	112.19	22.12	490	英德	113.22	24.1
456	佛山	113.06	23.02	491	云浮	112.02	22.57
457	高明	112.5	22.53	492	增城	113.49	23.18
458	高要	112.26	23.02	493	中山	113.22	22.31
459	高州	110.5	21.54	494	珠海	113.34	22.17