

# “华为杯”第十五届中国研究生 数学建模竞赛

题 目                      光传送网建模与价值评估

---

## 摘                      要：

在通信技术日益发达的今天，各种各样的信息传输技术也相继出现，如相移键控（PSK）、正交幅度调制（QAM）等调制解调技术。各种技术都有它的优缺点和适合它的工作环境。由于每种技术都并不完美，总有改进的空间，所以对一些技术进行适当的改善，增加传输准确率，抑或提高传输效率，又或者降低传输成本等等，都是值得我们去研究和讨论的。

本文介绍了 QAM 的概念、产生、优点和缺点，QAM 依靠其高效的带宽效率成为一种被广泛使用的调制解调技术。QAM 的每个星座有一种最佳的星座图分布和最低的误码率。常用的 QAM 的星座图，如 QPSK、8QAM、16QAM 的星座图，并不是最佳星座图，但易于硬件的实现。

第二部分运用了图论的有关知识，将构建光纤的传输网络的问题化成求网络价值的最大值的优化问题，用遗传算法结合 TSP 来构建模型。同时还使用了 Dijkstra 算法来寻求图中的最短路径问题。

我们通过 MATLAB 仿真改变 16QAM 方案中星座点的位置、数量，产生比原始 8QAM 具有更低 SNR 容限点，以此来提高系统容忍噪声的能力，从而延长链路的总长度。通过获取改善的 16QAM 的 SNR 容限点的性能，并且找到最佳

的 16QAM 星座图。

**关键词：**QPSK, MQAM, 误码率, 信噪比, 传送网规划, 星座图优, 遗传算法, TSP, 优化问题, Dijkstra 算法

# 目录

<b>1.问题重述.....</b>	<b>3</b>
1.1 问题背景.....	3
1.2 问题提出.....	4
<b>2.问题假设.....</b>	<b>5</b>
<b>3.符号说明.....</b>	<b>5</b>
<b>4.问题一模型建立与求解.....</b>	<b>6</b>
4.1 问题分析.....	6
4.2 算法介绍.....	6
4.3 QPSK 与 4QAM 的有效性和可靠性对比.....	9
4.5 模型建立.....	10
4.6 实验结果及分析.....	11
<b>5.问题二的模型建立与求解.....</b>	<b>13</b>
5.1 子问题 1 的分析与解答.....	13
5.2 子问题 2 的分析与解答.....	20
5.3 子问题 3 的分析与解答.....	26
<b>6.问题三模型建立与求解.....</b>	<b>28</b>
6.1 8QAM 调制与解调原理.....	28
6.2 QAM 调制原理.....	29
6.3 QAM 星座图设计.....	29
6.4 常见星座图简介.....	29
6.5 星座图的性能评价指标.....	30
6.6 8QAM 星座图的构造.....	31
6.8 结果分析.....	33
<b>参考文献.....</b>	<b>35</b>
<b>附录（代码） .....</b>	<b>36</b>

# 1.问题重述

## 1.1 问题背景

从诞生至今，50 多年里基于数字光纤通信技术的光传送网构建起了全球通信的骨架。从城市内的传输，直到跨越大洋的传输，光传送网为人类提供了大容量、高可靠性和低能耗的信息传输管道，人类对通信容量的追求也成为光传送技术发展的源源不断的动力。

光传送网的规划与建设是运营商、设备商以及政府必须考虑的课题。光传送的基本规律是——在相同技术条件下传输的容量会随着传输距离增加而减小。网络规划者需要在有限资源的条件下，综合考虑传输距离，传输容量、网络拓扑等各种因素，以最大化网络的价值。本课题中，请你们站在上述角度，从底层物理出发为光传送链路建模，制定光传送网规划，探索光传送网有关规律。

## 1.2 问题提出

### 1.1.1 纠前误码率与信噪比计算

星座图的编码分布模式也称为调制格式，对于给定的调制格式，BER 和 SNR 呈一一对应的关系，纠前 BER 门限对应的 SNR 记做“SNR 容限点”。给出本文中所示的三种调制格式及编码方式（相邻星座点距离相等），每个符号等概率出现，分别称为 QPSK，8QAM (Quadrature Amplitude Modulation, QAM)，16QAM。请给出 BER 与 SNR 的关系曲线，BER=0.02 时 SNR 容限点分别为多少？

### 1.1.2 光链路性能计算

当单跨传输距离为 80km 和 100km 两种情况，以纠前误码率 0.02 为门限，图 5 给出的传输格式最远的传输距离（每跨距离×跨段数量）是多少？

1.2.1 如果连接数从 16 增加到 33 条时，不考虑中间节点，给出你们的两个网络规划及其价值。网络价值最多是多少？

1.2.2 存在中间节点，且两个节点之间可以有多个连接的情况下，重新解决问题-1 并给出所有中间节点传输容量的分配，假定每条链路容量可任意分配，只要总容量不超过表 1 的规定。如果由市扩大为省（区）影响如何？（人口请从网上查找）

1.2.3 光传送网络价值有多个侧面，例如从运营商的角度，连接经济发达的地区会带来更多的收入，从政府的角度保障发展相对滞后地区的通信是均衡发展的要求等。你认为制定光传送网络规划的目标函数应该是什么？前面制定的规划有无变化？

### 1.3.1 改善星座图

由第一问可知，纠前 BER 不变时，降低 SNR 容限点可以提高系统容忍噪声的能力，从而延长链路的总长度。请尝试任意改变 16QAM 方案中星座点的位置、数量或每个点的概率，探索产生比图 5 中 8QAM（相邻各星座点之间距离相等）具有更低 SNR 容限点的调制方案？调制格式的信息熵需保持为 3bit。

信息熵定义为：

$$\Omega = -\sum_{k=1}^N p_k \log_2(p_k)$$

其中  $p_k$  为每个符号状态出现的概率，N 为状态数。图 5 所示的等概率情况下，QPSK、8QAM 和 16QAM 的信息熵分别为 2bit, 3bit 和 4bit。

## 2.问题假设

- (1) 非线性噪声等于单个放大器噪声的 2/3。
- (2) 当两个城市距离超过 3000km 时，其节点距离为 INF，网络价值为 0。
- (3) 在本文第二问题中，北京市和天津市合称为北京，广州市和深圳市合称为深圳。

## 3.符号说明

符号	意义
$a_i$ 、 $b_i$	坐标点
$E_0$	最小幅度的信号能量

---

$d_{MQAM}$	信号间距离
$p_e$	符号错误概率
$E_{av}$	平均能量
$T$	符号间隔
$M$	QAM 电平系数
$\theta_k$	角度
$In$	输入信号
$Gain$	增益函数
$L$	段长
$MQAM$	多元正交振幅调
$P_n$	线性噪声功率
$P_{nml}$	非线性噪声功率
$S_{NL}$	非线性噪声信号输入
$S_{NA}$	放大器噪声信号输入

---

## 4.问题一模型建立与求解

### 4.1 问题分析

光传输链路由多个相同跨段的级联而成,且本文分别以 80km 和 100km 为一个跨段,同时信号每传输 15km,光功率衰减一半,需要用放大器对光功率进行补偿。在信号、噪声同步放大的同时,放大器还引入自发辐射噪声,另一方面,光纤作为一种传输介质,其本身的非线性效应也会等效地引入噪声。放大器的自发辐射噪声和光纤的非线性噪声都是加性噪声。非线性噪声在每跨开头叠加,放大器噪声在每跨末尾叠加;每跨中间信噪比保持不变。每跨叠加的噪声间相互独

立。即我们要计算出自发辐射噪声和非线性噪声，将其与输入信号叠加，运用三种编码方式，得出最远传输距离。

## 4.2 算法介绍

由于 4QAM, 8QAM, 16QAM 是 MQAM 的一种，且 QPSK 在某种条件下与 4QAM 等价，本文稍后介绍 QPSK 与 4QAM 的关系，所以通过介绍 MQAM 的原理等价于三种编码方式。

对于 MQAM, 多元正交振幅调制是两个相互正交的同频载波调制后调制信号的双边带调制，利用这种已调信号在同一带宽内频谱正交的性质来实现两路并行的数字信息传输。MQAM 信号调制框图如图 4.1 所示：

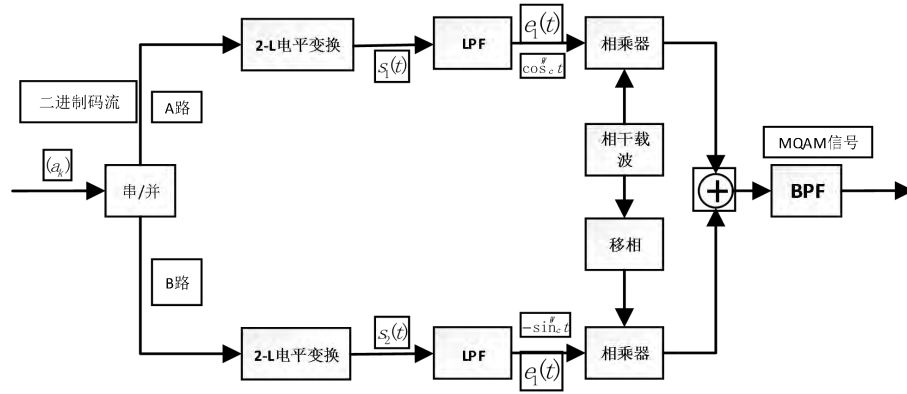


图 4.1 MQAM 信号调制框图

经过 MQAM 调制系统后 MQAM 信号表示式可表示为：

$$S_{MQAM}(t) = A_0 a_i \cos \omega_0 t + A_0 b_i \sin \omega_0 t = \sqrt{E_0} a_i \phi_1(t) + \sqrt{E_0} b_i \phi_2(t)$$

MQAM 信号最小幅度的信号能量为：

$$E_0 = \frac{A_0^2 T}{2}$$

若已调信号的最大幅度为 1，则 MQAM 信号星座图上信号点间的最小距离为：

$$d_{MQAM} = \frac{\sqrt{2}}{L-1} = \frac{\sqrt{2}}{\sqrt{M}-1}$$

MQAM 信号采用正交相干解调方法时，其解调器原理图如图 4.2 所示。

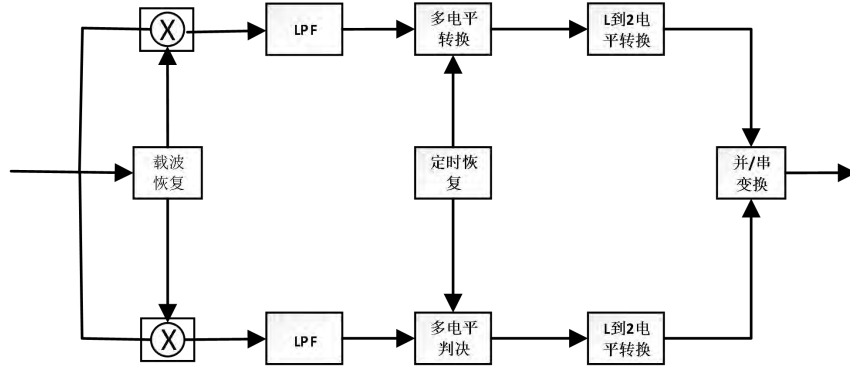


图 4.2 解调器原理图

对于 MQAM 其相干解调输出，其同相与正交支路“相关器”输出的信号加噪声混合值为：

$$\begin{cases} x_1 = \int_0^T x(t)\varphi_1(t)dt = \sqrt{E_0}a_i + n_{11} \\ x_2 = \int_0^T x(t)\varphi_2(t)dt = \sqrt{E_0}b_i + n_{12} \end{cases}$$

由此计算得符号错误概率为：

$$p_e = 1 - P_c = 1 - (1 - P_{e1})^2 = 2P_{e1} - P_{e1}^2 \approx \frac{2(L-1)}{L} \text{erfc}\left(\sqrt{\frac{E_0}{n_0}}\right)$$

$$\text{erfi}(x) = 1 - \frac{2}{\sqrt{\pi} * \int_0^x e^{-t^2}}$$

MQAM 的平均能量为：

$$E_{av} = 2 \left[ \frac{2E_0}{L} \sum_{i=1}^{\frac{L}{2}} (2i-1)^{(2)} \right] = \frac{2(M-1)}{3} E_0$$

以平均能量表示平均误符号概率：

$$P_e \approx 2 \left( \frac{L-1}{L} \right) \text{erfc} \left[ \sqrt{\frac{3E_{av}}{2(M-1)n_0}} \right] (L = \sqrt{M})$$

同时，为了传输和检测方便，同相和正交支路的 L 进制码元一般为双极性码元，其间隔相同。当 L 为偶数时，L 个信号电平取为  $\pm 1, \pm 3, \pm(L-1)$ ，所以对于本文 L=2, 其信号电平取  $\pm 1$ ，如果  $M=L*2$  为 2 的偶数次方，则方型星座的 MQAM 信号可等效为同相和正交支路的 L 进制抑制载波的 ASK 信号之和，所以 MQAM 信号是可看作由同相和正交支路的  $\sqrt{M}$  进制的 ASK 信号叠加而成，故它



的功率谱是两支路信号功率谱的叠加。

### 4.3 QPSK 与 4QAM 的有效性和可靠性对比

MQAM 信号表示式可表示为：

$$S_{QAM}(t) = \sqrt{E_0} a_i \phi_1(t) + \sqrt{E_0} b_i \phi_2(t) = \sqrt{\frac{2E_0}{T}} a_i \cos w_0 t + \sqrt{\frac{2E_0}{T}} b_i \sin w_0 t$$

其中  $0 \leq t \leq T, i=1,2,\dots,M$ , 设  $L = \sqrt{M}$

QAM (M=4) 的具体表示式为：

$$S_{QAM}(t) = \begin{bmatrix} s_1(t) = -A_0 \cos w_0 t - A_0 \sin w_0 t = \sqrt{2} A_0 \cos(w_0 t - \frac{3\pi}{4}) \\ s_2(t) = -A_0 \cos w_0 t + A_0 \sin w_0 t = \sqrt{2} A_0 \cos(w_0 t + \frac{3\pi}{4}) \\ s_3(t) = A_0 \cos w_0 t + A_0 \sin w_0 t = \sqrt{2} A_0 \cos(w_0 t + \frac{\pi}{4}) \\ s_4(t) = A_0 \cos w_0 t - A_0 \sin w_0 t = \sqrt{2} A_0 \cos(w_0 t - \frac{\pi}{4}) \end{bmatrix}$$

对于 QPSK, 其表达形式为：

$$S_{QPSK} = A_0 \cos \left[ w_0 t + (2i-1) \frac{\pi}{4} \right]$$

其中  $i=1,2,3,4$ ,  $0 \leq t \leq T$ , 将其展开得：

$$S_{QPSK} = \begin{cases} s_1(t) = A_0 \cos(w_0 t - \frac{3\pi}{4}) \\ s_2(t) = A_0 \cos(w_0 t + \frac{3\pi}{4}) \\ s_3(t) = A_0 \cos(w_0 t + \frac{\pi}{4}) \\ s_4(t) = A_0 \cos(w_0 t - \frac{\pi}{4}) \end{cases}$$

由上两种调制方式表达式可知 QPSA 和 4QAM 只有  $\sqrt{2}$  的差距, 4 个相位也完全对应相等分别表示一个双比特符号。因此 4QAM 就完全等效于 QPSK。所以 4QAM 调制和 4PSK 调制的有效性和可靠性相同。

$$P = P_{eQPSK} = \frac{1}{2} \operatorname{erfc} \left[ \sqrt{\frac{E}{n_0}} \sin \frac{\pi}{4} \right] = \frac{1}{2} \operatorname{erfc} \left[ \sqrt{\frac{E}{2n_0}} \right]$$

已知 PSK 信号数字表示为:

$$S_{PSK}(t) = \begin{cases} s_1(t) = A_0 \cos(w_0 t) \\ s_2(t) = -s_1(t) - A_0 \cos(w_0 t) \end{cases}$$

同理可得系统的误码率为:

$$P_{eQPSK} = \frac{1}{2} \operatorname{erfc} \left[ \sqrt{\frac{E}{n_0}} \sin \frac{\pi}{4} \right] = \frac{1}{2} \operatorname{erfc} \left[ \sqrt{\frac{E}{2n_0}} \right]$$

## 4.5 模型建立

对于线性噪声功率:

$$P_n = 2 * \pi * h * f * B * NF$$

由于光功率每过 15km 衰减一半, 所以我们设计增益函数如下:

$$Gain = \frac{1}{loss}$$

其衰减 loss 设计为:

$$loss = 10^{\frac{3 * \frac{L}{15000}}{10}}$$

则加上增益函数的线性噪声为:

$$P_n = 2 * \pi * h * f * B * (NF - \frac{1}{Gain})$$

非线性噪声功率为:

$$P_{n_{nl}} = \frac{2}{3} * (\frac{1}{N} \sum_{k=1}^N In^2)^2 * P_n$$

我们生成的入跨噪声信号是在复数平面下符合正态分布的 N\*N 的矩阵, 对于非线性噪声信号:

$$NL_{noise} = \frac{S_{NL}}{\sqrt{2 * P_{n_{nl}}}}$$

此时输入信号 In 变为:

$$In = In + NL_{noise}$$

对输入信号进行衰减和放大:

$$Out = \frac{In}{loss * Gain}$$

同理可得放大器的噪声:

$$LNA_{noise} = \frac{S_{NA}}{\sqrt{2 * P_N}}$$

由此可得输出为:

$$Out = Out + LNA_{noise}$$

我们已经得到加了两种噪声后的输入,将其送入 QPSK,8QAM,16QAM 中编码,得到当跨长分别为 100 时的最远距离。

## 4.6 实验结果及分析

对于问题一的第一问,画出的 BER 和 SNR 的关系曲线图如下:

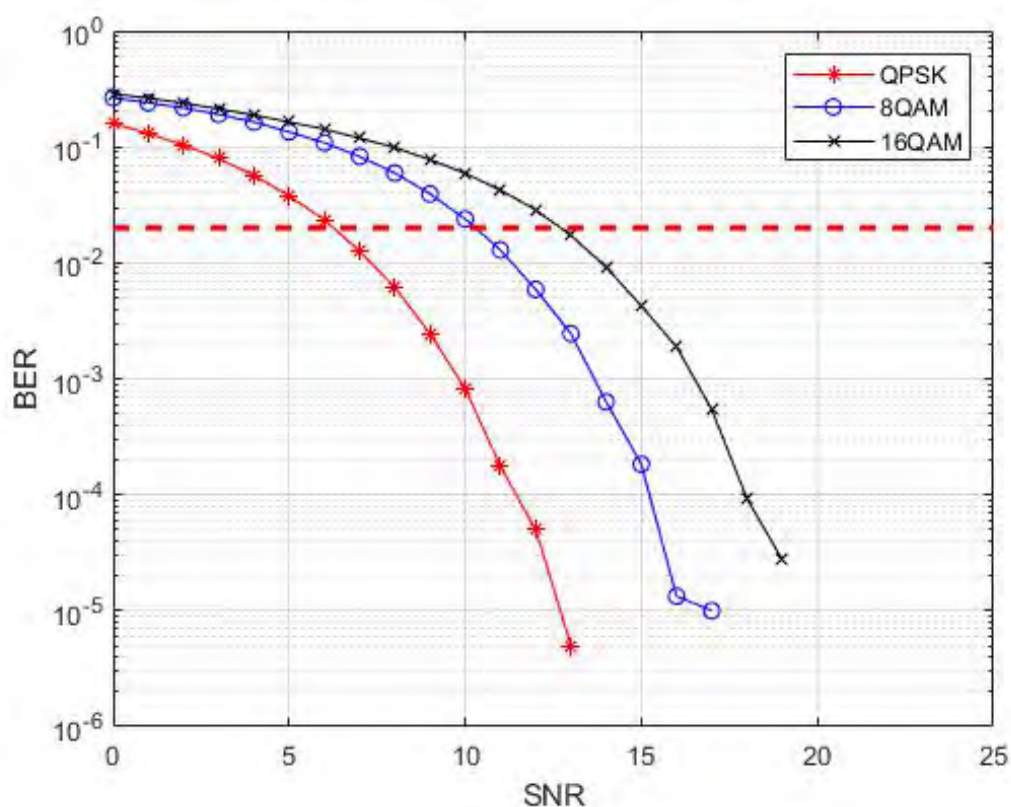


图 4.3 SNR-BER 关系曲线图

当 BER=0.02 时, SNR 容限点的取值如表 4.1 所示:

表 4.1 SNR 容限点

BER=0.02	QPSK	8QAM	16QAM
	4.5	10.5	18.5

对于第一题第二问，当单跨距离为 80km 时，三种传输格式最远传输段数曲线图如下：

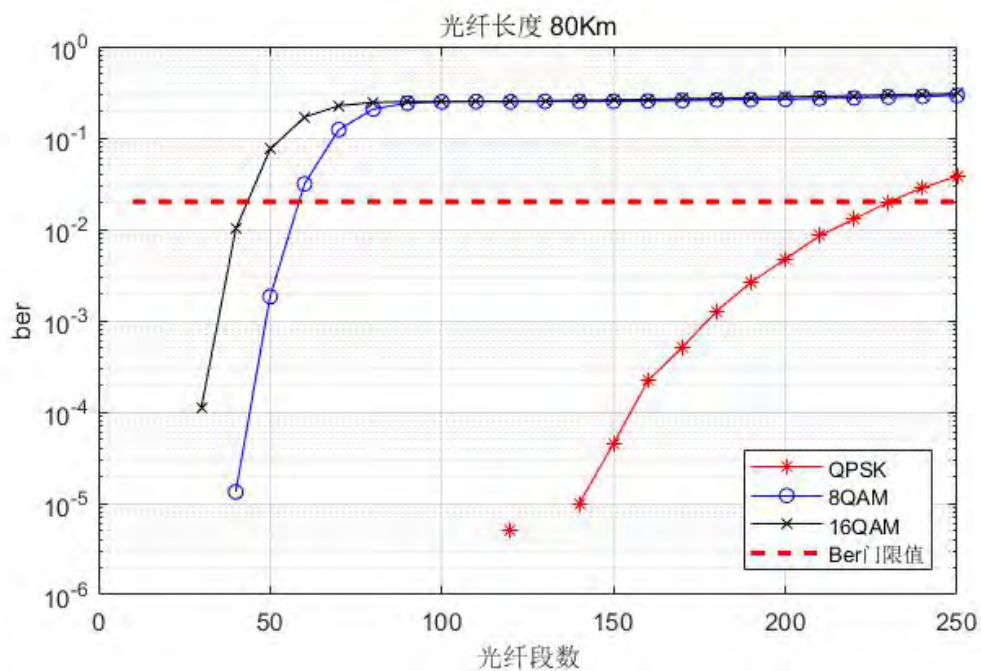


图 4.4 段数-BER 关系图

当单跨距离为 100km 时，三种传输格式最远传输段数曲线图如下：

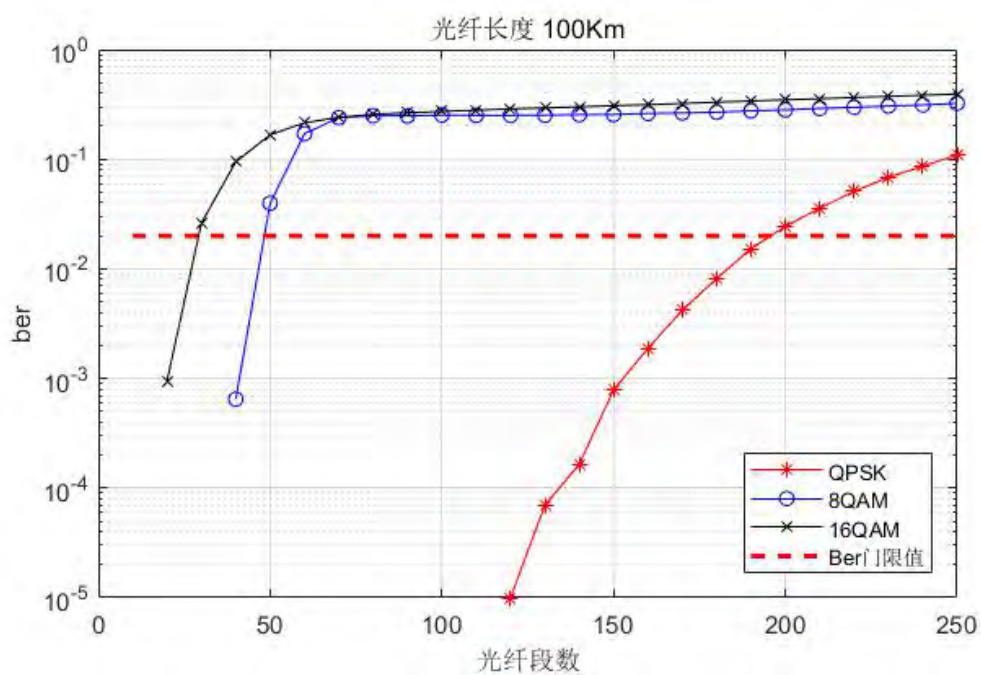


图 4.5 段数-BER 关系图

以纠前误码率 0.02 为门限，问题中给出的传输格式最远的传输距离（每跨距离×跨段数量）如表 4.2 所示：

表 4.2 三种传输格式最远距离

最远传 输距离 跨长	传输格式	QPSK	8QAM	16QAM
80km		18480km	4720km	3520km
100km		19700km	4900km	3000km

对比跨长 80km 和 100km，在 QPSK 和 8QAM 两种传输格式下，跨长为 100km 最远传输距离比 80km 的长，而在 16QAM 传输格式下，跨长为 80km 最远传输距离比 100km 的长。

## 5.问题二的模型建立与求解

### 5.1 子问题 1 的分析与解答

#### 5.1.1 问题分析

根据子问题 1 的题目要求，要为图中所示的 12 个城市进行光传输网络的规划。不考虑中间节点的连接，也就是说两个城市之间均可以采用直接连接的方式进行网络传输。那么根据图论中的有关知识，将 12 个城市看做 12 个图节点，把他们之间的传输线看做边，那么由 12 个节点构成的无向完全图共有  $12 \times (12-1) / 2 = 66$  条边，也即传输网最多可以有 66 条传输线路。题中又定义了每个连接的价值定义为传输的容量与连接区域人口数的乘积（取两区域人口数乘积的 0.5 次方）以及网络的价值：所有连接价值的加权和。并且规定了两种传送网规划的线路条数分别为 16 条和 33 条，需要设计两种规划方案使得传送网的网络价值分别达到最大值。

所以该问题可以转化成一种最优化问题，在一个 66 条边的完全无向图中找到一个有 16 条边的子图能够连接到所有节点的解决方案和一个具有 33 条边的子

图能够连接到所有节点的解决方案。题目中说明，每条连接的价值由传输的容量和两个连接点（城市）间的人口乘积（0.5 次方）的乘积决定的，也就是说每条边的价值都已经确定下来了，城市间的距离越短网络容量就越高，城市间的人口越多网络价值量也越大。所以为了简化问题的处理，将每一条边上的权值设置为 1，相当于由边的价值决定了边的权重，价值高的权重也就大，在计算网络的总价值时也是合理适用的。

对于这样的一种找出最优路径求解最大值的问题。前面已经讨论过，城市间距离越短，所用的光纤容量越大，可以得到较高的连接价值。所以我们考虑使用遗传算法求解 TSP 问题来模拟解决得到这样一条最短的路径，可是这样解出来的也只是连接了 12 个城市节点的一条回路，题目中要求的连接数分别为 16 条和 33 条。所以我们又加上了图的广度优先搜索算法，在迭代完成后对得到的图进行广度优先搜索，得到剩下的连接价值最高的边连接起来，最后就得到了网络价值最高的光传输方案。

表 5.1 是查阅到的 12 个城市之间的球面距离和每条边对应的连接价值。我们城市距离超过 3000km 的距离定义为 INF，其单波传送和容量都定义为 0。

表 5.1 城市信息表

城市	城市	距离 (km)	单波传 输容量 (Gb/s)	人口 m (sqrt)	容量 (Tb/s )	人口*容量 = 连接价值 (mTb/s)
成都	重庆	265	400	18.3226635	32	586.325234
西安	重庆	577	400	17.1953482	32	550.2511427
上海	北京	1068	200	29.1862982	16	466.9807705
重庆	深圳	1081	200	28.8374756	16	461.3996099
郑州	武汉	466	400	12.5857062	32	402.7425977
北京	武汉	1053	200	23.7318352	16	379.7093625
重庆	武汉	754	200	22.1990991	16	355.1855853
武汉	深圳	895	200	20.7846097	16	332.5537551
上海	武汉	688	200	19.6773982	16	314.8383712
北京	哈尔滨	1053	200	19.5877513	16	313.4040204
西安	郑州	431	400	9.74884609	32	311.9630747
北京	郑州	624	200	18.6676191	16	298.6819044
北京	西安	907	200	18.3826005	16	294.1216075
郑州	重庆	881	200	17.4619587	16	279.3913384
北京	重庆	1460	100	32.926585	8	263.41268
上海	郑州	830	200	15.478372	16	247.6539521
北京	深圳	1944	100	30.8285582	8	246.6284655

重庆	昆明	637	200	14.3652358	16	229.843773
上海	重庆	1443	100	27.301282	8	218.4102562
昆明	深圳	1171	200	13.4499071	16	215.198513
成都	武汉	980	200	13.2060592	16	211.2969474
上海	深圳	1213	100	25.5616901	8	204.4935207
西安	武汉	653	200	12.3935467	16	198.2967473
郑州	成都	1008	200	10.3879738	16	166.2075811
西安	成都	620	200	10.2293695	16	163.6699117
北京	成都	1526	100	19.5877513	8	156.7020102
哈尔滨	重庆	2509	100	18.3226636	8	146.5813085
哈尔滨	深圳	2824	100	17.1551742	8	137.2413932
成都	深圳	1335	100	17.1551742	8	137.2413932
成都	昆明	645	200	8.54575918	16	136.7321469
郑州	深圳	1358	100	16.3493119	8	130.7944953
上海	哈尔滨	1680	100	16.2413054	8	129.9304429
上海	成都	1662	100	16.2413054	8	129.9304429
西安	深圳	1403	100	16.0996894	8	128.7975155
北京	昆明	2098	100	15.3570831	8	122.8566645
上海	西安	1222	100	15.2420471	8	121.9363769
哈尔滨	武汉	1997	100	13.2060592	8	105.6484737
上海	昆明	1959	100	12.7334206	8	101.8673647
哈尔滨	成都	2579	100	10.9	8	87.2
哈尔滨	郑州	1644	100	10.3879738	8	83.10379053
武汉	昆明	1295	100	10.3537433	8	82.82994628
哈尔滨	西安	1960	100	10.2293695	8	81.83495586
北京	乌鲁木齐	2413	100	8.8	8	70.4
乌鲁木齐	重庆	2308	100	8.23164625	8	65.85317001
郑州	昆明	1511	100	8.14432318	8	65.15458541
西安	昆明	1206	100	8.01997506	8	64.1598005
乌鲁木齐	武汉	2767	100	5.93295879	8	47.46367032
北京	拉萨	2567	100	5.62849891	8	45.02799129
拉萨	重庆	1493	100	5.26497863	8	42.11982906
拉萨	深圳	2419	100	4.92950309	8	39.43602414
乌鲁木齐	成都	2065	100	4.8969378	8	39.17550255
上海	拉萨	2909	100	4.66690476	8	37.33523805
乌鲁木齐	郑州	2443	100	4.66690476	8	37.33523805
乌鲁木齐	西安	2113	100	4.59565012	8	36.76520094
乌鲁木齐	昆明	2520	100	3.83927076	8	30.71416611
拉萨	武汉	2230	100	3.79473319	8	30.35786554
拉萨	成都	1250	100	3.13209195	8	25.05673562
拉萨	郑州	2189	100	2.98496231	8	23.87969849
拉萨	西安	1758	100	2.93938769	8	23.51510153
拉萨	昆明	1273	100	2.45560583	8	19.64484665
乌鲁木齐	拉萨	1607	100	1.40712473	8	11.25699782

上海	乌鲁木齐	INF	0	7.29657454	0	0
哈尔滨	乌鲁木齐	INF	0	4.89693782	0	0
哈尔滨	拉萨	INF	0	3.13209195	0	0
哈尔滨	昆明	INF	0	8.54575918	0	0
乌鲁木齐	深圳	INF	0	7.70713955	0	0

表 5.2 是 12 个城市人口数量和对应的省份的人口数量，统计于 2017 年的城市常住人口。

表 5.2（空缺部分为直辖市，在省份人口计算时按照直辖市人口计算）

城市	人口/m	省份	人口/m
北京市+天津市	35.2757		
哈尔滨市	10.929	黑龙江省	37.887
乌鲁木齐市	2.2261	新疆	22.9847
拉萨市	0.9025	西藏	3.3054
西安市	9.6167	陕西省	38.3544
郑州市	9.881	河南省	95.5913
上海市	24.1833		
武汉市	10.914	湖北省	59.02
重庆市	30.7516		
成都市	16.0447	四川省	83.02
昆明市	6.728	云南省	47.705
广州市+深圳市	27.0267	广东省	111.69

### 5.1.2 算法简介

旅行商问题，即 TSP 问题（Traveling Salesman Problem）又译为旅行推销员问题、货郎担问题，是数学领域中著名问题之一。假设有一个旅行商人要拜访  $n$  个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。

遗传算法是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，能自动获取和指导优化的搜索空间，自适应地调整搜索方向，不需要确定的规则。遗传算法的这些性质，已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。它是现代有关智能计算中的关键技术。对于一个求函数最大（或最小）值的优化问题，一般可以描述为下列数学



规划模型：式中  $x$  为决策变量， $\max f(x)$  为目标函数式，式  $x \in R$ 、 $R \in U$  为约束条件， $U$  是基本空间， $R$  是  $U$  的子集。满足约束条件的解  $X$  称为可行解，集合  $R$  表示所有满足约束条件的解所组成的集合，称为可行解集合。包含选择、交叉、变异、适应度计算、个体评价等操作。

遗传算法是解决 TSP 问题的一个有效方法。

### 5.1.3 建立模型

根据遗传算法的基本步骤，假设种群内的一个个体为 1~66 条边构成的一个列向量，向量中的编号排序用 `randperm()` 函数进行随机打乱以模拟不同的个体差异，设定最开始有 50 个个体，便建立了一个规模  $inn=50$  的初代种群。再设置最大的迭代次数  $gnmax=1500$  次（这个次数是根据输出的效果不断地调整最后得到的），交叉概率  $pc=0.8$ ，变异概率  $pm=0.8$ 。第一步：进行当前种群个体的适应度评价，若适应度不够好，进行第二步。若适应度已经达到要求，则输出当前种群中适应度最高的个体；第二步：根据选择函数 `sel()` 随机选择当前种群 50 个个体中的 2 个个体，若选到相同个体则重新随机选择一次；第三步：根据交叉概率  $pc$  判断是否执行交叉操作，若执行交叉，则随机选择 2 个个体相同位置上的值进行交换，随后进行第四步。若不进行交叉，则进入第四步；第四步：根据变异概率  $pm$  判断是否执行变异操作，若执行变异，则在 1~66 个位置上随机产生一个变异位，到此，生成新的下一代种群，在迭代次数内的话回到第一步计算新种群个体的适应度。若不执行变异，生成新的下一代种群在迭代次数内回到第一步计算新种群个体适应度。模型的流程图如图 5.1 所示。

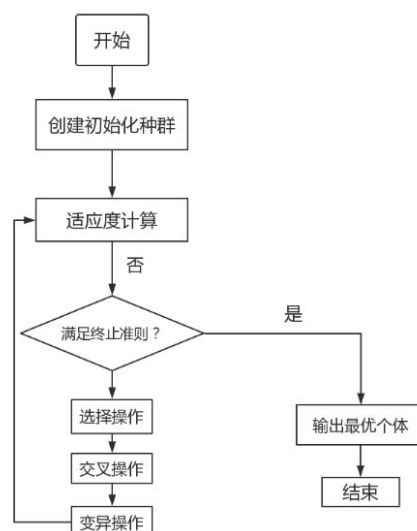


图 5.1 模型的基本流程

### 5.1.4 结果分析

在 Matlab 中构建完成模型以后，分别输入目标网络连接数为 16 和 33，运行会得到如下图 5.2 和图 5.3 的输出结果。

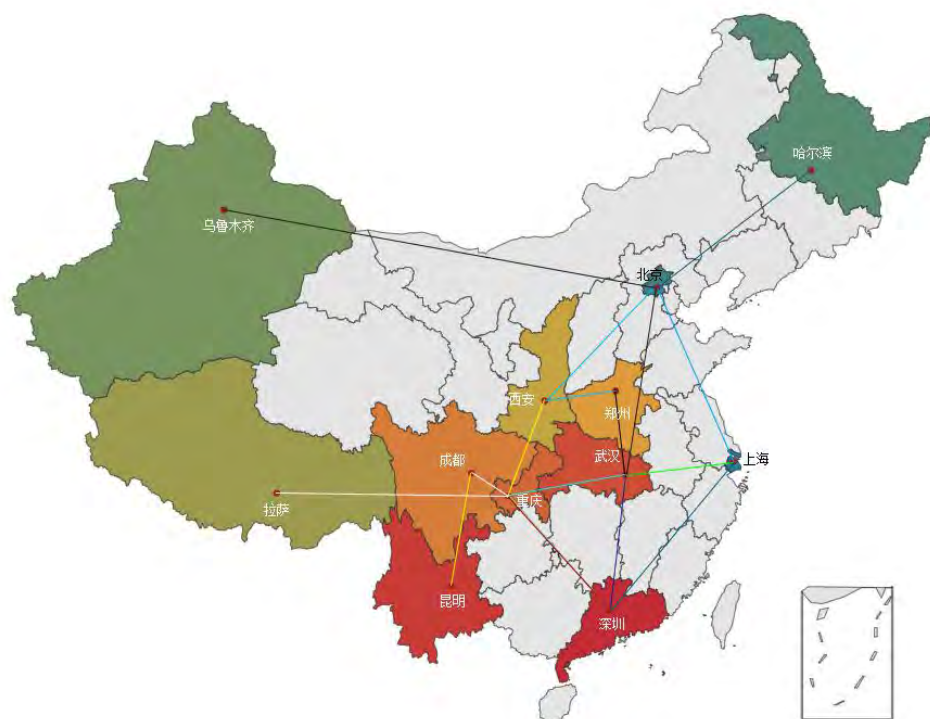
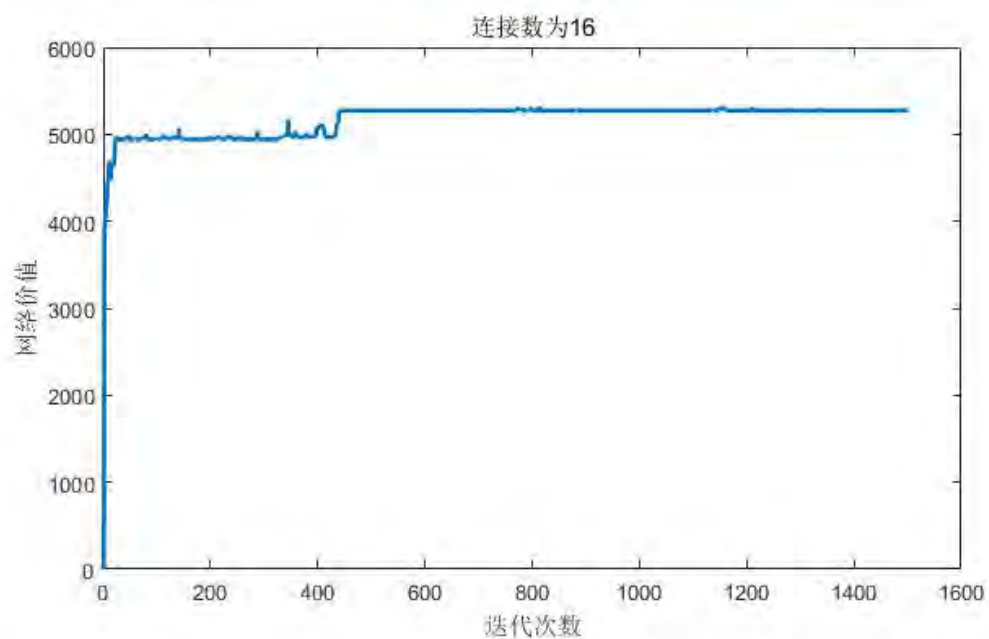


图 5.2 连接数为 16 的光传输网络

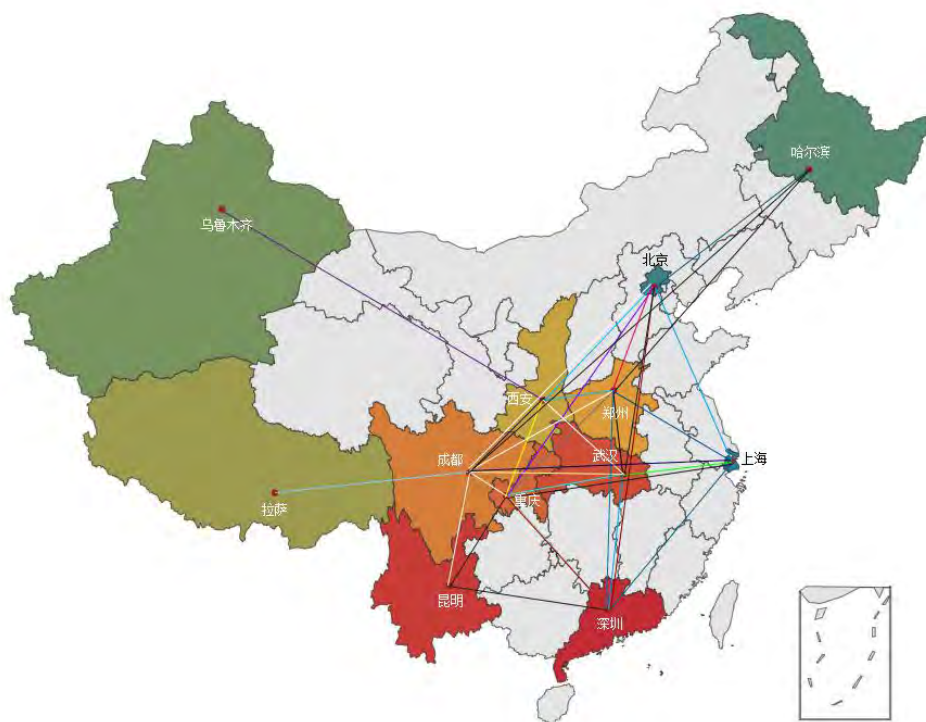
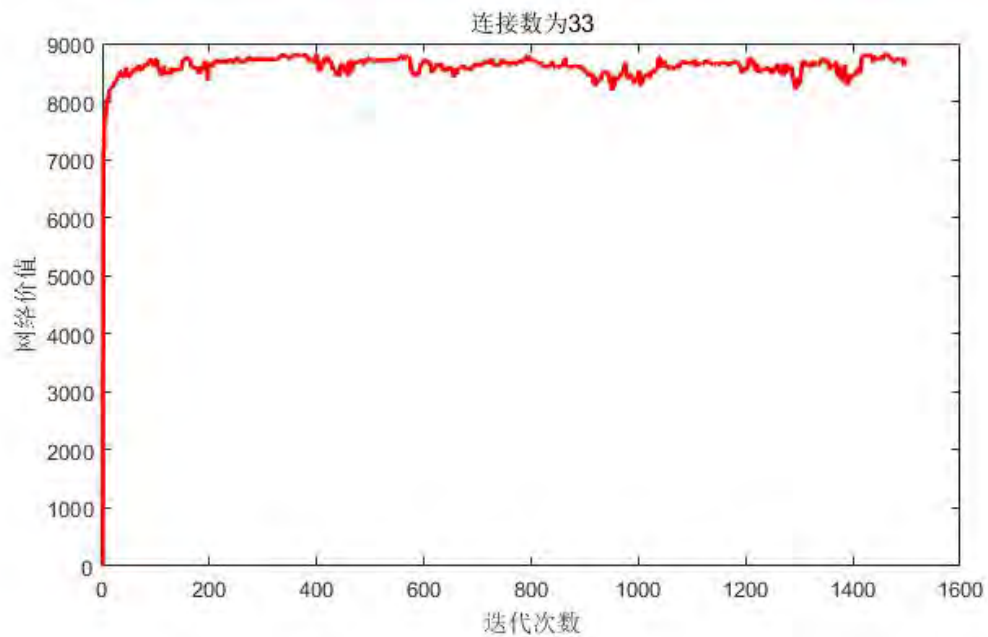


图 5.3 连接数为 33 的光传输网络

从上图的结果我们得知，当迭代次数逐渐增加时连接数为 16 的光传输网络的网路价值逐渐升高并趋于稳定，在迭代次数达到 450 左右时上升至 5300mTb/s 左右，继续增加迭代次数，网络价值总体保持稳定趋势。连接数为 33 的网络在

迭代几十次的时候就已经达到了它的收敛域 8500mTb/s 左右,并且在迭代次数不断增加到 1500 次时也是仅仅在范围内波动,总体上来看认为它是趋于稳定收敛的。所以,根据我们建立的模型,对于子问题一的 16 条连接的光传输网络,它的最大网络价值为 5272.9mTb/s。对于子问题的 33 条连接的光传输网络,它的最大网络价值为 8670.4mTb/s。

## 5.2 子问题 2 的分析与解答

### 5.2.1 问题分析

相较于子问题 1,子问题 2 多了一个较为重大的改变。不同城市直接的网络传输可以不经过直接连接的线路进行传送,而是可以通过中间城市当作中转进行传输,这样一来就会引发光纤容量分配的问题,这也是该问题中比较难以处理的地方。题目假设每条链路容量可任意分配,只要总容量不超过表中的规定即可。由于题目定义的网络价值的表达式没有变化,还是可以近似看成要使得网络价值最大,两个城市之间的节点的距离要尽可能的小,这样对应的链路容量就会大。由于这里还存在着链路容量分配的问题,所以分配出去的容量也要尽可能的小,这样才能保证乘积得到的网络价值尽可能的大。

所以,在子问题 1 的基础上,考虑在每一次对种群中的个体求解个体适应度即当前个体表达的解的网络价值的时候,采用 Dijkstra 算法,对图中的每个顶点开始求解它连接到其他节点的最短路径,并且对每个节点的容量进行分配计算,计算每个边的新的链路价值,最后把每个链路和中间链路的链路价值相加得到该个体代表的网络价值,作为该个体的适应度。在种群所有的个体中选择适应度最高的记录下来。

对于该问题的第二个小问题,考虑把所有的节点的城市人口换成对应的省的总人口,再次运行程序得到两个网络的传送分布图,对比之前的城市人口,进行分析。

### 5.2.2 算法简介

迪杰斯特拉算法(Dijkstra)是由荷兰计算机科学家狄克斯特拉于 1959 年提出的,因此又叫狄克斯特拉算法。是从一个顶点到其余各顶点的最短路径算法,解决的是有向图中最短路径问题。迪杰斯特拉算法主要特点是以起始点为中心向外层层扩展,直到扩展到终点为止。基本的算法思想如下:

按路径长度递增次序产生算法：

把顶点集合  $V$  分成两组：

(1)  $S$ ：已求出的顶点的集合。（初始时只含有源点  $V_0$ ）

(2)  $V-S=T$ ：尚未确定的顶点集合。

将  $T$  中顶点按递增的次序加入到  $S$  中，保证：

(1) 从源点  $V_0$  到  $S$  中其他各顶点的长度都不大于从  $V_0$  到  $T$  中任何顶点的最短路径长度。

(2) 每个顶点对应一个距离值。

$S$  中顶点：从  $V_0$  到此顶点的长度。

$T$  中顶点：从  $V_0$  到此顶点的只包括  $S$  中顶点作中间顶点的最短路径长度。

依据：可以证明  $V_0$  到  $T$  中顶点  $V_k$  的，或是从  $V_0$  到  $V_k$  的直接路径的权值；或是从  $V_0$  经  $S$  中顶点到  $V_k$  的路径权值之和。（反证法可证）

求最短路径步骤，算法步骤如下：

$G=\{V,E\}$

1. 初始时令  $S=\{V_0\}, T=V-S=\{\text{其余顶点}\}$ ， $T$  中顶点对应的距离值。

若存在  $\langle V_0, V_i \rangle$ ， $d(V_0, V_i)$  为  $\langle V_0, V_i \rangle$  弧上的权值。

若不存在  $\langle V_0, V_i \rangle$ ， $d(V_0, V_i)$  为  $INF$ 。

2. 从  $T$  中选取一个与  $S$  中顶点有关联边且权值最小的顶点  $W$ ，加入到  $S$  中。

3. 对其余  $T$  中顶点的距离值进行修改：若加进  $W$  作中间顶点，从  $V_0$  到  $V_i$  的距离值缩短，则修改此距离值。

重复上述步骤 2、3，直到  $S$  中包含所有顶点，即  $W=V_i$  为止。

### 5.2.3 建立模型

由于整体模型还是用的解决第一问的模型，只是在适应度函数中加入了 Dijkstra 算法去计算分配出来链路的容量，所以整体的模型流程如图 5.3 所示。

### 5.2.3 结果分析

图 5.4 所示的是连接数为 16 的情况下的光传输网络。图 5.5 所示的是连接数为 33 的情况下的光传输网络。

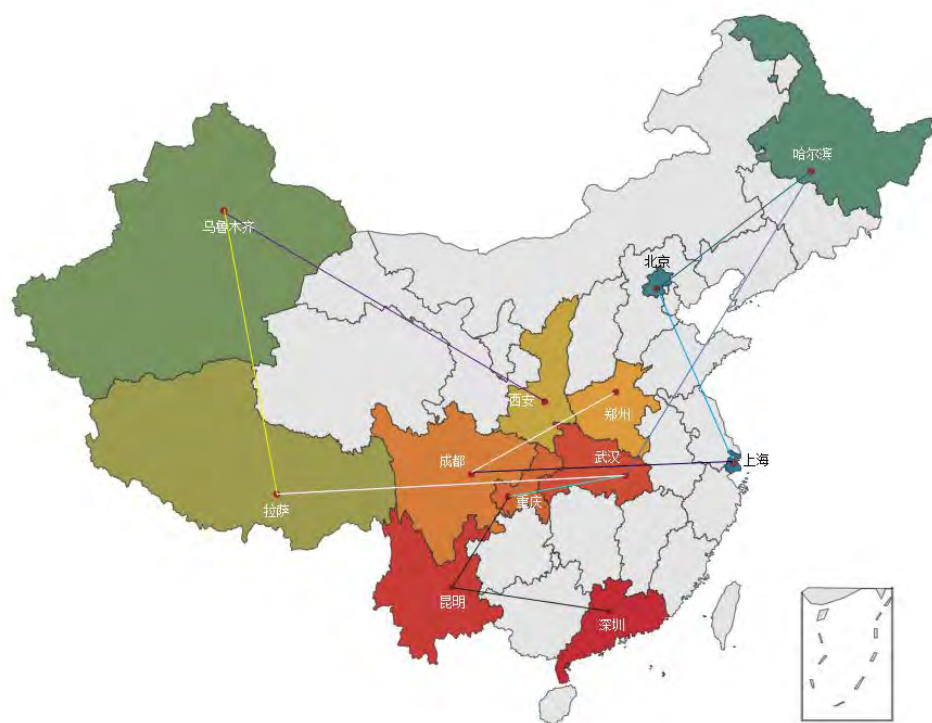
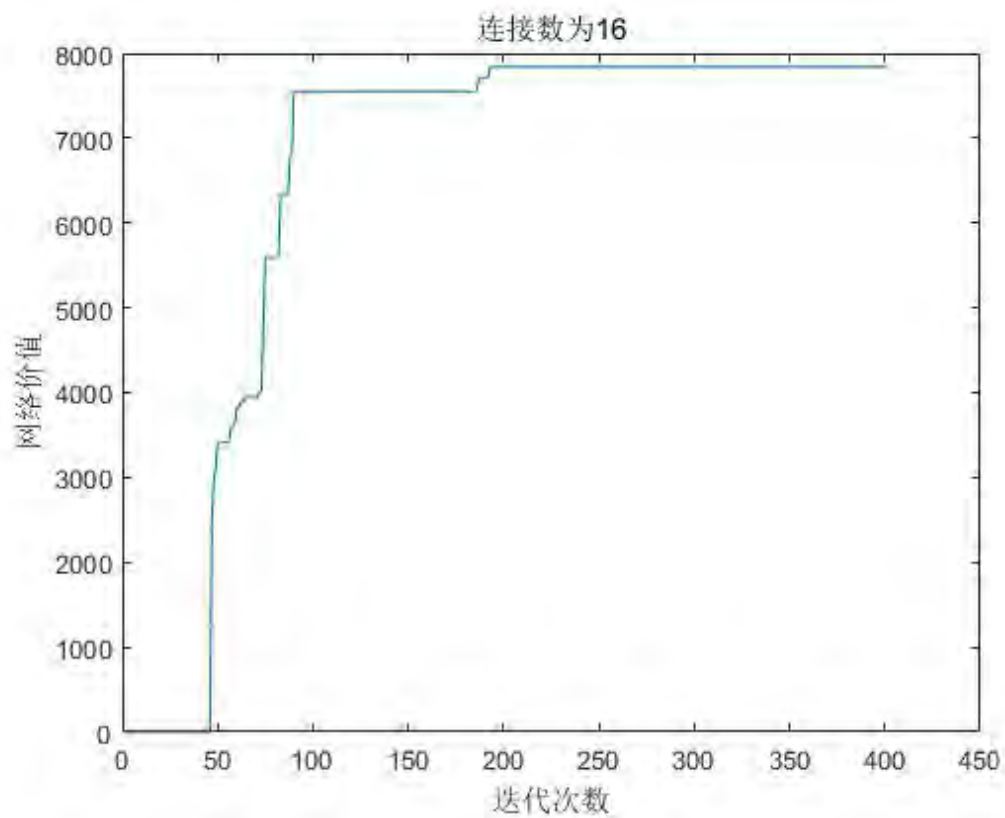


图 5.4 连接数为 16 的光传输网络

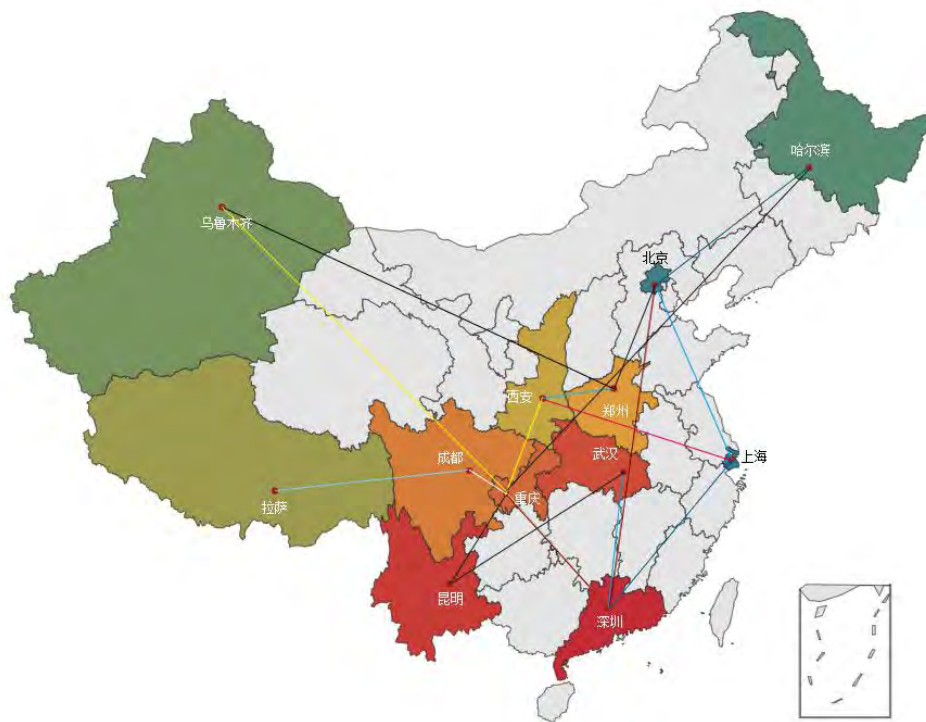
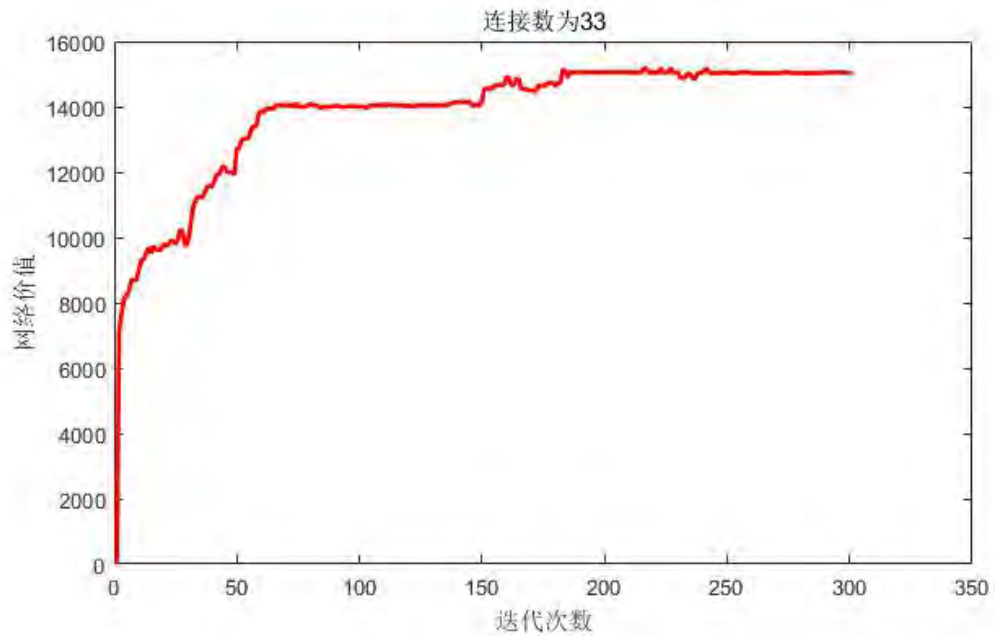


图 5.5 连接数为 33 的光传输网络

连接数为 16 的光传输网络的网路价值为 7839.9mTb/s，连接数为 33 的 6 的光传输网络的网路价值为 15027.3mTb/s。可以看出，相比较于之前的直接建立在城市两点之间的网络，网络价值有了很大的提升。

根据表 5.2 中的城市对应的省份人口数量，将模型中的节点中记录的城市人



口更新成为省份人口，重新运行程序得到新的两种光传输网络规划图。图 5.6 所示的是连接数为 16 的情况下的光传输网络。图 5.7 所示的是连接数为 33 的情况下的光传输网络。

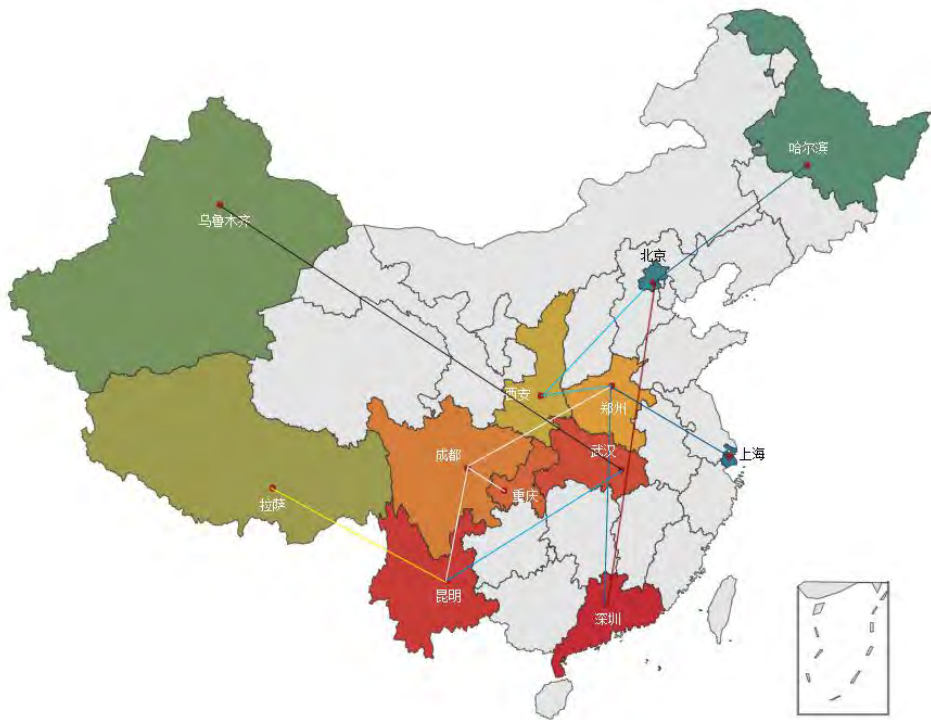
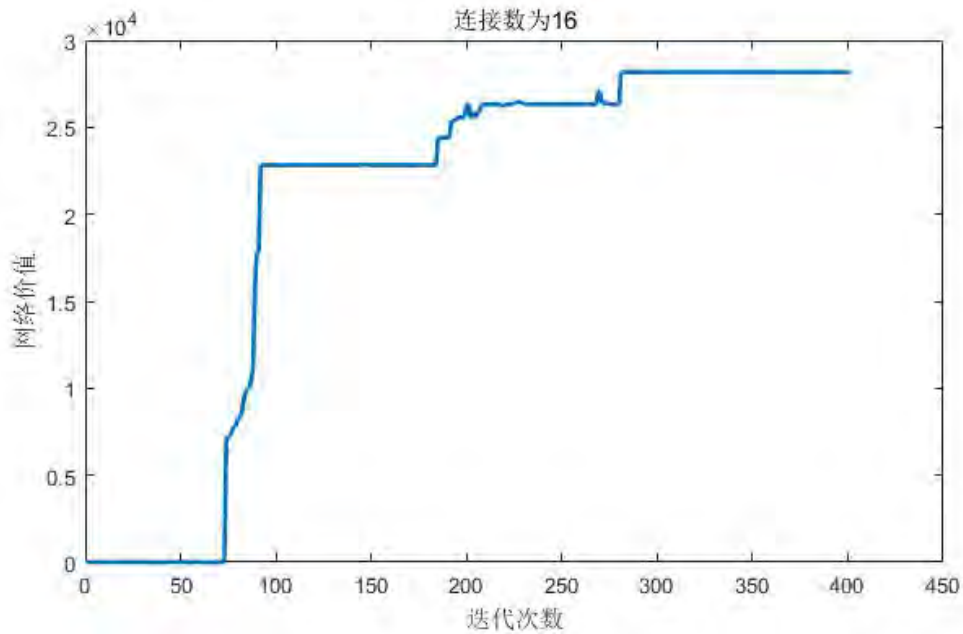


图 5.6 连接数为 16 的光传输网络



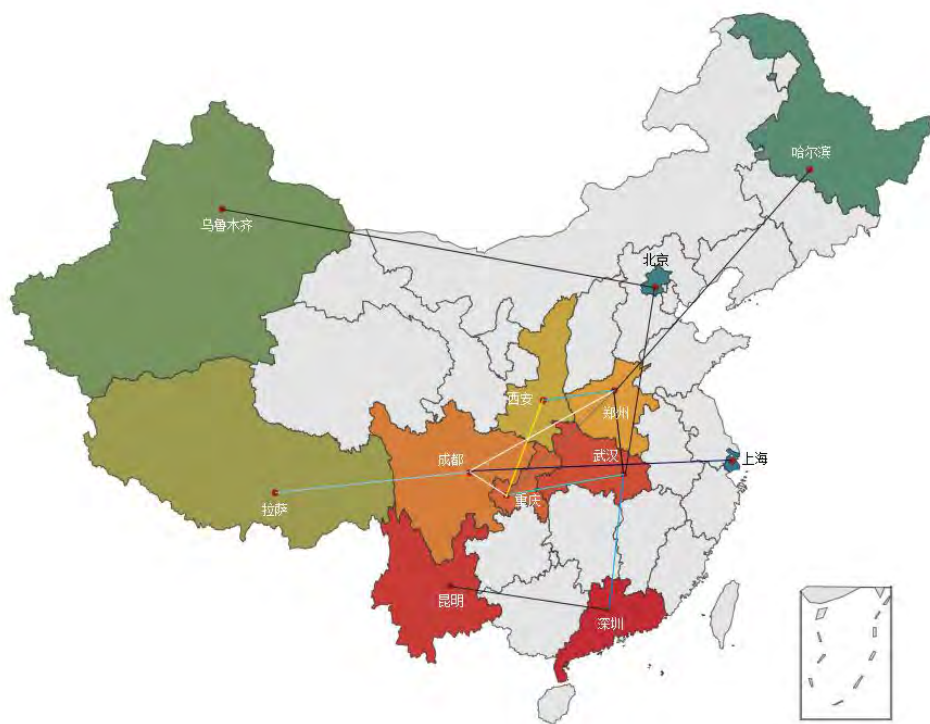
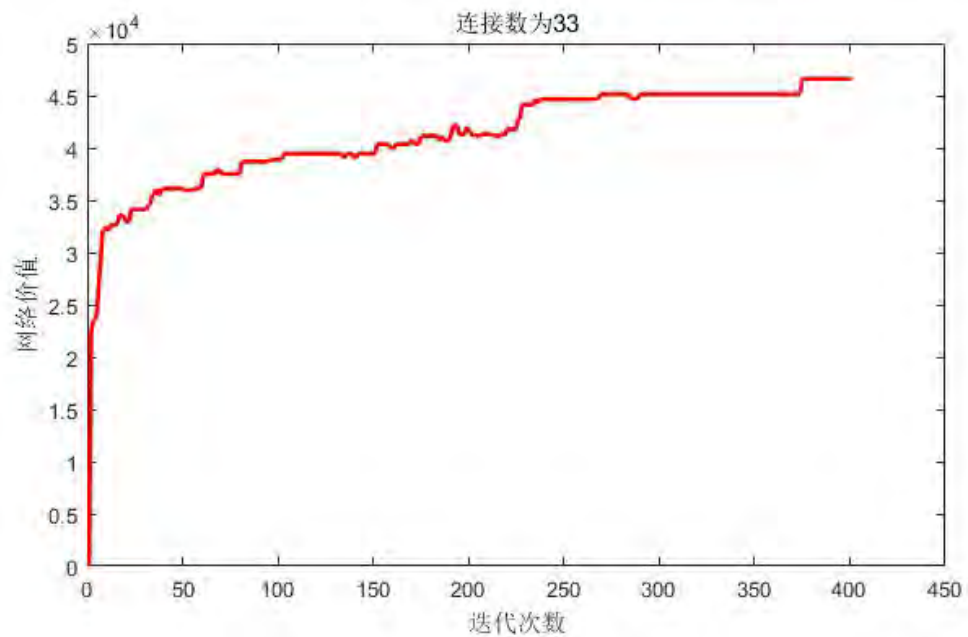


图 5.7 连接数为 33 的光传输网络

连接数为 16 的光传输网络的网络价值为 28160.6mTb/s，连接数为 33 的 6 的光传输网络的网络价值为 46603.3mTb/s。网络价值增加是必然的，因为图中节点代表的人口数量已经明显增加了。

根据图 5.4 和图 5.6 以及图 5.5 和图 5.7 的对比,发现把城市节点的人口更换成省份人口之后,原本是北京、深圳、上海、重庆等人口比其他城市多的一线城市当作中间节点的连线,明显的比用作省份人口当作节点人口容量的情况下的要少一些。说明人口的增加对于该光传传输网络的影响还是比较大的,人口增加的情况下会使得链路价值增加,有可能链路价值还会超过一些一线城市之间的链路价值。

## 5.3 子问题 3 的分析与解答

### 5.3.1 问题分析

考虑到子问题 3, 由于不同城市的经济发展状况不一致, 例如, 北京, 上海等城市的经济发展状况明显好于拉萨, 乌鲁木齐等城市, 与北上广进行连接的网络价值明显较大, 因此, 我们认为网络价值不仅与容量和人口有关, 还与 GDP(地区经济总量)有关。因此我们查找了 12 个城市 2017 的 GDP 总值。如表 6.9 所示

表 5.3 2017 国内主要城市 GDP 总值

城市名	2017 年 GDP 总值/亿元
北京市+天津市（北京）	46595
广州+深圳（深圳）	43786
上海市	30133
重庆市	19530
成都市	13890
武汉市	13400
郑州市	9003
西安市	7469
哈尔滨市	6609
昆明市	4856
乌鲁木齐	2799
拉萨	478.3

### 5.3.2 更改模型参数

我们重新定义了网络价值目标函数, 其表达式为:

$$\text{网络价值} = \sum \text{权重} * \text{容量} * \text{人口} * \text{GDP}$$

此时，我们认为当权重，容量，人口一定的时候，GDP 越大，网络价值越高，同时经济相对落后地区连接经济相对发达地区，也能促进当地经济发展。通过更改网络规划的目标函数，得到了连接数分别为 16 和 33 的网络规划图。

### 5.3.3 结果分析

通过得到下面两个新的传输网络图 5.8 和图 5.9 后我们发现，虽然网络的连接发生了比较大的变化，但是实质上没有什么改变。由于繁华城市的 GDP 和人口数量正相关，所以人口多的城市 GDP 普遍处于领先地位，乘以 GDP 之后链路价值大的线路，其价值更加被放大，该城市节点的网络链路数也会相应地增加。所以站在网络供应商的角度来看，尽可能的选择多的网络链路连接在一线城市或者是作为其他周边城市的中转节点，会获得利益的最大化

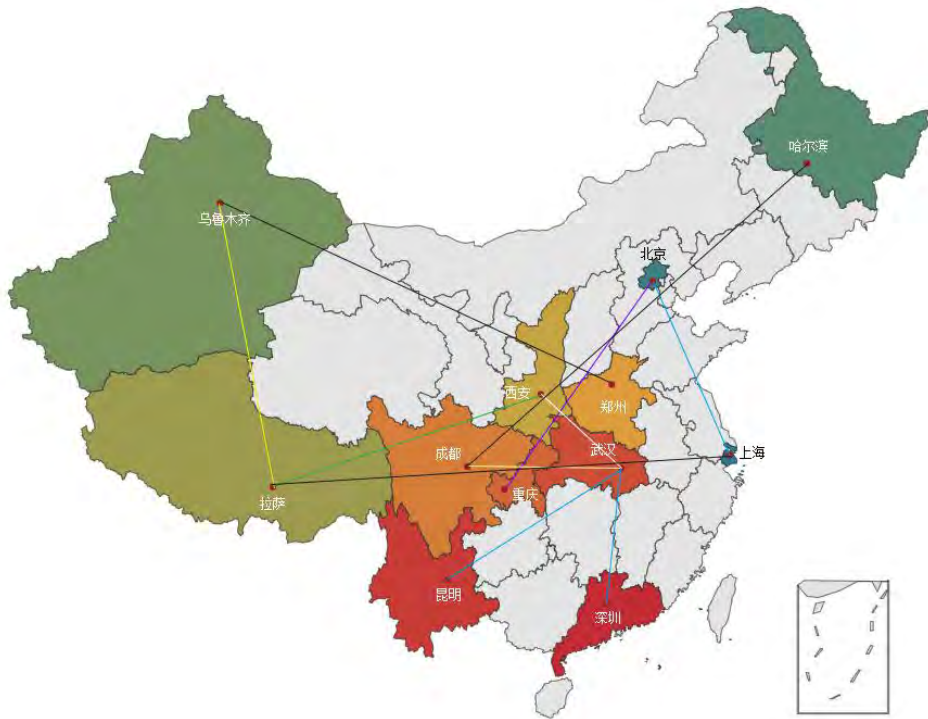


图 5.8 连接数为 16 的光传输网络

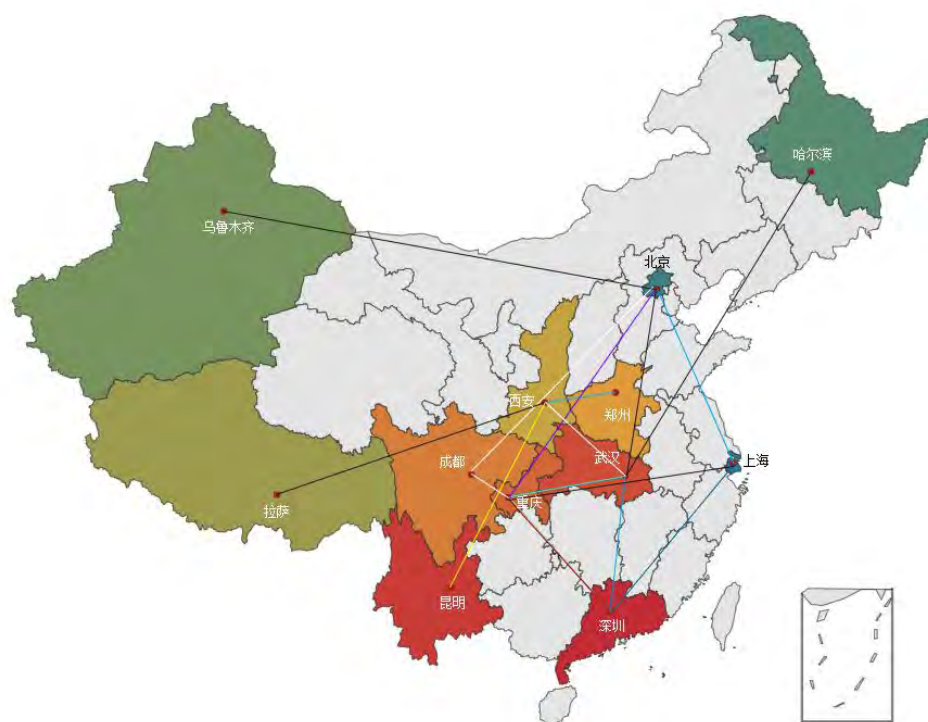


图 5.9 连接数为 33 的光传输网络

## 6.问题三模型建立与求解

### 6.1 8QAM 调制与解调原理

QAM (Quadrature Amplitude Modulation) 是一种振幅和相位联合调制键控, 频谱利用率较高的调制方式, 它用两路独立的基带数字信号对两个相互正交的同频载波的双边带调制, 并利用这种已调信号在同一带宽内频谱正交的性质来实现两路并行的数字信号传输。

正交调幅信号有两个相同频率的载波, 但是相位相差 90 度。一个信号叫 I 信号, 另一个信号叫 Q 信号。从数学角度将一个信号可以表示成正弦, 另一个表示成余弦。

两种被调制的载波在发射时已被混和。到达目的地后, 载波被分离, 数据被分别提取然后和原始调制信息相混和。

QAM 是一种矢量调制, 将输入比特先映射 (一般采用格雷码) 到一个复平

面（星座）上，形成复数调制符号，然后将符号的 I、Q 分量（对应复平面的实部和虚部，也就是水平和垂直方向）采用幅度调制，分别对应调制在相互正交（时域正交）的两个载波上。这样与幅度调制（AM）相比，其频谱利用率将提高 1 倍。QAM 是幅度、相位联合调制的技术，它同时利用了载波的幅度和相位来传递信息比特，因此在最小距离相同的条件下可实现更高的频带利用率。

## 6.2 QAM 调制原理

QAM(Quadrature Amplitude Modulation)正交幅度调制技术，是用两路独立的基带信号对两个相互正交的同频载波进行抑制载波双边带调幅，利用这种已调信号的频谱在同一带宽内的正交性，实现两路并行的数字信息的传输。该调制方式通常有 8QAM，16QAM，64QAM。QAM 调制实际上就是幅度调制和相位调制的组合，相位、幅度状态定义了一个数字或数字的组合。QAM 的优点是具有更大的符号率，从而可获得更高的系统效率。通常由符号率确定占用带宽。因此每个符号的比特（基本信息单位）越多，频带效率就越高。

调制时，将输入信息分成两部分：一部分进行幅度调制；另一部分进行相位调制。对于星型 8QAM 信号，每个码元由 3 个比特组成，可将它分成第一个比特和后两个比特两部分。前者用于改变信号矢量的振幅，后者用于差分相位调制，通过格雷编码来改变当前码元信号矢量相位与前一码元信号矢量相位之间的相位差。

QAM 是一种高效的线性调制方式，常用的是 8QAM，16QAM，64QAM 等。当随着 M 的增大，相应的误码率增高，抗干扰性能下降。

## 6.3 QAM 星座图设计

QAM 调制技术对应的空间信号矢量端点分布图称为星座图。QAM 的星座图呈现星状分层分布，同一层信号点的振幅相同，位于一个圆周上。常见的调制方式如 8QAM，16QAM，64QAM 所对应的星座图中分别有 8，16，64 个矢量端点。

## 6.4 常见星座图简介

多电平 QAM 星座图的形式主要有圆形、三角形和矩形等 3 种。其中，由于矩形星座图，易于实现、系统误码率较低，得到了广泛应用。

### (1)圆形星座图

圆形星座图的基本特征是所有星座点都处在以原点为圆心的一个或多个圆周上。实际应用中，为了提高系统性能，排列在各个圆周上的星座点应遵循以下原则。首先，各圆周上的星座点数与该圆的半径成正比关系，即圆的半径越大，圆周上的星座点数就越多，且半径与星座点数之比是一个常数，其次，同一圆周上各星座点应保持均匀分布，各星座点之间应保持一定的几何位置。

### (2)三角形星座图

三角形星座图中相邻最近 3 点的连线构成一个正三角形。这种星座图一般不在原点处安排星座点，因而，围绕原点构成一个正六边形。这样的安排使相邻的星座点之间距离相等，从而提高了系统性能。实际应用中，应尽可能将各星座点按正三角形排列的原则布置在一个圆环内，这样可以较好地利用发信功放的输出功率。

### (3)矩形星座图

矩形 QAM 星座图是以正方形作为星座图的基本形状。矩形星座图易于实现、系统误码率较低，得到了广泛应用。

## 6.5 星座图的性能评价指标

对于 M 值指定后的星座图设计，我们有一下几种指标可以用来评定星座图的性能：

(1)最小欧式距离，其值越大，系统抗干扰能力越强；

(2)误码率 BER-SNR 曲线越靠左下，系统的抗噪性能越好；

(3)峰值-均值比；以 8QAM 为例，一个已调信号可以表示成 8 个信号矢量：

$$S_i(t) = a_i \cos w_0 t - b_i \sin w_0 t, (i = 1, 2, 3, \dots, 8)$$

其中， $-\frac{T}{2} < t < T$ ， $2T$  是码元宽度。

由此定义星座图的均值信噪比和峰值信噪比分别为：（设信道为加性高斯白噪声信道， $\delta^2$  为噪声的单边功率谱密度）

$$SNR_{av} = \sum_{i=1}^8 \frac{(a_i^2 + b_i^2)}{8\delta^2}, \quad SNR_{pk} = \max_{ie(1,2,\dots,8)} \frac{(a_i^2 + b_i^2)}{\delta^2}$$

故星座图的峰值-均值比为：

$$r = \frac{SNR_{pk}}{SNR_{av}}$$

峰值-均值比小一些为好，这样可以充分利用功放的输出功率。

(4)最小相位偏移，是 MQAM 信号星座点相位的最小偏移，该参数反应了 MQAM 信号抗相位抖动能力和对时钟恢复精确度的敏感性，同样可以优化星座点的分布来获取最大值，从而获得更好的传输性能。

## 6.6 8QAM 星座图的构造

基于以上介绍，我们提出了基于 16QAM 改善的星座图结构（星座点个数为 8，星座点位置进行变换），如图 6.1 所示。当我们在星座点上添加高斯白噪声的时候，星座图变化如图 6.2 所示。

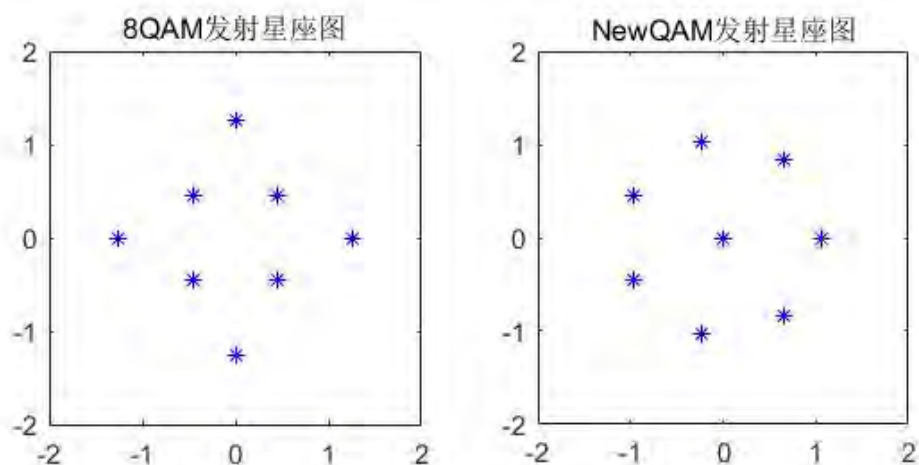


图 6.1 原始 8QAM 星座图和改善的 QAM 星座图

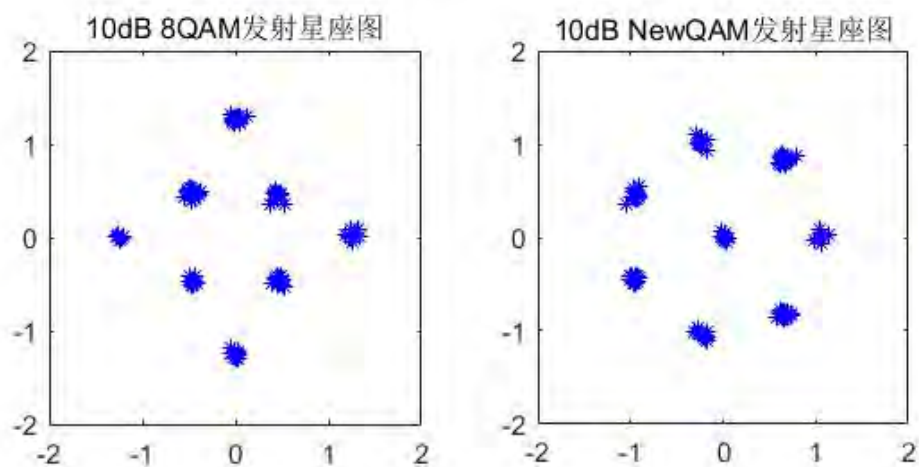


图 6.2 添加 10dB 噪声后 QAM 星座图

我们在等发射功率的前提下进行对比，为此，在进行系统仿真时，需要将



$E_b/N_0$  转换为 SNR。这是因为仿真中已调信号需通过高斯白噪声信道，高斯白噪声信道的噪声参数是与 SNR 直接相关的。 $E_b/N_0$  与 SNR 的转换公式如下：

$$SNR = Eb/N_0 + 10\log_{10}(k) + 10\log_{10}(coderate) - 10\log_{10}(upfactor)$$

其中  $k$  是每个符号或码元所含的信息比特数， $coderate$  是编码码率， $upfactor$  是上采样率。可以利用 MATLAB 中的 `qammod` 和 `qamdemod` 函数来构建 QAM 调制器和解调器。具体的程序代码见附录。最后我们画出了不同星座图的 BER-SNR 关系曲线，如图 6.3 所示。

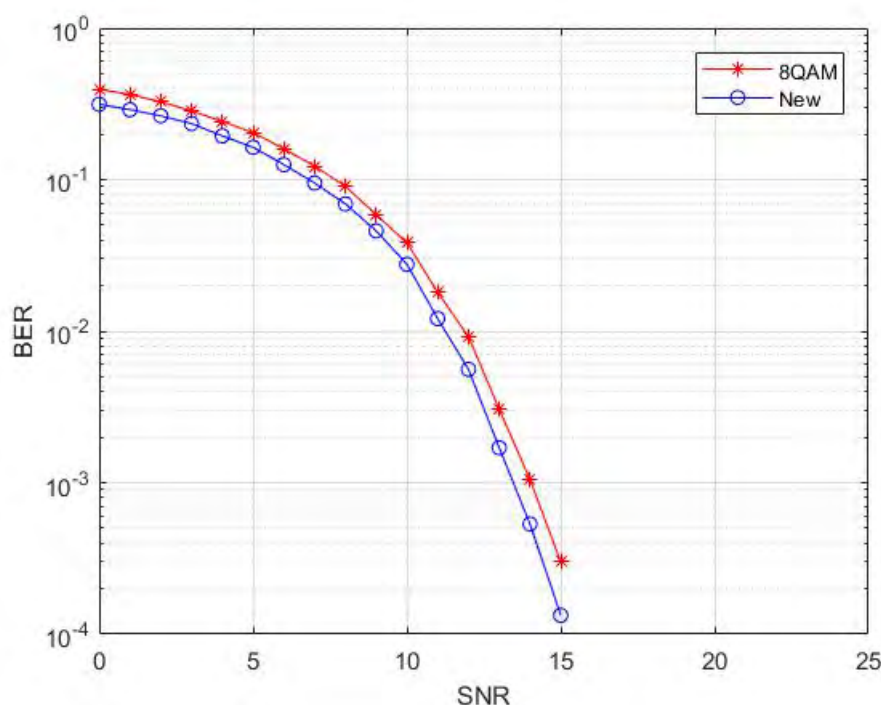


图 6.3 原始 8QAM 与新 QAM 调制格式的 BER-SNR 曲线关系

鉴于此，我们又使用遗传算法（GA）来实现改善 QAM 星座点的标记，得到如图 6.4 和图 6.5 所示

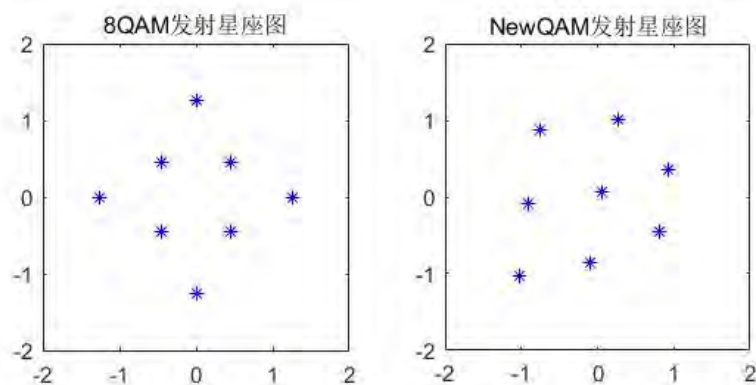


图 6.4GA 算法标记 QAM 星座点



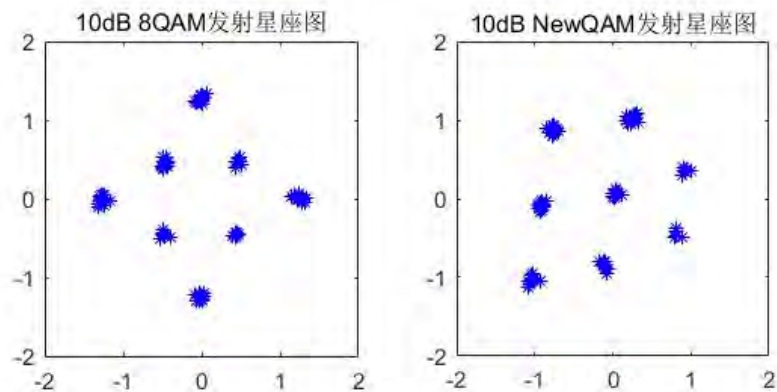


图 6.5 加 10dB 噪声 QAM 发射星座图

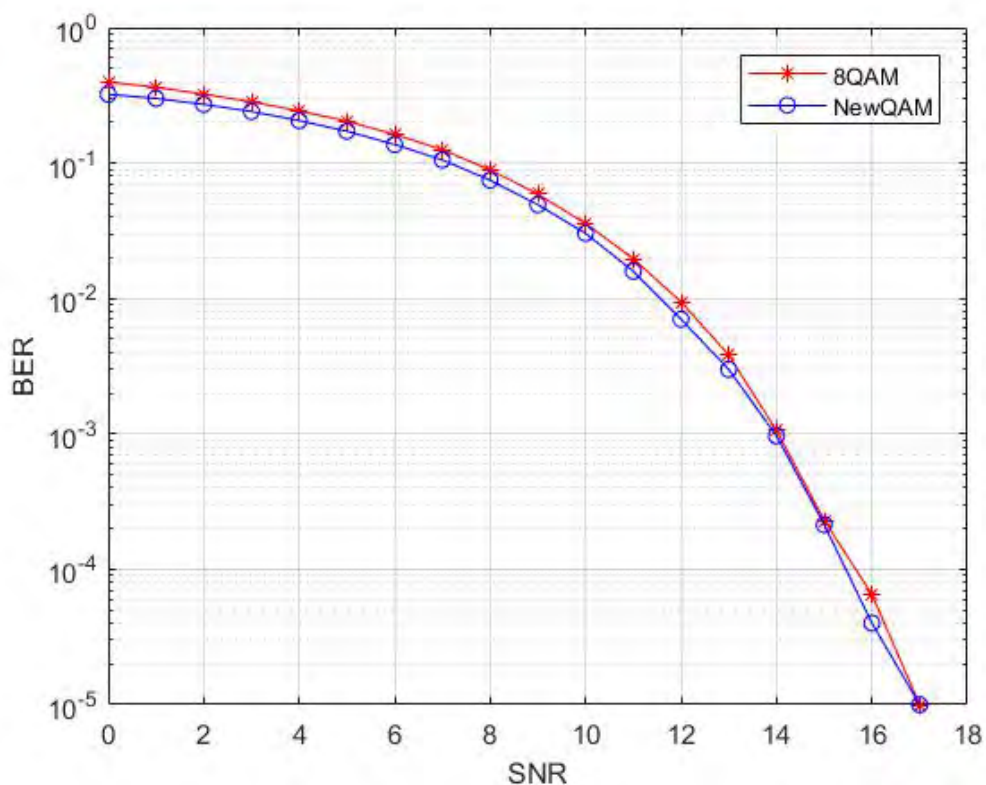


图 6.6 BER-SNR 关系曲线

## 6.8 结果分析

从图 6.3 和图 6.6 可以看出改善的 QAM 星座图的 BER-SNR 曲线最偏下,由此分析可知改变 16QAM 方案中星座点的位置、数量,且每个点以等概率出现,相较于 8QAM 具有更低 SNR 容限点,从而可以提高系统容忍噪声的能力,进而可以延长链路的总长度。QAM 改变了传统上认为最适合应用在卫星通信系统中技术是 QPSK,如今的卫星通信上的转发器的发射功率很高,以往经常遇到功率不足的困扰变成频带资源不足的困扰。QAM 调制技术相对于其它调制技术来说

频带利用率高，传输容量大，抗干扰能力强等优势。QAM 以其高速率传输，安全可靠的性能，被广泛地应用。

## 参考文献

- [1] 余晓玫、高飞，移动通信技术，出版地：西安电子科技大学出版社，p41-p47,2015
- [2] 张辉，曹丽娜编著.《现代通信原理与技术》[M].西安：西安电子科技大学出版社，2002
- [3] 曹志刚，钱亚生编著.《现代通信原理》[M].北京：清华大学出版社，1992
- [4] 曾兴雯，刘乃安，陈健等编著.《高频电路原理与分析》[M].西安：西安电子科技大学出版社，2009
- [5] 宫丰奎，张力，连冰，等.针对 QAM 信号的解调系统的设计[J].电视技术，2012
- [6] 刘顺兰，吴杰等编著《数字信号处理》.西安电子科技大学出版社.2009.7
- [7] 曹志刚等编著《现代移动通信原理》[M].北京：清华大学出版社，1999
- [8] 赵民建，袁梦涛，李世巨，等.全数字多星座图、可变符号率 QAM 调制器[J].电路与系统学报，2001.6
- [9] Liumulu,QPSK 调制原理详解,<https://wenku.baidu.com/view/5d9b84e881c758f5f61f6754.html>, 2018/9/16
- [10] 李明哲，《图论及其算法》，机械工业出版社，56~64，2010
- [11] 敖友云，迟洪钦，《基于遗传算法求解 TSP 问题的一种算法》，《计算机与数字工程》，2006,34(4):52-55
- [12] 高经纬，张煦，李峰，赵晖，《求解 TSP 问题的遗传算法实现》，《计算机时代》，2004(2):19-21
- [13] 侯丽媛，《Dijkstra 算法及其优化策略的分析》，《科技信息》，2010(14):496-496

## 附录（代码）

问题一部分代码：

### Main1-1.m

```
clear all;clc;close all;
SnrSeq = (0:1:25);
Iters = 1e5;
result = zeros(3,length(SnrSeq));
for ii=1:length(SnrSeq)
    TxBit1 = randi([0 3],1,Iters);
    TxSym1 = qammod(TxBit1,4)/sqrt(2);

    TxBit2 = randi([0 7],1,Iters);
    TxSym2 = qammod2(TxBit2,8)/sqrt(4.7);

    TxBit3 = randi([0 15],1,Iters);
    TxSym3 = qammod(TxBit3,16)/sqrt(10);

    Noise = (randn(1,Iters)+randn(1,Iters)*1i)*10^(-SnrSeq(ii)/20)/sqrt(2);
    RxSym1 = TxSym1+Noise;
    RxSym2 = TxSym2+Noise;
    RxSym3 = TxSym3+Noise;

    RxBit1 = qamdemod(RxSym1*sqrt(2),4);
    RxBit2 = qamdemod2(RxSym2*sqrt(4.7),8);
    RxBit3 = qamdemod(RxSym3*sqrt(10),16);

    Bc1 = dec2bin(RxBit1,2)==dec2bin(TxBit1,2);
    Bc2 = dec2bin(RxBit2,3)==dec2bin(TxBit2,3);
    Bc3 = dec2bin(RxBit3,4)==dec2bin(TxBit3,4);

    result(1,ii) = sum(Bc1(:)==0)/Iters/2;
    result(2,ii) = sum(Bc2(:)==0)/Iters/3;
    result(3,ii) = sum(Bc3(:)==0)/Iters/4;
end
figure(1);clf;hold on;
plot(SnrSeq,result(1,:),'r-*');
plot(SnrSeq,result(2,:),'b-o');
plot(SnrSeq,result(3,:),'k-x');
plot(SnrSeq,0.02*ones(size(SnrSeq)),'r--','linewidth',2);
xlabel('SNR');ylabel('BER');
set(gca,'yscale','log');
```

```

legend('QPSK','8QAM','16QAM');
grid on; box on;
Iters = 100;
TxBit1 = randi([0 3],1,Iters);
TxSym1 = qammod(TxBit1,4)/sqrt(2);
TxBit2 = randi([0 7],1,Iters);
TxSym2 = qammod2(TxBit2,8)/sqrt(4.7);
TxBit3 = randi([0 15],1,Iters);
TxSym3 = qammod(TxBit3,16)/sqrt(10);

figure(2);clf;hold on;
subplot(1,3,1);plot(TxSym1,'b*');title('QPSK 发射星座图');axis equal;axis([-2 2 -2 2])
subplot(1,3,2);plot(TxSym2,'b*');title('8QAM 发射星座图');axis equal;axis([-2 2 -2 2])
subplot(1,3,3);plot(TxSym3,'b*');title('16QAM 发射星座图');axis equal;axis([-2 2 -2 2])

SNR = 10;
Noise = (randn(1,Iters)+randn(1,Iters)*1i)*10^(-SnrSeq(ii)/20)/sqrt(2);
RxSym1 = TxSym1+Noise;
RxSym2 = TxSym2+Noise;
RxSym3 = TxSym3+Noise;
figure(3);clf;hold on;
subplot(1,3,1);plot(RxSym1,'b*');title('10dB QPSK 接收星座图');axis equal;axis([-2 2 -2 2])
subplot(1,3,2);plot(RxSym2,'b*');title('10dB 8QAM 接收星座图');axis equal;axis([-2 2 -2 2])
subplot(1,3,3);plot(RxSym3,'b*');title('10dB 16QAM 接收星座图');axis equal;axis([-2 2 -2 2])

```

## Main1-2.m

```

clear all;clc;close all;

f = 193e12;
B = 50e9;
Nf = 4;
TxPower = 1e-3;
TxAmp = sqrt(TxPower);
Lnum = (10:10:250);
%% 第一部分
Iters = 1e5;
result = zeros(3,length(Lnum));
L = 80;
for ii=1:length(Lnum)
    TxBit1 = randi([0 3],1,Iters);
    TxSym1 = qammod(TxBit1,4)/sqrt(2)*TxAmp;

    TxBit2 = randi([0 7],1,Iters);

```

```

TxSym2 = qammod2(TxBit2,8)/sqrt(4.7)*TxAmp;

TxBit3 = randi([0 15],1,Iters);
TxSym3 = qammod(TxBit3,16)/sqrt(10)*TxAmp;

RxSym1 = channel(TxSym1,L,Lnum(ii),f,B,Nf);
RxSym2 = channel(TxSym2,L,Lnum(ii),f,B,Nf);
RxSym3 = channel(TxSym3,L,Lnum(ii),f,B,Nf);

RxBit1 = qamdemod(RxSym1*sqrt(2)/TxAmp,4);
RxBit2 = qamdemod2(RxSym2*sqrt(4.7)/TxAmp,8);
RxBit3 = qamdemod(RxSym3*sqrt(10)/TxAmp,16);

Bc1 = dec2bin(RxBit1,2)==dec2bin(TxBit1,2);
Bc2 = dec2bin(RxBit2,3)==dec2bin(TxBit2,3);
Bc3 = dec2bin(RxBit3,4)==dec2bin(TxBit3,4);

result(1,ii) = sum(Bc1(:)==0)/Iters/2;
result(2,ii) = sum(Bc2(:)==0)/Iters/3;
result(3,ii) = sum(Bc3(:)==0)/Iters/4;
end
figure(1);clf;hold on;
plot(Lnum,result(1,:), 'r-');
plot(Lnum,result(2,:), 'b-o');
plot(Lnum,result(3,:), 'k-x');
plot(Lnum,0.02*ones(size(Lnum)), 'r--', 'linewidth', 2);
legend('QPSK', '8QAM', '16QAM', 'Ber 门限值');
xlabel('光纤段数');ylabel('ber');
title('光纤长度 80Km');
set(gca, 'yscale', 'log');
grid on; box on;

%% 第二部分
Iters = 1e5;
result = zeros(3,length(Lnum));
L = 100;
for ii=1:length(Lnum)
    TxBit1 = randi([0 3],1,Iters);
    TxSym1 = qammod(TxBit1,4)/sqrt(2)*TxAmp;

    TxBit2 = randi([0 7],1,Iters);
    TxSym2 = qammod2(TxBit2,8)/sqrt(4.7)*TxAmp;

    TxBit3 = randi([0 15],1,Iters);

```

```

TxSym3 = qammod(TxBit3,16)/sqrt(34.133)*TxAmp;

RxSym1 = channel(TxSym1,L,Lnum(ii),f,B,Nf);
RxSym2 = channel(TxSym2,L,Lnum(ii),f,B,Nf);
RxSym3 = channel(TxSym3,L,Lnum(ii),f,B,Nf);

RxBit1 = qamdemod(RxSym1*sqrt(2)/TxAmp,4);
RxBit2 = qamdemod2(RxSym2*sqrt(4.7)/TxAmp,8);
RxBit3 = qamdemod(RxSym3*sqrt(34.133)/TxAmp,16);

Bc1 = dec2bin(RxBit1,2)==dec2bin(TxBit1,2);
Bc2 = dec2bin(RxBit2,3)==dec2bin(TxBit2,3);
Bc3 = dec2bin(RxBit3,4)==dec2bin(TxBit3,4);

result(1,ii) = sum(Bc1(:)==0)/Iters/2;
result(2,ii) = sum(Bc2(:)==0)/Iters/3;
result(3,ii) = sum(Bc3(:)==0)/Iters/4;
end

```

```

figure(2);clf;hold on;
plot(Lnum,result(1,:), 'r-*');
plot(Lnum,result(2,:), 'b-o');
plot(Lnum,result(3,:), 'k-x');
plot(Lnum,0.02*ones(size(Lnum)), 'r--', 'linewidth', 2);
legend('QPSK', '8QAM', '16QAM', 'Ber 门限值');
xlabel('光纤段数');ylabel('ber');
title('光纤长度 100Km');
set(gca, 'yscale', 'log');
grid on; box on;

```

### Channel.m

```

function Out = channel(In, L, Num, f, B, Nf)

Tmp=In;
for ii=1:Num
    Tmp=channel_opt(Tmp, L, f, B, Nf);
end
Out = Tmp;

end

```

```

function out = channel_opt(In, L, f, B, Nf)
h = 6.62606896*(1e-34);
%计算光纤的衰减和 Gain 值
Loss_dB = 3*(L/15000);
Loss = 10^(Loss_dB/10);
Gain = 1/Loss;
%线性噪声功率
Pn = 2*pi*h*f*B*(Nf-1/Gain);

InPow = mean(In.^2)/1e3;
Pno = 2*pi*h*f*B*Nf;
%非线性噪声功率
Pnnl = (InPow/(1e-3))^2*Pno*2/3;

%生成入跨噪声信号
NLNoise = (randn(size(In))+randn(size(In))*1i)/sqrt(2)*sqrt(Pnnl);
In2 = In + NLNoise;S

%进行衰减和放大
Out1 = In2/Loss*Gain;

%放大器噪声
LnaNoise = (randn(size(In))+randn(size(In))*1i)/sqrt(2)*sqrt(Pn);
out = Out1+LnaNoise;
end

```

## 第二题部分代码：

### Ga\_TSP.m

```

function ga_TSP(linenum)

% mainly amended by Chen Zhen, 2012~2016
CiteNum = 12;
BitNum=(CiteNum-1)*CiteNum/2; %you chan choose 10, 30, 50, 75
[Clist,CityLoc,CityPop]=testcase;

inn=50; %初始种群大小
gnmax=500; %最大代数
pc=0.8; %交叉概率
pm=0.8; %变异概率

%产生初始种群
s=zeros(inn,BitNum);
for i=1:inn
    s(i,:)=randperm(BitNum);
end

```



```

[~,p]=objf(s,Clist,CityPop,linenum);

gn=1;
ymean=zeros(gn,1);
ymax=zeros(gn,1);
xmax=zeros(inn,BitNum);
scnew=zeros(inn,BitNum);
smnew=zeros(inn,BitNum);
while gn<gnmax+1
    for j=1:2:inn
        seln=sel(p); %选择操作
        scro=cro(s,seln,pc); %交叉操作
        scnew(j,:)=scro(1,:);
        scnew(j+1,:)=scro(2,:);
        smnew(j,:)=mut(scnew(j,:),pm); %变异操作
        smnew(j+1,:)=mut(scnew(j+1,:),pm);
    end
    s=smnew; %产生了新的种群
    [f,p]=objf(s,Clist,CityPop,linenum); %计算新种群的适应度
    %记录当前代最好和平均的适应度
    [fmax,nmax]=max(f);
    ymean(gn)=1000/mean(f);
    ymax(gn)=1000/fmax;
    %记录当前代的最佳个体
    x=s(nmax,:);
    xmax(gn,:)=x;
    gn=gn+1;
    fopt(gn) = fobj(x(end,1:linenum),Clist,CityPop);
end

[f,X]=fobj(xmax(end,1:linenum),Clist,CityPop);
figure;clf;hold on;
plot(CityLoc(:,2),CityLoc(:,1),'rs');
for ii=1:CiteNum
    for jj=1:CiteNum
        if (X(ii,jj)==1)
            plot([CityLoc(ii,2) CityLoc(jj,2)],...
                [CityLoc(ii,1) CityLoc(jj,1)],'b-');
        end
    end
end
end
title(['容量为' num2str(f)])
ylabel('纬度');
xlabel('经度');

```

```

figure;
plot(fopt);
ylabel('网络价值');
xlabel('迭代次数');
title(['连接数为' num2str(linenum)]);
pause(0.01);
figure;
end

%-----
%计算所有种群的适应度
function [f,p]=objf(s,Clist,CityPop,linenum)

inn=size(s,1); %读取种群大小
f=zeros(inn,1);
for i=1:inn
    f(i)=fobj(s(i,1:linenum),Clist,CityPop); %计算函数值，即适应度
end
f=f'; %取距离倒数
%根据个体的适应度计算其被选择的概率
fsum=0;
for i=1:inn
    fsum=fsum+f(i)^15;% 让适应度越好的个体被选择概率越高
end
ps=zeros(inn,1);
for i=1:inn
    ps(i)=f(i)^15/fsum;
end

%计算累积概率
p=zeros(inn,1);
p(1)=ps(1);
for i=2:inn
    p(i)=p(i-1)+ps(i);
end
p=p';
end

%-----
%根据变异概率判断是否变异
function pcc=pro(pc)
test(1:100)=0;
l=round(100*pc);
test(1:l)=1;

```

```

n=round(rand*99)+1;
pcc=test(n);
end

%-----
%“选择”操作
function seln=sel(p)

seln=zeros(2,1);
%从种群中选择两个个体，最好不要两次选择同一个个体
for i=1:2
    r=rand; %产生一个随机数
    prand=p-r;
    j=1;
    while prand(j)<0
        j=j+1;
    end
    seln(i)=j; %选中个体的序号
    if i==2&&j==seln(i-1) %%若相同就再选一次
        r=rand; %产生一个随机数
        prand=p-r;
        j=1;
        while prand(j)<0
            j=j+1;
        end
    end
end
end

%-----
%“交叉”操作
function scro=cro(s,seln,pc)

bn=size(s,2);
pcc=pro(pc); %根据交叉概率决定是否进行交叉操作，1 则是，0 则否
scro(1,:)=s(seln(1),:);
scro(2,:)=s(seln(2),:);
if pcc==1
    c1=round(rand*(bn-2))+1; %在[1,bn-1]范围内随机产生一个交叉位
    c2=round(rand*(bn-2))+1;
    chb1=min(c1,c2);
    chb2=max(c1,c2);
    middle=scro(1, chb1+1:chb2);
    scro(1, chb1+1:chb2)=scro(2, chb1+1:chb2);
end

```

```

scro(2, chb1+1:chb2)=middle;
for i=1:chb1 %似乎有问题
    while find(scro(1, chb1+1:chb2)==scro(1,i))
        zhi=find(scro(1, chb1+1:chb2)==scro(1,i));
        y=scro(2, chb1+zhi);
        scro(1,i)=y;
    end
    while find(scro(2, chb1+1:chb2)==scro(2,i))
        zhi=find(scro(2, chb1+1:chb2)==scro(2,i));
        y=scro(1, chb1+zhi);
        scro(2,i)=y;
    end
end
for i=chb2+1:bn
    while find(scro(1, 1:chb2)==scro(1,i))
        zhi=logical(scro(1, 1:chb2)==scro(1,i));
        y=scro(2, zhi);
        scro(1,i)=y;
    end
    while find(scro(2, 1:chb2)==scro(2,i))
        zhi=logical(scro(2, 1:chb2)==scro(2,i));
        y=scro(1, zhi);
        scro(2,i)=y;
    end
end
end
end

%-----
%“变异”操作
function snnew=mut(snew,pm)

bn=size(snew,2);
snnew=snew;

pmm=pro(pm); %根据变异概率决定是否进行变异操作，1 则是，0 则否
if pmm==1
    c1=round(rand*(bn-2))+1; %在[1,bn-1]范围内随机产生一个变异位
    c2=round(rand*(bn-2))+1;
    chb1=min(c1,c2);
    chb2=max(c1,c2);
    x=snew(chb1+1:chb2);
    snnew(chb1+1:chb2)=fliplr(x);
end

```

```

end

function [Clist,CityLoc,CityNum]=testcase
CityLoc=[39.91667 116.41667,;
         45.75000 126.63333;
         43.45 87.36;
         34.26667,108.95000;
         34.76667,113.65000;
         31.14 121.29;
         30.35 114.17;
         29.35 106.33;
         30.40 104.04;
         29.39 91.08;
         25.04 102.42;
         23.16667,113.23333];
CityNum = [1961.24 1063.60 311.03 846.78 862.65 2301.91 978.54 2884.62 1404.76...
          55.94 643.20 1035.79];

axesm utm    %设置投影方式，这是 MATLAB 自带的 Universal Transverse Mercator （UTM）
方式
Z=utmzone(CityLoc);%utmzone 根据 latlon20 里面的数据选择他认为合适的投影区域，可以是一个
台站的经纬度，也可以是所有台站的经纬度（此时是平均）
setm(gca,'zone',Z)
h = getm(gca);
R=zeros(size(CityLoc));
for i=1:length(CityLoc)
    [x,y]= mfwdtran(h,CityLoc(i,1),CityLoc(i,2));
    Clist(i,:)= [x;y]/1e3;
end
end
Fobj.m
function [f,X]=fobj(XId,CityLoc,CityNum)
CN = size(CityLoc,1);
X = 1-tril(ones(CN,CN),0);
X = X(:);
Ind = find(X==1);
X(Ind)=1:length(Ind);
for ii=1:length(XId)
    X(X==(XId(ii)))=Inf;
end
X=(X==Inf);
X = reshape(X,CN,CN);
X = X+X';

```

%判断是否连接了所有结点的问题

```
Xd = sum(sum(X~=0,1)==0);
```

```
if (Xd>0)
```

```
    f = 0;
```

```
    return;
```

```
end
```

```
f = 0;
```

```
SnrReq = [Inf 3000 1200 600];
```

```
RateTable = [0 8 16 32];
```

```
for ii=1:CN
```

```
    for jj=ii+1:CN
```

```
        if (X(ii,jj)>=1)
```

```
            Dist= norm(CityLoc(ii,:)-CityLoc(jj,:));
```

```
            Ind = find(SnrReq>Dist);
```

```
            ModRate = RateTable(Ind(end));
```

```
            f = f+ sqrt(CityNum(ii)*CityNum(jj))*ModRate;
```

```
        end
```

```
    end
```

```
end
```

**第三题部分代码:**

**Main3-1.m**

```
SnrSeq = (0:1:25);
```

```
Iters = 1e4;
```

```
result = zeros(3,length(SnrSeq));
```

```
for ii=1:length(SnrSeq)
```

```
    TxBit1 = randi([0 7],1,Iters);
```

```
    TxSym1 = qammod2(TxBit1,8)/sqrt(4.7);
```

```
    TxBit2 = randi([0 7],1,Iters);
```

```
    TxSym2 = qammod2(TxBit2,9)/sqrt(0.8793);
```

```
    Noise = (randn(1,Iters)+randn(1,Iters)*1i)*10^(-SnrSeq(ii)/20)/sqrt(2);
```

```
    RxSym1 = TxSym1+Noise;
```

```
    RxSym2 = TxSym2+Noise;
```

```
    RxBit1 = qamdemod2(RxSym1*sqrt(4.7),8);
```

```
    RxBit2 = qamdemod2(RxSym2*sqrt(0.8793),9);
```

```
    Bc1 = dec2bin(RxBit1,3)==dec2bin(TxBit1,3);
```

```

Bc2 = dec2bin(RxBit2,3)==dec2bin(TxBit2,3);

result(1,ii) = sum(Bc1(:)==0)/Iters/2;
result(2,ii) = sum(Bc2(:)==0)/Iters/3;
end

figure(1);clf;hold on;
plot(SnrSeq,result(1,:), 'r-*');
plot(SnrSeq,result(2,:), 'b-o');
plot(SnrSeq,result(3,:), 'k-x');
xlabel('SNR');ylabel('ber');
set(gca, 'yscale', 'log');
legend('8QAM', 'New');
grid on; box on;

Iters = 100;
TxBit1 = randi([0 7],1,Iters);
TxSym1 = qammod2(TxBit1,8)/sqrt(4.7);
TxBit2 = randi([0 7],1,Iters);
TxSym2 = qammod2(TxBit2,9)/sqrt(0.8793);

figure(2);clf;hold on;
subplot(1,2,1);plot(TxSym1, 'b*');title('8QAM 发射星座图');axis equal;axis([-2 2 -2 2])
subplot(1,2,2);plot(TxSym2, 'b*');title('NewQAM 发射星座图');axis equal;axis([-2 2 -2 2])

SNR = 10;
Noise = (randn(1,Iters)+randn(1,Iters)*1i)*10^(-SnrSeq(ii)/20)/sqrt(2);
RxSym1 = TxSym1+Noise;
RxSym2 = TxSym2+Noise;
figure(3);clf;hold on;
subplot(1,2,1);plot(RxSym1, 'b*');title('10dB 8QAM 发射星座图');axis equal;axis([-2 2 -2 2])
subplot(1,2,2);plot(RxSym2, 'b*');title('10dB NewQAM 发射星座图');axis equal;axis([-2 2 -2 2])

Main3-2.m

SnrSeq = (0:1:25);
Iters = 1e5;
result = zeros(3,length(SnrSeq));

bestInd=gp_main();
ampscale = sqrt(mean(abs(bestInd).^2));
for ii=1:length(SnrSeq)

    TxBit1 = randi([0 7],1,Iters);

```

```

TxSym1 = qammod2(TxBit1,8)/sqrt(4.7);

TxBit2 = randi([0 7],1,Iters);
TxSym2 = qammod3(TxBit2,bestInd)/ampscale;

Noise = (randn(1,Iters)+randn(1,Iters)*1i)*10^(-SnrSeq(ii)/20)/sqrt(2);
RxSym1 = TxSym1+Noise;
RxSym2 = TxSym2+Noise;

RxBit1 = qamdemod2(RxSym1*sqrt(4.7),8);
RxBit2 = qamdemod3(RxSym2*ampscale,bestInd);

Bc1 = dec2bin(RxBit1,3)==dec2bin(TxBit1,3);
Bc2 = dec2bin(RxBit2,3)==dec2bin(TxBit2,3);

result(1,ii) = sum(Bc1(:)==0)/Iters/2;
result(2,ii) = sum(Bc2(:)==0)/Iters/3;
end

figure(3);clf;hold on;
plot(SnrSeq,result(1,:), 'r-*');
plot(SnrSeq,result(2,:), 'b-o');
xlabel('SNR');ylabel('ber');
set(gca,'yscale','log');
legend('8QAM','New');
grid on; box on;

Iters = 100;
TxBit1 = randi([0 7],1,Iters);
TxSym1 = qammod2(TxBit1,8)/sqrt(4.7);
TxBit2 = randi([0 7],1,Iters);
TxSym2 = qammod3(TxBit2,bestInd)/ampscale;

figure(4);clf;hold on;
subplot(1,2,1);plot(TxSym1, 'b*');title('8QAM 发射星座图');axis equal;axis([-2 2 -2 2])
subplot(1,2,2);plot(TxSym2, 'b*');title('NewQAM 发射星座图');axis equal;axis([-2 2 -2 2])

SNR = 10;
Noise = (randn(1,Iters)+randn(1,Iters)*1i)*10^(-SnrSeq(ii)/20)/sqrt(2);
RxSym1 = TxSym1+Noise;
RxSym2 = TxSym2+Noise;

```



```
figure(5);clf;hold on;  
subplot(1,2,1);plot(RxSym1,'b*');title('10dB 8QAM 发射星座图');axis equal;axis([-2 2 -2 2])  
subplot(1,2,2);plot(RxSym2,'b*');title('10dB NewQAM 发射星座图');axis equal;axis([-2 2 -2 2])
```