

参赛密码 _____
(由组委会填写)



“华为杯”第十四届中国研究生 数学建模竞赛

学 校 上海电力学院

参赛队号 10256001

1.李阮

队员姓名 2.牛玺童

3.董娜

参赛密码 _____
(由组委会填写)



“华为杯”第十四届中国研究生 数学建模竞赛

题 目 地下物流系统的构建研究

摘 要:

本文建立了地下物流节点选择聚类模型、地下物流节点选择验证模型、地下网络设计双层优化模型、地下物流网络改进模型、地下物流网络时序与优化模型,通过定性与定量的方法,从交通拥挤指数、一级节点转运率、节点服务覆盖率、货运量指数、建设及运营成本、地下网络利用率、地下物流网络满意度、城市交通需求指数进行深入的研究与探讨:

数据的搜集和处理:借助题目中已给信息和网络资源查阅的相关资料,确定区域中心之间距离、各区域货运量以及南京市地铁的分布。

针对问题一:

在保证一级节点转运率尽量低的情况下,为实现交通拥挤指数的改善和节点服务覆盖率尽量大的聚类目标,采用改进的 ISODATA 算法进行聚类,得到了 8 个一级节点和 22 个二级节点,并利用超效率 DEA 模型检验区域节点的效率,检验结果和聚类得到的一二级节点结果大致相同,由此验证了一二级节点的正确性。

一级节点	对应的二级节点	一级节点	对应的二级节点
894	898,896	870	864, 855 ,836, 876
888	891 ,886	832	826, 839
803	792 ,799 ,796	897	899, 892
817	814 ,793 ,827, 828	853	844 ,857 ,859

针对问题二:

在第一问的基础上构建了地下物流网络双层优化模型。以改善交通拥堵为上层目标,以最小化建设及运营成本为下层目标,基于改进的粒子群算法对模型进行求解,选择出最优的地下通道网络。通过对地下物流网络双层优化模型的求解,得到了地下物流的规划方案,该方案的年均建设成本折旧及运营费用为

5,756,533,247 元。

针对问题三：

在第二问模型的基础上，基于减少建设成本、运营成本和提高地下物流网络抗风险能力建立了地下物流网络改进模型。该模型由三个子模型构成，即地下物流网络满意度、地下物流网络抗风险、地下物流网络调整模型。分别采用改进的 Dijkstra 算法、改进的粒子群算法和模拟退火算法进行模型的求解工作，经过 37 次循环后，得到改进后的地下物流网络模型，改进后的地下物流网络抗风险指数为 6.523，网络满意度为 6.982，从而说明第二问的模型有改进的需要。

针对问题四：

基于动态规划理论建立了时序与动态地下物流网络优化模型，该模型以交通拥堵情况的改善程度最大化和建造及运营成本的最小化为目标，综合考虑了城市交通需求的增长和每年建设的线路长度基本相同的约束条件，利用动态规划的经典求解过程对模型进行求解。结果表明，第三问的改进网络模型不再满足未来 30 年的交通需求的增长，需要进行扩容，如按照动态规划的设计思路进行设计，将会满足城市交通需求增长，因此依据时序与动态地下物流网络优化模型进行的动态网络建设更加符合当地实际需求。

关键词：地下物流网络、节点选择、网络满意度、交通拥挤指数、动态规划

1 问题重述

1.1 问题背景

世界大城市都遇到的一个“困局”就是交通拥堵。荷兰导航经营商 TomTom 在 2015 年发布了全球最拥堵城市排名,有十个中国大陆的城市位列前三十名。2014 年,据中国交通部发布的数据,我国城市人口可支配收入的经济损失的 20%是交通拥堵带来的,相当于每年国内生产总值(GDP)损失 5~8%。我国 15 座大城市的居民每天上班相对于欧洲发达国家要多消耗 28.8 亿分钟。大量研究表明:世界总消耗量的 20%是由于“时走时停”的交通导致原油消耗所产生的。高峰期,北京市主干线上拥堵 1 小时,300 万辆机动车所需燃油为 240 万~330 万升。2015 年城市交通规划年会发布数据显示:在我国石油消费方面,消费总量的 54%是交通石油消费,全社会总能耗 10%以上是交通能耗,并且逐年上升。高能耗也表示着高污染和高排放^[1-2]。

交通需求激增所带来的地面道路上车辆、车次数量的巨增是导致城市交通拥堵的主要原因。虽然货车占城市机动车总量的比例不大,但是由于货运车辆一般体积较大且载重时行驶较慢,如果车流中混入重型车,道路的通行能力会明显降低,因此,占用城市道路资源比例较大。如北京,货运车辆按常规的车辆换算系数计算(不同车辆在行驶时占用道路净空间的程度),其占用的道路资源达 40%。因此,为解决城市交通和环境问题,世界各国都在进行积极探索,而处理好货运交通已成为共识。大量实践证明,地面道路不可能无限制地增加,仅通过增加地面交通设施来满足不断增长的交通需求,既不科学也不现实。因此“统筹规划地上地下空间开发”势在必行,越来越多发达国家对“地下物流系统”提高重视^[3]。

1.2 问题提出

由于我国人口众多、大城市密集、交通状况不佳已经到了迫切需要改善的程度,而且地下工程技术水平也基本满足要求,因此构建地下物流系统已经势在必行,而构建地下物流系统网络是其中的关键性步骤。但是目前面临着建造成本高、网络形式多样、流量大等问题。因此为了缓解交通拥堵、降低物流成本,正确选择地下物流网络的节点群、地下通道的网络设计和地下通道筹建是当前多国政府正考虑和着手落实的重点领域^[4]。

1.3 本文所要解决的问题

(1)地下物流节点选择:根据该区域的实际情况建立该区域节点选择模型,确定该区域地下物流网络节点群。

(2)地下物流通道网络设计:在地下物流网络节点群的基础上选择合适的地下路线以建立该区域的“地下物流系统”网络。在转运率变化不大的情况下,若考虑优化网络,可适当调整一、二级节点位置。除园区至一级节点的地下通道外其他地下通道均采用 5 吨的地下运输车辆。

(3)网络改进:以上是分步设计网络,并未从全局出发,根据你们对地下物流网络模型仿真结果,研究设计得到的地下物流通道网络有无修改的必要?能否通过增加、减少节点的个数,调整节点的位置或级别,增加、减少、改变路径的

方法缩短货物运输总里程(同时节省运输时间),降低运输成本。进一步从增强 ULS 的抗风险能力(如某通道中断,某方向货运量激增)考虑,需要对第二问的 ULS 作怎样的改动。

(4)时序与动态优化:地下物流系统的造价高,风险大,改建困难,所以应做好顶层设计,考虑满足该市近 30 年内的交通需求(可以认为需求量每年呈 5% 增长)并根据建设进度分八年完成“地下物流网络系统”的建设(每年可建设道路长度大致相等),制定南京市仙林区地下物流系统网络各线路的建设时序及演进过程,与第三问得到的地下物流系统网络进行对比并比较优劣。

2 模型假设与符号说明

2.1 符号说明

符号	意义
TPI	交通拥堵指数
$S_i(t)$	第 i 区域节点第 t 天的货运量
φ	节点转运率
$S_{\min dyi}$	最近的一级节点 $\min qyi$ 转运至其他所有的节点的货物量
S_{qi}	qi 个工业园区的货运量
F_{JY}	节点服务覆盖率
Ssi	第 i 个节点服务覆盖面积
$Squre$	该区域的总面积
F_1	交通拥堵指数改善情况
$TPI_i(t)$	第 i 个节点在第 t 天的交通拥堵指数
$TPI_i(0)$	初始当天的交通拥堵指数
$SHY(yi)$	一级节点的货物量
$SHT(ti)$	二级节点的货物量
$S(i, j)$	通车的地下节点之间的距离受航班和车辆行驶速度的约束
yn	进出 4 个物流园区放入地下的货物
$\{Z_1, \dots, Z_8\}$	8 个初始聚类中心
K	预期的聚类中心数目
θ_N	每一聚类域中最少的样本数目
θ_s	一个聚类域中样本距离分布的标准差
θ_c	两个聚类中心间的最小距离
L	一次迭代运算中可以合并的聚类中心的最多对数
D_j	距离最小
N_i	聚类数目
S_i	聚类域
$\overline{D_j}$	各聚类域中样本与各聚类中心间的平均距离
\overline{D}	全部模式样本和其对应聚类中心的总平均距离
σ_j	每个聚类中样本距离的标准差向量

θ	相对效率
λ	线性组合系数
$G(x)$	拥堵改善约束
$g(x, y_1, y_2, \dots, y_m)$	物流成本约束
$FTIY_{yi}$	地下一级节点货运量指数
S_{yi} 、 S_{ti}	一、二级节点 yi 货运量
$S_{yi \max}$ 、 $S_{ti \max}$	一、二级节点中地下货运量最大
$FTIT_{ti}$	地下二级节点货运量指数
$SI_i(t)$	地下节点 i 在第 t 天的进货量
$SO_i(t)$	地下节点 i 在第 t 天的出货量
F_{down}	地下物流系统中成本优化模型为评价指标
$\lambda(i, j)$	地下物流系统的建造单位成本
C_{yn}	第一节点的造价
C_m	第二节点的单位造价
Z	项目折旧
$F(x, y)$	上层模型的适应度函数
η	变异因子
PGB_i	节点网络饱和度
$LGB_{i,j}$	线路饱和度
TUO	地下网络利用率
ΔC_{grid}	线路优化前后造价变化
q_1 、 q_2	相应指标的权重
ACC_{yi} 、 ACC_{ti}	一、二级节点允许净增进货量
ACO_{yi} 、 ACO_{ti}	一、二级节点允许净增出货量
$FTAR$	地下物流网络抗风险指数
$n_{bd \max}$	最大通道中断数
T	控制参数
L_k	链长度
P	Metropolis 的接受准则相对应的转移概率
$CTI(t)$	城市交通需求指数
$CFTI(t)$	城市地下物流交通指数
$S_{dis}(i, j)$	道路长度

2.2 模型假设

- 两节点之间的距离取直线距离，认为两节点间的地质等因素不影响建设施工。
- 南京市仙林区的地铁线路对于地下物流系统网络规划不产生影响。
- 班车返回途中也会运载同样多的货物量。
- 班车在两节点之间一直不间断发车，每天 18 小时，每小时 5 班次。
- 扩容过程假设不能拆除线路只能增设线路。
- 各节点货运量和各通道的流量的年增长率为 5%。

3 问题分析

3.1 针对问题一

本题目的在于根据该区域实际的情况建立节点选择模型,用来确定该区域的地下物流网络节点群。该节点群尽量满足以下要求:交通拥堵可以得到缓解、节点的转运率相对较小、节点覆盖面积尽量大、一二级节点的收发货量上限等。在满足上述要求的基础之上建立合理的节点物流网络节点群。该题可以考虑成是带约束的聚类问题,因此选择迭代自组织数据分析算法(ISODATA)。聚类的最终结果目的是使得一级节点转运率最小、节点服务覆盖率尽量大、交通拥堵指数最大程度上的改善。相关约束量为:一、二级节点的货运量上限、通车的地下节点之间的距离受航班和车辆行驶速度的约束、进出4个物流园区的货物尽可能放入地下运输。在满足以上约束的条件下完成对节点的选取。模型校验:为了检验上述节点分配的合理性,接下来将借助SE-DEA模型对其进行检验。

3.2 针对问题二

该题目是在问题一的基础之上,适当的调整一、二级节点的位置以及考虑地下运输车辆的类型的情况下选择合适的地下线路,建立该区域的“地下物流系统”网络。该网络需要符合缓解交通拥堵和降低物流成本的要求。为此基于双层优化建立了地下物流系统网络双层优化模型,模相关约束如下:上层为拥堵改善模块,用来寻找改善的最优的情况,相应的约束为交通拥堵指数、地上货物运输量、地下货物运输量。下层为物流成本优化模块,用于计算地下物流网络成本的优化,相应的约束为地下物流线路长度、地上车辆运输货物量。双层优化的两层相互影响和相互修正,上层决策结果一般会影响下层目标,而下层将决策结果反馈给上层又对上层产生影响,从而实现上下层决策的相互作用。

3.3 针对问题三

从全局出发,验证上一问的结果有无修改的必要。并且在考虑抗风险的情况下,网络作一定的调整。为了验证上述网络的合理性,建立地下物流网络改进模型。该模型包括三个子模型:地下物流网络满意度模型、地下物流网络抗风险检验模型和地下物流网络调整模型

地下物流网络满意度模型首先利用Dijkstra算法在改进模型的基础之上进行路径优化,对改进模型进行验证。然后在考虑地下网络利用率和交通拥堵指数两个指标基础之上,进行满意度指标的计算。

地下物流网络抗风险检验模型需要考虑一、二级节点的进、出货运量的最大净增量和地下物流网络最大允许的中断线路条数下的抗风险指数。利用模拟退火算法进行网络的抗风险性校验。

地下物流网络调整模型为网络优化模型,意在对网络进行满意度和抗风险能力的优化。基于双层优化建立地下物流网络调整模型,上层为满意度优化模块,下层为抗风险能力优化模块。改进的粒子群算法对双层优化模型进行优化改进。

3.4 针对问题四

考虑地下物流系统可以满足该区地区近 30 年的交通需求，且根据建设进度分八年完成“地下物流网络系统”的建设时序以及演进过程。本问题是典型的动态规划问题，考察如何在规定的时间内逐步完成网络的规划，设计的主要目标仍是交通拥堵情况的改善程度最大化和建造及运营成本的最小化。考虑城市交通需求、地下节点及通道内的货物每天要清仓、节点的地面收发货量上限、通车的地下节点之间的距离受航班和车辆行驶速度的约束等因素情况下建立地下物流网络时序与动态优化模型。利用多阶段决策过程最优化问题的动态规划算法对模型进行求解，得到八年建设过程的建设规划。对第三问设计的地下物流网络进行城市交通需求验证，并给出扩容方案。

4 问题一模型的建立与求解

4.1 地下物流节点选择模型的建立

该问题是一个带有约束条件的聚类问题，需要满足约束条件的前提下完成一、二级节点的选取。

4.1.1 地下物流节点的选择指标

(1) 交通拥堵指数(Traffic Performance Index(TPI))

交通拥堵指数是综合反映道路网畅通或拥堵的概念性数值，简称交通指数。交通指数取值范围为 0 至 10，每 2 个数为一等级，分别对应“畅通”、“基本畅通”、“轻度拥堵”、“中度拥堵”、“严重拥堵”五个级别，数值越高，表明交通拥堵状况越严重。

由于该地区非人口密集区，可以近似认为区域交通的拥堵指与区域进出口之和成正比，通过对已知数据进行拟合，得到的拟合正比例函数为：

$$TPI(S_i(t)) = 0.00267S_i(t) \quad (1)$$

式中， $S_i(t)$ 表示第 i 区域节点第 t 天的货运量。

(2) 一级节点转运率(φ)

从物流园区经由最近的一级节点转运至其他所有一级节点的货物量占到物流园区总出货量的百分比，成为一级节点的转运率：

$$\varphi = \frac{\sum_{\min d_{yi}=1}^4 S_{\min d_{yi}}}{\sum_{qi=1}^4 S_{qi}} \quad (2)$$

式中， $S_{\min d_{yi}}$ 最近的一级节点 $\min q_{yi}$ 转运至其他所有的节点的货物量； S_{qi} 第 qi 个工业园区的货运量。

(3) 节点服务覆盖率

节点服务覆盖面积占区域总面积的百分比：

$$INSsi = \frac{\sum_{i=1}^n Ssi}{Squire} \quad (3)$$

式中， Ssi 表示第 i 个节点服务覆盖面积， $Squire$ 该区域的总面积。

(4) 节点强度

考虑加权网络边带的权重，将点的强度 S_i 定义为：

$$S_i = \sum_{j \in N_i} \omega_{ij} \quad (4)$$

式中， ω_{ij} 表示点 i 和点 j 的权重。

两点之间的权重由货运量和之间距离共同决定：

$$\omega_{ij} = \mu_1 S(i, j) + \mu_2 S_{dis}(i, j) \quad (5)$$

式中， $S(i, j)$ 表示点 i 和点 j 之间的货运量， $S_{dis}(i, j)$ 表示点 i 和点 j 之间的距离。

南京市仙林区是江苏省发展高等教育重点地区，高等学府如南京大学、南京师范大学均在仙林区设有校区，人口密度虽不大，但货物吞吐量较大，根据货运 OD 数据，典型日货物运输总量达 32.7 万吨。综合考量地下物流系统的建设成本以及相应货物运输量，对应的权重取为 $\mu_1 = \mu_2 = 0.5$ ，即地下物流系统的规划者对于距离和货运量之间没有偏好。将节点按强度大小在图中以不同面积的点表示，得到节点强度在复杂网络模型中的示意图(图 1)。

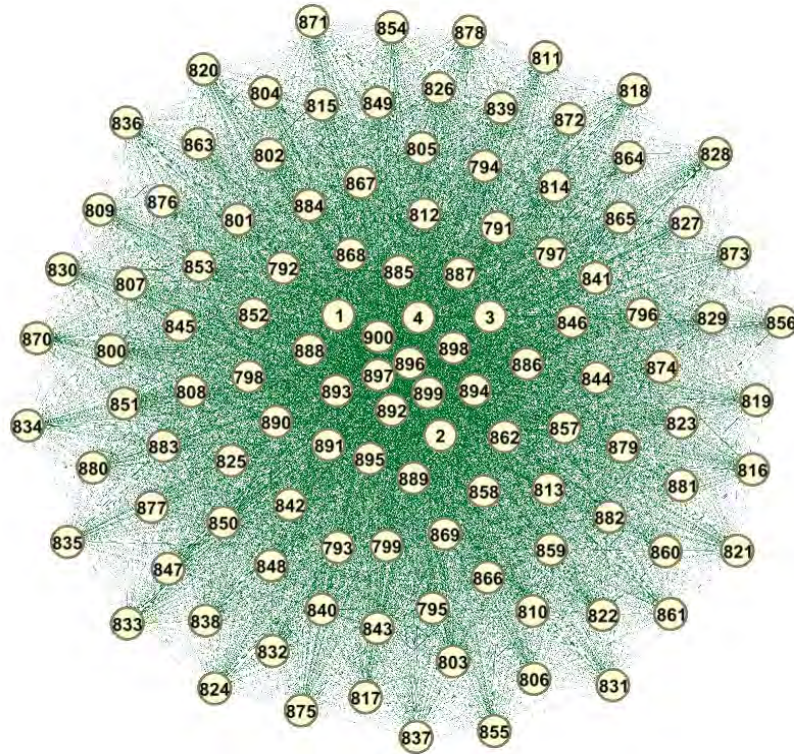


图 1 各区域节点和园区节点强度关系

4.1.2 地下物流节点选择模型聚类目标

(1) 一级节点转运率(φ)尽量小

$$F_{\varphi Y} = -\varphi = -\frac{\sum_{\min dyi=1}^4 S_{\min dyi}}{\sum_{qi=1}^4 S_{qi}} \quad (6)$$

式中， $S_{\min dyi}$ 最近的一级节点 $\min qyi$ 转运至其他所有的节点的货物量； S_{qi} 第 qi 个工业园区的货运量。

(2) 节点服务覆盖率尽量大

节点服务覆盖面积占区域总面积的百分比：

$$F_{JY} = \frac{\sum_{i=1}^n Ss_i}{S_{qure}} \quad (7)$$

式中， Ss_i 表示第 i 个节点服务覆盖面积， S_{qure} 该区域的总面积。

(3) 交通拥堵指数改善情况：

$$F_1 = \sum_{i=1}^n [TPI_i(0) - TPI_i(t)] \quad (8)$$

式中， $TPI_i(t)$ 表示第 i 个节点在第 t 天的交通拥堵指数， $TPI_i(0)$ 表示初始当天的交通拥堵指数。

综上，目标函数可表示为：

$$\max F_{Xbest} = (F_{\phi Y} + F_{JY} + F_1) = \frac{\sum_{i=1}^n Ss_i}{S_{qure}} + \sum_{i=1}^n [TPI_i(0) - TPI_i(t)] - \frac{\sum_{\min dyi=1}^4 S_{\min dyi}}{\sum_{qi=1}^4 S_{qi}} \quad (9)$$

4.1.3 地下物流节点选择模型约束条件

(1) 一级节点的货物量上限约束

一级节点的地面收发货量上限为 4000 吨。

$$SHY(yi) \leq 4000 \quad (10)$$

(2) 二级节点的货物量上限约束

二级节点的地面收发货量上限为 3000 吨。

$$SHT(ti) \leq 3000 \quad (11)$$

(3) 通车的地下节点之间的距离受航班和车辆行驶速度的约束(单位：米)。

$$S(i, j) \leq 9800 \quad (12)$$

(4) 进出 4 个物流园区的货物尽可能放入地下运输

$$yn \leq \frac{\sum_{qi=1}^4 S_{qi}}{S_{yi \max}} \quad (13)$$

节点选择模型聚类目标及约束为：

$$\max F_{Xbest} = (F_{\phi Y} + F_{JY} + F_1)_{\max} = \frac{\sum_{i=1}^n Ss_i}{S_{qure}} + \sum_{i=1}^n [TPI_i(0) - TPI_i(t)] - \frac{\sum_{\min dyi=1}^4 S_{\min dyi}}{\sum_{qi=1}^4 S_{qi}} \quad (14)$$

$$s.t. \quad SHY(yi) \leq 4000$$

$$SHT(ti) \leq 3000$$

$$S(i, j) \leq 9800$$

$$yn \leq \frac{\sum_{qi=1}^4 S_{qi}}{S_{yi \max}}$$

4.2 节点选择聚类模型的求解

由于聚类分析算法能够挖掘数据之间的内在联系, 广泛应用于数据归类处理等问题中, 相关的衍生算法应运而生。ISODATA 算法是聚类分析衍生算法应用较为普遍的算法之一, 加入了一些试探步骤, 并且可以结合成人机交互的结构, 使其能利用中间结果所取得的经验更好地进行分类。本文基于 ISODATA 算法思想, 加入一级节点聚类点的寻优约束, 尽可能保证一级节点的个数最少的同时能够有较低的一级节点转运率, 对节点选择模型进行求解^[4-5]。

4.2.1 ISODATA 算法基本步骤

ISODATA 算法的主要步骤分为以下 6 步: (1)选择某些初始值。可选不同的参数指标, 也可在迭代过程中人为修改, 以将 N 个模式样本按指标分配到各个聚类中心中去。(2)计算各类中诸样本的距离指标函数。(3)~(5)按给定的要求, 将前一次获得的聚类集进行分裂和合并处理((4)为分裂处理, (5)为合并处理), 从而获得新的聚类中心。(6)重新进行迭代运算, 计算各项指标, 判断聚类结果是否符合要求。经过多次迭代后, 若结果收敛, 则退出运算并保留结果。具体流程见图 2。

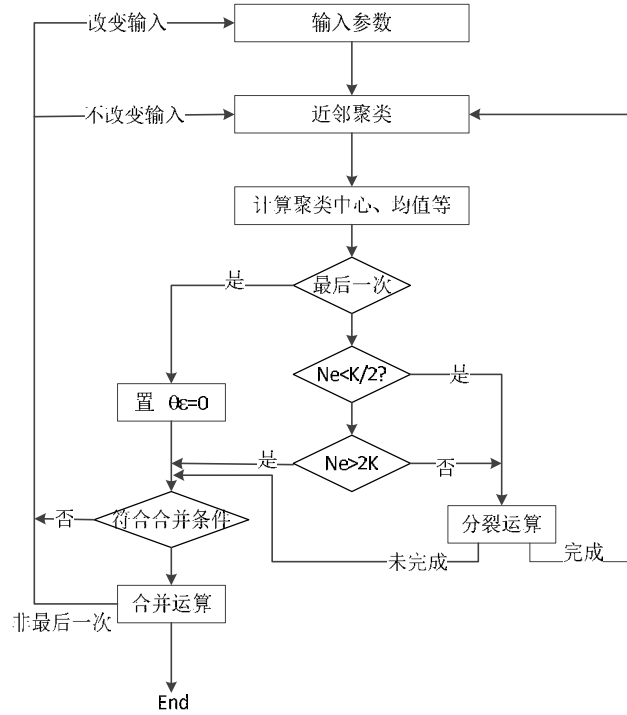


图 2 ISODATA 算法流程图

4.2.2 改进的 ISODATA 算法求解过程

依据 ISODATA 算法流程对包含约束和目标的节点选择模型进行求解。

(1)输入的样本为 110 个模式 $\{x_i, i=1, 2, \dots, 110\}$

预选 8 个初始聚类中心 $\{Z_1, Z_2, \dots, Z_8\}$, 它可以不等于所要求的聚类中心的数目, 其初始位置可以从样本中任意选取。

预选: K 为预期的聚类中心数目, 本文 $K=3$;

θ_N 为每一聚类域中最少的样本数目, 若少于此数即不作为一个独立的聚类, 本文 $\theta_N=3$;

θ_s 为一个聚类域中样本距离分布的标准差, 本文 $\theta_s=1.8$;

θ_c 为两个聚类中心间的最小距离，若小于此数，两个聚类需进行合并，本文 $\theta_c = 4$ ；

L 为一次迭代运算中可以合并的聚类中心的最多对数，本文 $L = 1$ ；

I 为迭代运算的次数，本文 $I = 100$ 。

(2) 将 110 个模式样本分给最近的聚类 N_i ，假若 $D_j = \min \{\|x - Z_i\|, i=1, 2, \dots, N_c\}$ ，即 $\|x - Z_i\|$ 的距离最小，则 $x \in N_i$ 。

(3) 如果 N_i 中的样本数目 $N_i < \theta_N$ ，则取消该样本子集，此时 $N_c = N_c - 1$ 。

(4) 修正各聚类中心

$$Z_j = \frac{1}{N_j} \sum_{x \in N_j} x, j=1, 2, \dots, N_c \quad (15)$$

(5) 计算各聚类域 S_i 中模式样本与各聚类中心间的平均距离：

$$\overline{D}_j = \frac{1}{N_j} \sum_{x \in N_j} \|x - Z_j\|, j=1, 2, \dots, N_c \quad (16)$$

(6) 计算全部模式样本和其对应聚类中心的总平均距离：

$$\overline{D} = \frac{1}{N} \sum_{j=1}^N N_j \overline{D}_j \quad (17)$$

(7) 判别分裂、合并及迭代运算以及一级节点转运率寻优

循环计算一级节点转运率(ϕ)，进行局部寻优，遍历各个节点，直至找出一级节点转运率最低的一组一级节点，若迭代运算次数已达到 I 次，则置 $\theta_c = 0$ ，转至第十一步。若 $N_c \leq K/2$ ，即聚类中心的数目小于或等于规定值的一半，则转至第八步，对已有聚类进行分裂处理。若迭代运算的次数是偶数次，或 $N_c \leq 2K$ ，不进行分裂处理，转至第十一步；否则，转至第八步，进行分裂处理。

(8) 计算每个聚类中样本距离的标准差向量

$$\sigma_j = (\sigma_{1j}, \sigma_{2j}, \dots, \sigma_{nj})^T \quad (18)$$

其中向量的各个分量为

$$\sigma_{ij} = \sqrt{\frac{1}{N_j} \sum_{k=1}^{N_j} (x_{ik} - Z_{ij})^2} \quad (19)$$

式中， $i=1, 2, \dots, n$ 为样本特征向量的维数， $j=1, 2, \dots, N_c$ 为聚类数， N_j 为中的样本个数。

(9) 求每一标准差向量 $\{\sigma_j, j=1, 2, \dots, N_c\}$ 中的最大分量，以 $\{\sigma_{j\max}, j=1, 2, \dots, N_c\}$ 代表。

(10) 在任一最大分量集 $\{\sigma_{j\max}, j=1, 2, \dots, N_c\}$ 中，若有 $\sigma_{j\max} > \theta_s$ ，同时又满足 $\overline{D}_j > D$ 和 $N_j > 2(\theta_N + 1)$ ，即 N_i 中样本总数超过规定值一倍以上； $N_c \leq \frac{K}{2}$ 。则将 Z_j 分裂为两个新的聚类中心和，且 N_c 加 1。

(11) 计算全部聚类中心的距离

$$D_{ij} = \|Z_i - Z_j\|, i=1, 2, \dots, N_c - 1; j=i+1, \dots, N_c \quad (20)$$

(12) 比较 D_{ij} 与 θ_c 的值，将 $D_{ij} < \theta_c$ 的值按最小距离次序递增排列，即

$$\{D_{i_1, j_1}, D_{i_2, j_2}, D_{i_3, j_3}, \dots, D_{i_L, j_L}\} \quad (21)$$

式中 $D_{i_1, j_1} < D_{i_2, j_2} < \dots < D_{i_L, j_L}$ 。

(13)将距离 D_{i_k, j_k} 的两个聚类中心 Z_{ik} 和 Z_{jk} 合并，得新的中心为：

$$Z_k^* = \frac{1}{N_{ik} + N_{jk}} (N_{ik} Z_{ik} + N_{jk} Z_{jk}) \quad (22)$$

式中，被合并的两个聚类中心向量分别以其聚类域内的样本数加权，使 z_k^* 为真正的平均向量。

(14)若是最后一次迭代运算(即第 100 次)，则算法结束；否则，根据参数的变化决定，若需要操作者改变输入参数，转至第一步；若输入参数不变，转至第二步。在本步运算中，迭代运算的次数每次应加 1。

4.3 节点选择

通过上述算法进行节点选择模型利用 matlab 编程进行求解，相关程序见附件 2。

4.3.1 各级节点情况

通过对节点的聚类计算，得到了 8 个一级节点和 22 个二级节点，具体节点详见表 1。

表 1 节点选择情况

一级节点	对应的二级节点	一级节点	对应的二级节点
894	898	870	864, 855, 836, 876
888	891, 886	832	826, 839
803	792, 799, 796	897	896, 899, 892
817	814, 793, 827, 828	853	844, 857, 859

通过聚类计算后，8 各节点分属于不同的区域，如图 3 所示，不同区域之间的二级节点只能通过一级节点和其他区域的节点通过地下物流系统网络进行货物的运输。

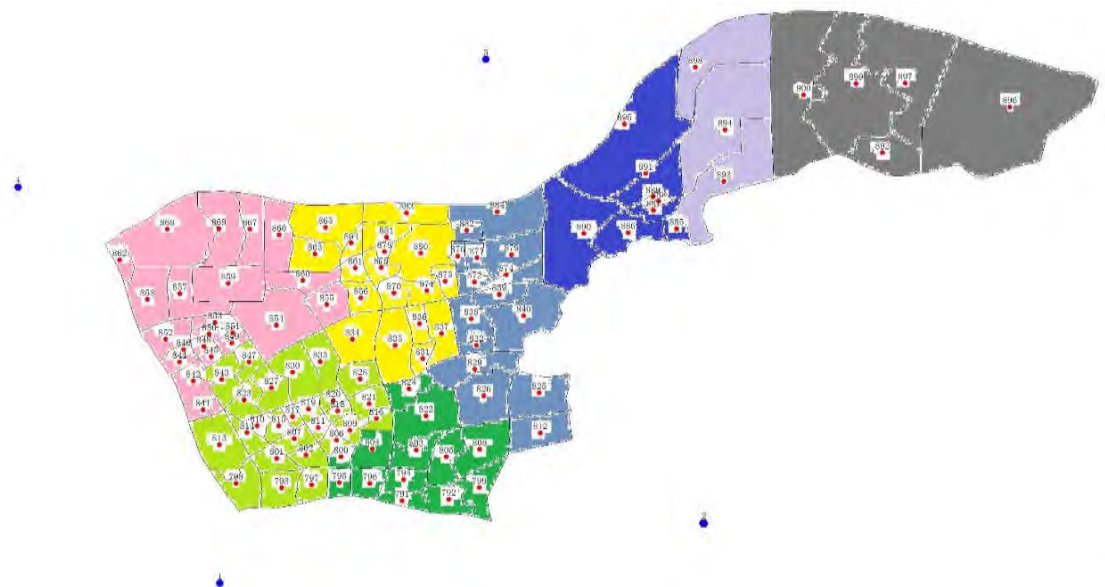


图 3 ISODATA 聚类结果

4.3.2 节点服务范围

4.3.4 服务范围覆盖率:

4.3.4 各节点实际货运量

表2 一级节点实际货运量情况

区域划分	一级节点	地面（单位：吨）		地下（单位：吨）	
		进货量	出货量	进货量	出货量
区域 1	894	1978.43	1868.793	3502.63	3612.267
区域 2	888	1236.793	1976.56	3651.26	2911.493
区域 3	803	1589.635	2263.896	3658.783	2984.522
区域 4	817	1578.885	2365.789	4859.631	4072.727
区域 5	870	1786.938	1689.561	3856.231	3953.608
区域 6	832	2569.437	1389.563	4205.123	5384.997
区域 7	897	2428.56	1523.461	4425.158	5330.257
区域 8	853	2110.178	1251.791	4258.963	4517.35

区域划分	二级节点	地面（单位：吨）		地下（单位：吨）	
		进货量	出货量	进货量	出货量
区域 1	898	1534.27	1174.304	3201.405	3561.308
区域 2	891	743.211	1456.709	2002.598	1289.1

区域 3	886	645.408	1357.905	1865.495	1152.998
	796	803.409	1078.507	1236.957	961.859
	799	606.507	1127.403	1497.874	976.978
	792	803.786	908.405	1597.369	1492.75
区域 4	814	653.852	790.436	1382.602	1246.018
	793	597.346	845.205	1258.725	1010.866
	827	774.501	832.783	1582.371	1524.089
	828	645.832	642.743	1370.274	1373.363
区域 5	864	684.381	803.284	1532.562	1413.659
	855	842.104	931.421	1626.301	1536.984
	836	742.194	732.354	1208.303	1218.143
	876	843.195	612.363	1183.592	1414.424
区域 6	826	1363.83	1455.749	1429.501	1337.584
	839	987.42	1681.213	2042.501	1348.708
区域 7	892	1472.91	1133.193	1128.371	1467.569
	899	1891.33	928.39	1837.371	2800.304
	897	1381.21	1037.382	1231.582	1575.401
区域 8	844	824.281	720.301	1720.873	1824.853
	857	812.62	802.479	1593.928	1604.069
	859	920.358	932.204	1287.275	1275.429

4.3.4 一级节点转运率

根据一级节点转运率的定义来看, 仅仅需要计算从物流园区仅有最近的一级节点转运至其他所有一级节点的货物量占物流园区总出货量的百分比, 相关第一级节点转运率如表 4 所示。

表 4 一级节点转运率

距离物流园区最近的一级节点	一级节点转运率
817	34.12%
803	30.42%
888	29.14%
853	77.91%

4.4 节点选择验证模型

为了检验上述节点分配的有效性, 接下来将借助 SE-DEA 模型对其进行检验。

4.4.1 DEA 模型

效率性的研究大都基于各种投入产出效率模型, 主要用于相同类型决策单元之间多项投入和产出的效率评价。假设有 m 种投入和 q 种产出 n 个 DMU 的技术效率, 投入导向型的 BCC-DEA 模型表达式为^[6]:

$$\min \theta \quad (23)$$

$$\begin{aligned}
s.t. \quad & \sum_{j=1}^n \lambda_j x_{ij} \leq \theta x_{ik}, i=1,2,\dots,m; j=1,2,\dots,n; \\
& \sum_{j=1}^n \lambda_j y_{rj} \geq y_{rk}, r=1,2,\dots,q; \\
& \sum_{j=1}^n \lambda_j = 1 \\
& \lambda \geq 0
\end{aligned}$$

(23)式中, θ 为相对效率, x_{ij} 为投入, y_{rj} 为产出, λ 为线性组合系数。BCC模型是在 CCR 模型的基础上加了约束条件 $\sum_{j=1}^n \lambda_j = 1$ 构成的, 主要是为了使投影的生产规模与被评价 DMU 的生产规模处于同一水平^[7]。

4.4.2 超效率 DEA(SE-DEA)模型

在 DEA 模型的分析结果中, 通常会出现多个 DMU 被评价为有效的情况。这些有效 DMU 的效率高低无法进一步区分。1993 年, Anderson 和 Petersen 提出了“超效率”模型(Super Efficiency Model)。超效率模型的核心就是将评价 DMU 从参考集中剔除, 因此有效 DMU 的效率值一般会大于 1, 从而可以对有效 DMU 进行区分。如图 1, 在计算 C 点的效率时将其排除于参考集之外, 则前沿面变为虚线部分, 此时的 C 点的效率变为“ OC'/OC ”, 而 B' 点的效率值保持不变仍为“ OB/OB' ”。

SE-DEA 模型可表示为:

$$\begin{aligned}
& \min \theta \\
s.t. \quad & \sum_{\substack{j=1 \\ j \neq k}}^n \lambda_j x_{ij} \leq \theta x_{ik} \quad i=1,2,\dots,m; j=1,2,\dots,n(j \neq k); \lambda \geq 0 \\
& \sum_{\substack{j=1 \\ j \neq k}}^n \lambda_j y_{rj} \geq y_{rk} \quad r=1,2,\dots,q; \\
& \sum_{\substack{j=1 \\ j \neq k}}^n \lambda_j = 1
\end{aligned} \tag{24}$$

随着 SE-DEA 的兴起与应用, 对于多个 DMU 有效评价排序成为可能, 本文基于此通过构建 SE-DEA 模型, 对各区域碳排放绩效进行了评价。

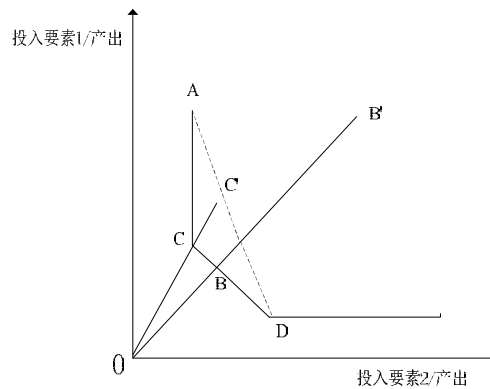


图 5 SE-DEA 模型

4.4.3 模型的求解

利用超效率 DEA 模型进行各区域效率排名，以得到的 110 个区域的数据进行分析。

投入量为： 总货运量(吨)、区域面积(平方米)

产出量为： 交通拥堵指数

利用 matlab 编程求解，求解程序和结果见附件 3，效率值排序：

表 5 区域效率排名

排名	区域编号	效率值	排名	区域编号	效率值	排名	区域编号	效率值
1	888	3.8496144	39	792	0.520303678	77	804	0.308013982
2	897	2.885166	40	830	0.512026259	78	815	0.304832795
3	803	1.2797922	41	800	0.509997891	79	878	0.304639389
4	896	1.1638243	42	855	0.505618506	80	834	0.301953777
5	832	1.0260281	43	882	0.488673225	81	794	0.300639455
6	870	0.9488263	44	900	0.485373222	82	813	0.300016114
7	817	0.8898592	45	854	0.483401424	83	879	0.299481235
8	892	0.8880555	46	810	0.48317723	84	895	0.295208608
9	853	0.8877956	47	827	0.469530152	85	893	0.293718495
10	894	0.8178688	48	795	0.463200176	86	865	0.274210469
11	816	0.8023539	49	819	0.457511459	87	860	0.272013649
12	883	0.7585575	50	838	0.453820704	88	849	0.271604681
13	825	0.7568447	51	839	0.446753791	89	845	0.267138836
14	829	0.7532832	52	802	0.438933761	90	851	0.267102069
15	824	0.7496116	53	806	0.421612755	91	866	0.262026104
16	833	0.7442404	54	868	0.417037	92	852	0.260562724
17	862	0.7295982	55	822	0.413085874	93	856	0.257622431
18	828	0.7181125	56	826	0.409529542	94	899	0.243962369
19	844	0.7135279	57	823	0.407691477	95	858	0.243806601
20	837	0.6943999	58	805	0.407358043	96	861	0.242929158
21	874	0.6601055	59	809	0.403796828	97	890	0.232069977
22	848	0.6590785	60	796	0.399130839	98	814	0.228424824
23	821	0.6483181	61	886	0.395882008	99	891	0.227311387
24	871	0.6460543	62	798	0.395284241	100	836	0.221983479
25	799	0.6411639	63	820	0.388980378	101	808	0.210370827
26	807	0.6339361	64	847	0.376567983	102	793	0.200212031
27	869	0.6239345	65	843	0.372894951	103	880	0.195679341
28	877	0.6105267	66	887	0.370373764	104	797	0.195572974
29	884	0.6100282	67	811	0.364172107	105	842	0.19455659
30	857	0.598156	68	840	0.359978279	106	867	0.182730445
31	864	0.5955675	69	889	0.359239188	107	831	0.16043627
32	835	0.5851386	70	846	0.353461401	108	841	0.157636237
33	863	0.5827643	71	801	0.346742424	109	898	0.155793561
34	885	0.5708269	72	859	0.346561524	110	881	0.103896873
35	791	0.5691414	73	875	0.34305924			
36	818	0.5600191	74	872	0.323063161			

37	850	0.5404994	75	873	0.314332368
38	812	0.5217084	76	876	0.313550824

与节点选择模型计算所得结果大致相同,不过有个别节点在顺序上存在出入,但整体上一二级节点的选择并没有偏差,间接地说明了节点选择模型的正确性。

5 问题二模型的建立与求解

5.1 地下物流网络模型

本问题为复杂网络问题,在问题一中已经确定一、二级节点的基础上,完成地下物流网络规划工作,作为地下物流规划者来说,应从对于交通拥堵的缓解程度和降低物流成本的角度出发选择一、二级节点的数目及位置,从而确定该区域地下物流网络节点群。

5.1.1 地下物流网络模型的提出

地下物流系统的网络形式多样且随区域特征变化,本文总结了地下物流系统常见的几种布局形式如表 6 所示。结合一、二级节点和布局结构的特征以及迄今为止世界上尚未有成功经验可以借鉴,采用开发难度较小连接较多节点的树状结构进行设计。

表 6 地下物流系统常见的布局形式

布局结构	优点	缺点
栅格布局	连接的节点比较多,网络的运输容量和连通度比较高	投资费用和开发难度相对较大
树状布局	投资费用和开发难度小,可以连接到较多的节点	对节点有一定的选择性,连通度较低
混合布局	连接的节点较多,网络的连通性较好	投资费用和开发难度较高
线状布局	投资费用和开发难度小,对于节点的选择性比较大	连通能力和运输容量相对较低
环状布局	开发难度和费用不高	连接的节点数量较少,任何一条线路的出现问题将影响整个网络

5.1.2 地下物流网络设计双层优化模型的建立

双层优化模型 Bracken J 和 Mc Gill J T 于 1973 年最早提出了多层规划的概念。双层规划广泛应用在交通因素、经济调度、配电系统优化规划等领域已有研究报道。数学上双层优化可描述为:

$$\min J_1 = F(x, y_1, y_2, y_3, \dots, y_m) \quad (25)$$

$$s.t. \quad G(x) \leq 0$$

$$\min J_2 = f(x, y_1, y_2, \dots, y_m) \quad (26)$$

$$s.t. \quad g(x, y_1, y_2, \dots, y_m) \leq 0$$

本文采用双层优化规划方法对地下物流网络选取进行优化,上层为拥堵改善模块,用于寻找拥堵情况改善的最优程度,包括交通拥堵指数、地上货物运输量、地下货物运输量。下层为物流成本优化模块,用于计算物流成本的优化,包括地下物流线路长度、地上车辆运输货物量。双层优化含有两个层次,上层决策结果一般会影响下层目标和约束条件,而下层则将决策结果反馈给上层,从而实现上

下层决策的相互作用，如图 6 所示为本文双层优化逻辑图。

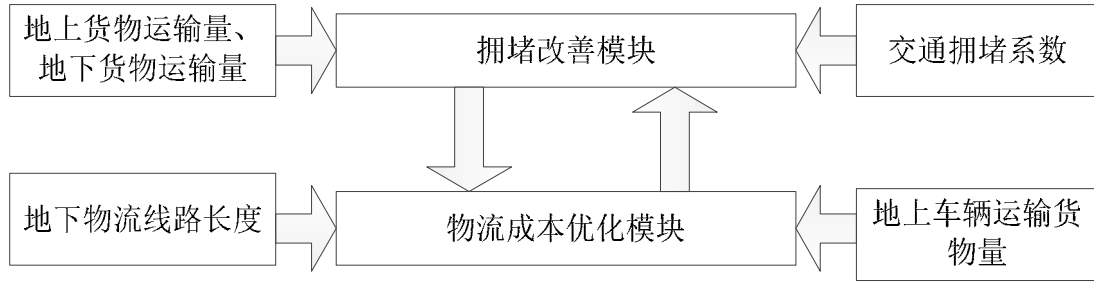


图 6 双层优化逻辑图

5.2 上层拥堵改善模型

5.2.1 指数定义

(1)地下一级节点货运量指数 $FTIY_{yi}$:

定义：地下一级节点货运量指数 $FTIY_{yi}$ 表示为一级节点 yi 与一级节点中地下货运量最大的百分比：

$$FTIY_{yi} = \frac{S_{yi}}{S_{yi\max}} \quad (27)$$

(2)地下二级节点货运量指数 $FTIT_{ti}$:

定义：地下二级节点货运量指数 $FTIT_{ti}$ 表示为二级节点 ti 与二级节点中地下货运量最大值的百分比：

$$FTIT_{ti} = \frac{S_{ti}}{S_{ti\max}} \quad (28)$$

5.2.2 目标函数

选取地下物流系统中拥堵改善性能作为评价指标：

$$\max F_{up} = F_1 + F_2 \quad (29)$$

(1)交通拥堵指数改善情况：

$$F_1 = \sum_{i=1}^n [TPI_i(0) - TPI_i(t)] \quad (30)$$

式中， $TPI_i(t)$ 表示第 i 个节点在第 t 天的交通拥堵指数， $TPI_i(0)$ 表示初始当天的交通拥堵指数。

(2)货运量分散程度：

由于地下物流系统的造价和运营成本较高，因此应要求地下节点的货运容量应充分被利用，即各级节点的货运量指数应尽可能大：

$$F_2 = \sum_{yi=1}^{yn} FTIY_{yi} + \sum_{ti=1}^{tn} FTIT_{ti} \quad (31)$$

既目标函数为：

$$\max F_{up} = \sum_{i=1}^n [TPI_i(0) - TPI_i(t)] + \sum_{yi=1}^{yn} FTIY_{yi} + \sum_{ti=1}^{tn} FTIT_{ti} \quad (32)$$

5.2.3 拥堵改善模型的约束条件

(1) 该区域的货运 OD 的交通拥堵指数将改善为交通畅通(0-2)，最少是基本

畅通(2-4)。

$$0 \leq TIP(t) \leq 4 \quad (33)$$

(2) 地下节点及通道内的货物每天要清仓。

$$SI_i(t) = SO_i(t) \quad (34)$$

式中, $SI_i(t)$ 表示地下节点 i 在第 t 天的进货量, $SO_i(t)$ 表示地下节点 i 在第 t 天的出货量。并且满足:

$$S_i(t) = SI_i(t) + SO_i(t) \quad (35)$$

5.3 下层物流成本优化模型

5.3.1 目标函数

选取地下物流系统中成本优化模型为评价指标:

$$\min F_{down} = F_3 + F_4 + F_5 \quad (36)$$

(1) 年均节点间线路造价折旧:

假设地下物流系统通道的设计年限为 100 年, 年综合折旧率为 1%。

$$F_3 = \left[\sum_{i=1, j=1}^n S_i(i, j) \times \lambda(i, j) + C_{yn} yn + C_m tn \right] \times Z \quad (37)$$

式中, $\lambda(i, j)$ 为地下物流系统的建造单位成本, 表示节点 i 到节点 j 的单位建造成本。 C_{yn} 表示第一节节点的造价, C_m 表示第二节节点的单位造价, Z 表示项目折旧。

现就 $\lambda(i, j)$ 进行分析: 若节点 i 和节点 j 中有园区节点, 则:

$$\lambda(i, j) = \begin{cases} 5 & \text{双向四轨 (10吨)} \\ 4 & \text{双向双轨 (10吨)} \\ 3.5 & \text{双向四轨 (5吨)} \\ 3 & \text{双向双轨 (5吨)} \end{cases} \quad (38)$$

若节点 i 和节点 j 中无园区节点, 则:

$$\lambda(i, j) = \begin{cases} 3 & \text{双向双轨 (5吨)} \\ 3.5 & \text{双向四轨 (5吨)} \end{cases} \quad (39)$$

(2) 货物运输成本:

假设每吨每公里的平均运输成本始终相等, 为 1 元/吨公里:

$$F_4 = \sum_{i=1, j=1}^n \frac{S_i(i, j) \times S_{dis}(i, j)}{2} \quad (40)$$

即目标函数为:

$$\min F_{down} = \left[\sum_{i=1, j=1}^n S_i(i, j) \times \lambda(i, j) + C_{yn} yn + C_m tn \right] \times Z + \sum_{i=1, j=1}^n \frac{S_i(i, j) \times S_{dis}(i, j)}{2} \quad (41)$$

5.3.2 拥堵改善模型的约束条件

(1) 一级节点的地面收发货量上限为 4000 吨。

$$SHY(yi) \leq 4000 \quad (42)$$

(2) 二级节点的地面收发货量上限为 3000 吨。

$$SHT(ti) \leq 3000 \quad (43)$$

(3) 通车的地下节点之间的距离受航班和车辆行驶速度的约束(单位: 米)。

$$S(i, j) \leq 6000 \quad (44)$$

综上，双层优化模型的表达式为：

$$\max F_{up} = \sum_{i=1}^n [TPI_i(0) - TPI_i(t)] + \sum_{yi=1}^{yn} FTIY_{yi} + \sum_{ti=1}^{tn} FTIT_{ti} \quad (45)$$

$$s.t. \quad 0 \leq TIP(t) \leq 4$$

$$SI_i(t) = SO_i(t)$$

$$\min F_{down} = \left[\sum_{i=1, j=1}^n S_i(i, j) \times \lambda(i, j) + C_{yn} yn + C_{tn} tn \right] \times Z + \sum_{i=1, j=1}^n \frac{S_i(i, j) \times S_{dis}(i, j)}{2} \quad (46)$$

$$s.t. \quad SHY(yi) \leq 4000$$

$$SHT(ti) \leq 3000$$

$$S(i, j) \leq 6000$$

5.4 地下物流网络模型求解方法

基于改进粒子群的双层规划求解算法上层首先给出决策，通过控制变量的值来引导下层，下层在上层的决策方案给定的情况下来寻找自己的最佳策略，并将自己的最佳策略通过变量回馈给上层；为了使自己的利益最大，上层再根据下层的反应调整自身的方案，重新做出决策，然后在上层和下层之间交互迭代，同步优化双层规划的上层和下层，最终解得双层规划模型的近似全局最优解。本文基于改进粒子群的双层规划求解算法的思想，在转运率变化不大的情况下，考虑优化网络来缓解交通拥堵状况以及降低下层物流成本，适当调整一、二级节点位置，从而选择合适的地下路线建立该地区的“地下物流系统”网络。

5.4.1 改进粒子群算法基本步骤

改进的粒子群算法的主要步骤分为 6 步^[7]：(1)对粒子群中粒子的速度 v_i 与位置 x_i 进行随机初始化， $i \in [1, m]$ ，其中 m 为粒子的数目，设置学习因子 c_1 和 c_2 的值和最大迭代次数 T_{max} 的值；随机产生满足下层规划模型约束条件的初始解。(2)设置粒子群中每个粒子的个体极值和整个粒子群的全局极值。(3)更新粒子的位置和速度；(4)如果此时算法满足预设的终止搜索的条件，则转到(6)否则转到(5)；(5)根据相应公式更新得到的局部最优解，利用改进的粒子群优化算法求出上层全局最优解的所对应的下层的局部最优价，转至(3)。(6)求得双层规划问题的最优解和并输出，计算出上层模型和下层模型对应于最优解的目标函数值。具体流程如图 7。

5.4.2 改进的粒子群算法求解过程

依据改进的粒子群算法对双层优化模型进行求解。

(1)搜寻区域中有 m 个粒子，第 i 个粒子的位置可以表示为 $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ ，速度可以表示为 $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ ，掌握粒子移动的距离和移动的方向。设置学习因子 c_1 和 c_2 的值和最大迭代次数 T_{max} 的值。

(2)设置粒子群中每个粒子的个体极值和整个粒子群的全局极值。

(3)利用下式更新粒子的速度和位置。

$$v_i(t+1) = v_i(t) + c_1 r_1 (pbest_j(t) - x_j(t)) + c_2 r_2 (gbest(t) - x_i(t)) \quad (47)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (48)$$

$$w = (wMax - wMin) \times \exp\left(-\frac{1}{1 + \frac{iterMax - iter}{iterMax}}\right) + wMin \quad (49)$$

式中 $c1$ 和 $c2$ 代表学习因子^[8]，通常取值为 2； $r1$ 和 $r2$ 为均匀分布在 $[0, 1]$ 之间的随机数； w 为惯性权重系数，通常选为一个合适的值，用来调节全局搜索和局部搜索之间的平衡，可以在迭代的过程中动态的调整 w 的取值。

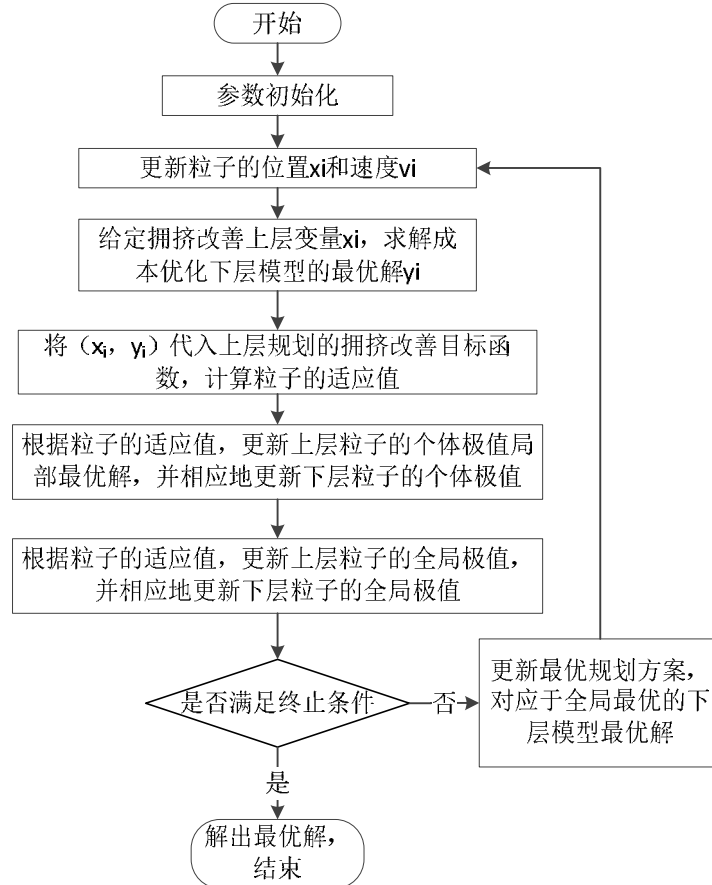


图 7 地下物流网络双层优化模型求解流程图

(4) 将上层模型的解即粒子 i 的位置 x_i 代入到下层模型，利用改进的粒子群算法求解下层模型的最优解 y_i 。

(5) 将 (x_i, y_i) 代入本文上层模型的适应度函数 $F(x, y)$ ，得到粒子的适应值 $F(x_i, y_i)$ 。

(6) 如果此时第 i 个粒子的适应值 $F(x_i, y_i)$ ，优于当前个体极值的适应值，则该粒子的位置 x_i 更新为 $pbest$ ，下层规划对应的最优解相应的更新为 y_i 。

(7) 如果此时第 i 个粒子的适应值 $F(x_i, y_i)$ ，优于当前全局极值的适应值，则该粒子的位置 x_i 更新 $gbest$ ，下层规划对应的最优解相应的更新为 y_i 。

(8) 如果此时算法满足预设的终止搜索的条件，则求得双层规划问题的最优解 $gbest$ 和 $ybest$ 并输出，并且求出上层模型和下层模型对应于最优解的目标函数值。否则利用下式更新 $gbest$ 。然后重新带入步骤(3)更新粒子的速度和位置进行求解。

$$gbest = gbest \times (1 + \eta \times 0.5) \quad (50)$$

式中 η 是变异因子，服从标准正态分布，以此来协助粒子摆脱局部最优点，

尽快找到全局最优解。

(9)直到求出最优解。*gbest* 和 *ybest* 并输出，并且求出上层模型和下层模型对应于最优解的目标函数值。

5.5 地下物流网络模型的求解

相关求解程序见附件 4。

5.5.1 网络构成

通过双层优化算法得到的优化网络如图 8 所示，其中粗红线表示双向四轨（双洞）（10 吨）、细红线表示双向双轨（双洞）（10 吨）、粗蓝线表示双向四轨（双洞）（5 吨）、细蓝线表示双向双轨（双洞）（5 吨）。

从图中可以看出，由园区至最近的一级节点的线路为双向四轨（双洞）（10 吨）线路，说明园区对于这些一级节点的依赖较大。节点 888 和 891 以及 817 和 827 之间的线路为双向四轨（双洞）（5 吨）线路，表明在各自区域内该一二级节点之间的关联度较强，对应的二级节点可称为“准一级节点”。

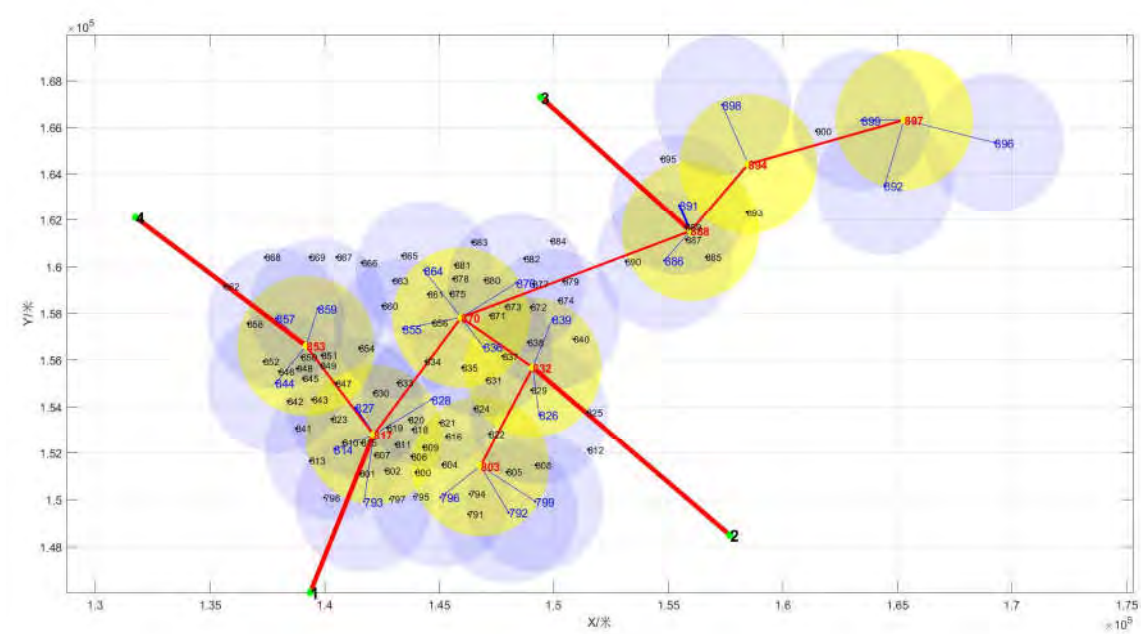


图 8 地下物流网络图

5.5.2 各节点实际货运量

由于地下物流网络的建设，各级节点均能够利用地下物流网络进行货物的转运工作，根据结果来看，绝大部分节点地下货运量明显高于地面货运量。

表 7 一级节点实际货运量情况

区域划分	一级节点	地面（单位：吨）		地下（单位：吨）	
		进货量	出货量	进货量	出货量
区域 1	894	1633.843	1871.288	4533.589	4296.144
区域 2	888	1036.591	2176.561	3651.26	2511.29
区域 3	803	2589.635	1263.566	5658.383	6984.452
区域 4	817	1782.565	2363.989	4869.631	4288.207

区域 5	870	2156.898	4156.393	3856.231	1856.736
区域 6	832	1569.787	2569.633	4289.189	3289.343
区域 7	897	2728.561	1056.451	4325.058	5997.168
区域 8	853	2710.378	2051.691	4278.553	4937.24

表 8 二级节点实际货运量

区域划分	二级节点	地面（单位：吨）		地下（单位：吨）		实际总的货运量
		进货量	出货量	进货量	出货量	
区域 1	898	1534.207	1174.304	3201.405	3561.308	9471.224
区域 2	891	743.211	1456.709	2002.598	1289.1	5491.618
	886	645.408	1357.905	1865.495	1152.998	5021.806
区域 3	796	803.409	1078.507	1236.957	961.859	4080.732
	799	606.507	1127.403	1497.874	976.978	4208.762
	792	803.786	908.405	1597.369	1492.75	4802.31
区域 4	814	653.852	790.436	1382.602	1246.018	4072.908
	793	597.346	845.205	1258.725	1010.866	3712.142
	827	774.501	832.783	1582.371	1524.089	4713.744
	828	645.832	642.743	1370.274	1373.363	4032.212
	864	684.381	803.284	1532.562	1413.659	4433.886
区域 5	855	842.104	931.421	1626.301	1536.984	4936.81
	836	742.194	732.354	1208.303	1218.143	3900.994
	876	843.195	612.363	1183.592	1414.424	4053.574
区域 6	826	1363.832	1455.749	1429.501	1337.584	5586.666
	839	987.42	1681.213	2042.501	1348.708	6059.842
区域 7	892	1472.391	1133.193	1128.371	1467.569	5201.524
	899	1891.323	928.39	1837.371	2800.304	7457.388
	897	1381.201	1037.382	1231.582	1575.401	5225.566
区域 8	844	824.281	720.301	1720.873	1824.853	5090.308
	857	812.62	802.479	1593.928	1604.069	4813.096
	859	920.358	932.204	1287.275	1275.429	4415.266

5.5.3 各通道的实际流量

各通道的流量实际反映了地下物流系统网络对于交通拥挤的缓解程度。通过表 9 可以看出大部分的通道的地下物流量高于地面流量，说明缓解交通拥挤的问题上地下物流网络起到了至关重要的作用。

表 9 各通道的实际流量

节点	相连线路节点	地面流量（单位：吨）	地下流量（单位：吨）	总流量（单位：吨）
	898	1708.511	3201.405	4099.405
894	888	359.203	678.903	1566.903
	897	985.306	684.863	1581.863
	891	1199.92	2002.598	2893.598
888	886	1003.313	901.675	1787.675
	894	960.203	846.903	1740.903
	870	573.394	768.346	1638.346
803	796	1342.234	1236.957	2032.957
	799	2342.415	1263.231	2062.231
	792	2881.355	1154.321	1946.321
	832	1243.385	987.257	1819.257
817	814	984.234	1382.602	2196.602
	793	892.345	1258.725	2051.725

	827	1224.345	1582.371	2409.371
	828	682.352	1370.274	2198.274
	870	781.234	984.347	1854.347
	853	824.245	678.246	1531.246
	864	576.245	1532.562	2396.562
	855	842.104	1626.301	2481.301
	836	742.194	1208.303	2044.303
870	876	843.195	1183.592	2059.592
	888	789.352	579.041	1467.041
	817	539.156	632.893	1449.893
	832	783.466	625.974	1457.974
	826	1363.832	1429.501	2255.501
832	839	987.42	2042.501	2881.501
	803	568.896	884.231	1687.231
	870	672.357	693.361	1563.361
	892	1472.391	1128.371	2020.371
897	899	891.323	1837.371	2736.371
	896	881.201	1231.582	2127.582
	894	678.4	874.146	1768.146
	844	824.281	1720.873	2564.873
853	857	812.62	1593.928	2450.928
	859	920.358	1287.275	2146.275
	817	976.324	631.421	1448.421

5.5.4 一级节点转运率变化情况

经过地下物流系统网络的设计后，一级节点的转运率均得到了一定程度的降低，说明一级节点转运率随着网络模型的构建而降低，从另一个角度说明网络构建的科学性。

表 10 一级节点转运率

距离物流园区最近的一级节点	一级节点转运率	变化率
817	33.12%	-2.93%
803	27.68%	-9.01%
888	28.99%	-0.51%
853	56.98%	-26.86%

5.5.5 地下物流网络折旧及运营成本

根据双层优化模型对网络进行优化后得到的年建造折旧及运营成本为 5,756 533,247 元。

6 问题三模型的建立与求解

6.1 地下物流网络改进模型的建立

本问题是网络优化问题，考察通过全局的角度如何进行网络节点和网络图的优化问题，地下物流线路改进的目标是提高网络利用率和较少交通拥堵以及各环

节的费用总和最低。

6.2 地下物流网络改进指标

(1)地下物流网络利用率
节点网络饱和度:

$$PGB_i = \frac{S_i(t)}{S_{\max i}} \quad (51)$$

线路饱和度:

$$LGB_{i,j} = \frac{S(i,j)}{S_{\max}(i,j)} \quad (52)$$

地下网络利用率:

$$TUO = \sum_{i=1, j=1}^n (PGB_i + LGB_{i,j}) \quad (53)$$

(2)交通拥堵指数改善情况:

$$F_1 = \sum_{i=1}^n [TPI_i(t) - TPI_i(t + \xi)] \quad (54)$$

式中, $TPI_i(t)$ 表示第 i 个节点在第 t 天的交通拥堵指数, $TPI_i(t + \xi)$ 表示网络优化后的交通拥堵指数。

(3)线路优化前后造价变化:

$$\Delta C_{grid} = \sum_{i=1, j=1}^n \{S_i(i,j) \times [\Delta \lambda(i,j)] + C_{yn} \Delta yn + C_m \Delta tn\} \quad (55)$$

6.3 地下物流网络改进模型

地下物流网络的建立是为了解决地上交通拥挤问题, 因此, 对于地下物流网络是否需要改进的依据是能够在满足经济性的前提下解决地上交通拥挤等问题。将地下物流网络改进模型分为三个子模型, 即地下物流网络满意度模型、地下物流网络抗风险检验模型、地下物流网络调整模型。

6.3.1 地下物流网络满意度模型

地下物流网络满意度模型为评价模型, 意在判别已知地下物流网络模型在解决交通拥挤问题上的作用以及网络整体的利用程度的计算。

地下物流网络满意度指数 $FTGW$, 该指数由地下网络利用率、交通拥堵指数改善情况两个指标构成:

$$FTGW = q_1 TUO + q_2 F_1 \quad (56)$$

式中, q_1 、 q_2 为相应指标的权重。

6.3.2 地下物流网络抗风险检验模型

地下物流网络模型抗风险检验模型为检验模型, 意在判断地下物流网络模型的抗风险能力。

最大通道中断数 $n_{bd \max}$: 已知地下物流网络最大允许的中断线路条数。

一级节点允许净增进货量 ACC_{yi} : 一级节点能够允许的最大净增进货量。

一级节点允许净增出货量 ACO_{yi} ：一级节点能够允许的最大净增出货量。

二级节点允许净增进出货量 ACC_{ti} ：二级节点能够允许的最大净增进出货量。

二级节点允许净增出货量 ACO_{ti} ：二级节点能够允许的最大净增出货量。

定义：地下物流网络抗风险指数 $FTAR$

$$FTAR = m_1 n_{bd \max} + m_2 ACC_{yi} + m_3 ACO_{yi} + m_4 ACC_{ti} + m_5 ACO_{ti} \quad (57)$$

指数越大，则表示系统抗风险能力越强。

6.3.3 地下物流网络调整模型

地下物流网络调整模型为网络优化模型，意在通过网络进行满意度和抗风险能力的优化。基于双层优化建立地下物流网络调整模型，上层为满意度优化模块，下层为抗风险能力优化模块。

(1)上层满意度调整

目标函数：
$$\max F_5 = q_1 TUO + q_2 F_1 \quad (58)$$

拥堵改善模型的约束条件：

a. 该区域的货运 OD 的交通拥堵指数将改善为交通畅通(0-2)，最少是基本畅通(2-4)。

$$0 \leq TIP(t) \leq 4 \quad (59)$$

b. 地下节点及通道内的货物每天要清仓。

$$SI_i(t) = SO_i(t) \quad (60)$$

式中， $SI_i(t)$ 表示地下节点 i 在第 t 天的进出货量， $SO_i(t)$ 表示地下节点 i 在第 t 天的出货量。并且满足：

$$S_i(t) = SI_i(t) + SO_i(t) \quad (61)$$

c. 一级节点的地面收发出货量上限为 4000 吨。

$$SHY(yi) \leq 4000 \quad (62)$$

d. 二级节点的地面收发出货量上限为 3000 吨。

$$SHT(ti) \leq 3000 \quad (63)$$

e. 通车的地下节点之间的距离受航班和车辆行驶速度的约束(单位：米)。

$$S(i, j) \leq 9800 \quad (64)$$

(2)下层抗风险能力优化

目标函数：

$$F_6 = m_1 n_{bd \max} + m_2 ACC_{yi} + m_3 ACO_{yi} + m_4 ACC_{ti} + m_5 ACO_{ti} \quad (65)$$

下层抗风险能力优化的约束条件：

a. 最大通道中断数约束：

$$0 \leq n_{bd \max} \leq 2n \quad (66)$$

b. 一级节点允许净增进出货量约束：

$$0 \leq ACC_{yi} + S_{yi} \leq 4000 \quad (67)$$

c. 一级节点允许净增出货量约束：

$$0 \leq ACO_{yi} + S_{yi} \leq 4000 \quad (68)$$

d. 二级节点允许净增进出货量约束：

$$0 \leq ACC_{ti} + S_{ti} \leq 3000 \quad (69)$$

e. 二级节点允许净增出货量约束：

$$0 \leq ACO_{ti} + S_{ti} \leq 3000 \quad (70)$$

综上，双层优化模型的表达式为：

$$\begin{aligned} \max F_5 &= q_1 TUO + q_2 F_1 \\ \text{s.t.} \quad & 0 \leq TIP(t) \leq 4 \end{aligned} \quad (71)$$

$$\begin{aligned} & SI_i(t) = SO_i(t) \\ & SHY(yi) \leq 4000 \\ & SHT(ti) \leq 3000 \\ & S(i, j) \leq 9800 \\ \max F_6 &= m_1 n_{bd \max} + m_2 ACC_{yi} + m_3 ACO_{yi} + m_4 ACC_{ti} + m_5 ACO_{ti} \\ \text{s.t.} \quad & 0 \leq n_{bd \max} \leq 2n \\ & 0 \leq ACC_{yi} + S_{yi} \leq 4000 \\ & 0 \leq ACO_{yi} + S_{yi} \leq 4000 \\ & 0 \leq ACC_{ti} + S_{ti} \leq 3000 \\ & 0 \leq ACO_{ti} + S_{ti} \leq 3000 \end{aligned} \quad (72)$$

6.4 地下物流网络改进模型的求解

6.4.1 地下物流网络满意度模型的求解

地下物流网络满意度的求解本身并不需要对应的算法求解，但考虑需要对优化后的地下网络进行满意度的重计算直至两者之差小于 0.1 时便输出结果，有必要对计算满意度之前进行优化校验，即该网络是否还需优化，若仍出现路线的重新优化，则对此次满意度计算结果进行标记，防止出现偶然因素导致满意度前后之差小于 0.1，导致满意度的误判。

Dijkstra 算法是解决路径优化的常见算法，其基本思路是假设每个点都有一对标号 (dj, pj) ，其中 dj 是从起点 s 到点 j 的最短路径的长度； pj 则是从 s 到 j 的最短路径中 j 点的前一点。求解从起源点 s 到点 j 的最短路径算法的基本过程如图 9 所示^[9]。

针对经典 Dijkstra 路径优化算法在大规模网络计算中存在时间复杂度难题，改进其对未标记节点遍历过程，直接寻求在遍历范围和方向上趋向目标节点，使得搜索过程不必全部遍历或只较少地遍历未标记结点，将时间复杂度从 $O(n^2)$ 降低为 $O(n)$ ，提高算法的运行效率。

整个算法循环只有一次，而且，如果目的点不是最后一个标记点的话，算法循环要少于 n 次，使得时间复杂度大大降低，提高了运行效率，由于目标节点为结束标志，因此算法准确性也得到保证。

6.4.2 地下物流网络抗风险检验模型

模拟退火算法的思想来源于对固体退火降温过程的模拟。即将固体加热至很高温度，再让其慢慢地冷却。基于对固体退火过程的模拟，模拟退火算法用冷却进度表来控制算法的进程，使算法在控制参数慢慢降温并趋于零时，求得组合优化问题的相对全局最优解^[10]。

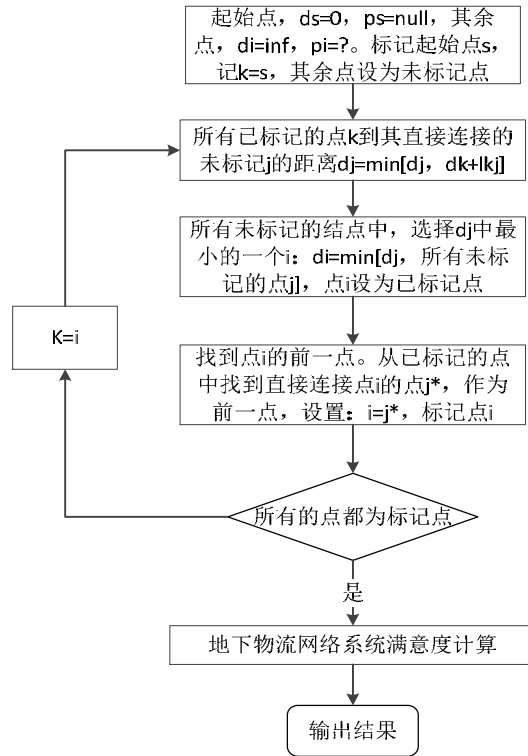


图 9 满意度计算前优化校验流程

(1) 模拟退火算法基本步骤

模拟退火算法的主要步骤分为 4 步: a.给定控制参数 T 的初始 T_0 、衰减函数、终值以及 Mapkob 链长度、迭代初始解 x_0 以及初始目标函数 $f(x_0)$; b.进行试探搜索; c.如果迭代终止条件满足,则算法结束,当前解为全局最优解, 否则继续; d.根据给定的衰减函数产生新的控制参数, 进入下一个温度点的平衡点寻优; 具体流程如下图。

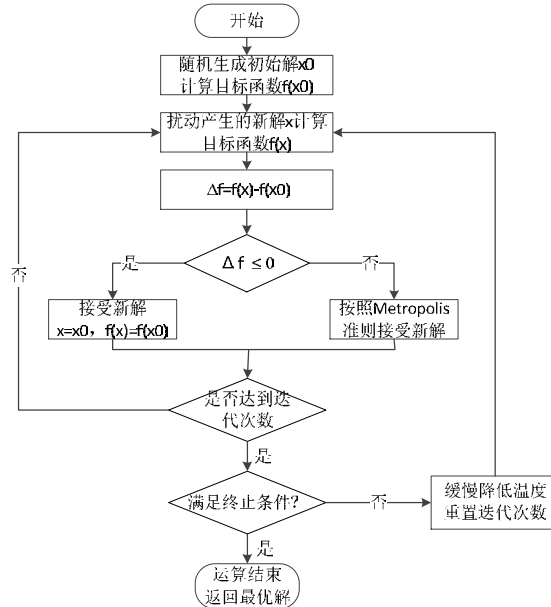


图 10 模拟退火算法流程图

模拟退火算法求解过程^[11]:

(1)给定控制参数 T 、衰减函数、终值以及 Mapkob 链长度 L_k , 迭代初始解 x_0 。以及初始目标函数 $f(x_0)$ 。

(2) 参数 $T = T(k)$ 时,按照如下过程作 L_k 次试探搜索:

1)根据当前解 X_k 的性质产生一随机向量 Z_k , 从而得到一当前解邻域的新的试探点 X'_k 。

$$X'_k = \begin{cases} X_k + Z_k, & \text{对于连续变量} X \\ X_{(k+m)}, & \text{对于离散变量} X \end{cases} \quad (73)$$

其中为离散变量的取值序列, k 为当前解的离散位置

2)产生一个在区间(0, 1)上均匀分布的随机数 θ , 计算出在给定当前迭代点 X_k 和温度 T_k 下与 Metropolis 的接受准则相对应的转移概率 P 。

$$P = \begin{cases} 1, & \text{当 } f(X'_k) < f(X_k) \\ \exp\left(\frac{f(X'_k) - f(X_k)}{T_k}\right), & \text{当 } f(X'_k) \geq f(X_k) \end{cases} \quad (74)$$

如果 $\theta < P$ 则接受新解, $X_k = X'_k, f(X_k) = f(X'_k)$, 否则当前解不变。

3)试探搜索小于 L_k 次, 返回步骤(1), 否则进入步骤(3),

(3) 如果迭代终止条件满足, 则算法结束,当前解为全局最优解, 否则继续进行步骤(4)。

(4) 根据给定的温度衰减函数产生新的温度控制参数 T_{k+1} , 及 Mapkob 链长度 L_{k+1} , 转入步骤(2), 进入下一温度点的平衡点寻优。

6.4.3 地下物流网络调整模型

继续使用第二问改进的粒子群算法对网络调整双层优化模型进行求解^[12]。

(1)搜寻区域中有 m 个粒子, 第 i 个粒子的位置可以表示为 $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$, 速度可以表示为 $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$, 掌握粒子移动的距离和移动的方向。设置学习因子 c_1 和 c_2 的值和最大迭代次数 T_{\max} 的值。

(2)设置粒子群中每个粒子的个体极值和整个粒子群的全局极值。

(3)利用下式更新粒子的速度和位置。

$$v_i(t+1) = v_i(t) + c_1 r_1 (pbest_j(t) - x_j(t)) + c_2 r_2 (gbest(t) - x_i(t)) \quad (75)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (76)$$

$$w = (wMax - wMin) \times \exp\left(-\frac{1}{1 + \frac{iterMax - iter}{iterMax}}\right) + wMin \quad (77)$$

式中 c_1 和 c_2 代表学习因子, 通常取值为 2; r_1 和 r_2 为均匀分布在 $[0, 1]$ 之间的随机数; w 为惯性权重系数, 通常选为一个合适的值, 用来调节全局搜索和局部搜索之间的平衡, 可以在迭代的过程中动态的调整 w 的取值。

(4) 将上层模型的解即粒子 i 的位置 x_i 代入到下层模型, 利用改进的粒子群算法求解下层模型的最优解 y_i 。

(5)将 (x_i, y_i) 代入满意度调整上层模型的适应度函数 $F(x, y)$, 得到粒子的适应值 $F(x_i, y_i)$ 。

(6)如果此时第 i 个粒子的适应值 $F(x_i, y_i)$, 优于当前个体极值的适应值, 则该粒子的位置 x_i 更新为 $pbest$, 抗风险能力调整下层规划对应的最优解相应的更

新为 y_i 。

(7)如果此时第 i 个粒子的适应值 $F(x_i, y_i)$ ，优于当前全局极值的适应值，则该粒子的位置 x_i 更新 $gbest$ ，抗风险能力下层规划对应的最优解相应的更新为 y_i 。

(8)如果此时算法满足预设的终止搜索的条件，则求得双层规划问题的最优解 $gbest$ 和 $ybest$ 并输出，并且求出上层模型和下层模型对应于最优解的目标函数值。否则利用下式更新 $gbest$ 。然后重新带入步骤(3)更新粒子的速度和位置进行求解。

$$gbest = gbest \times (1 + \eta \times 0.5) \quad (78)$$

式中 η 是变异因子，服从标准正态分布，以此来协助粒子摆脱局部最优点，尽快找到全局最优解。

(9)直到求出最优解。 $gbest$ 和 $ybest$ 并输出，并且求出上层模型和下层模型对应于最优解的目标函数值^[13]。

6.4.4 三级子模型寻优过程

对于网络的优化来说，应用原始的网络数据进行满意度和抗风险验证模型的初值设定，对于调整后的模型有必要进行优化结果校验，再进行地下物流网络满意度的计算，因此有必要将优化验证模块归入地下网络满意度模型。

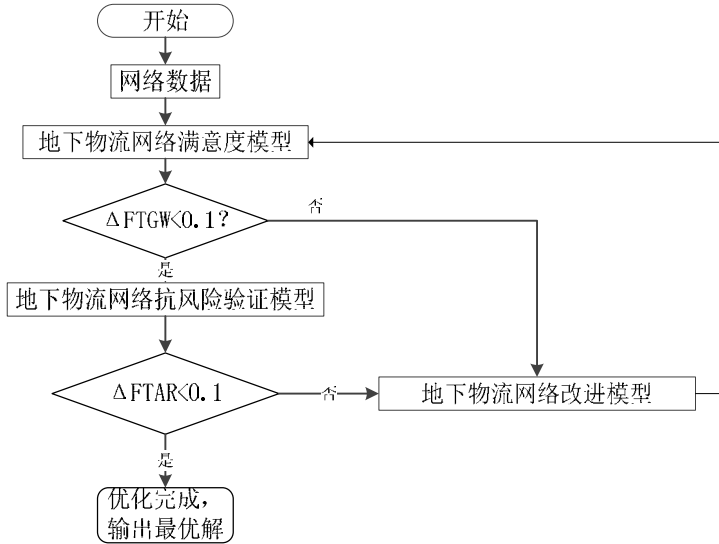


图 11 三级子模型寻优

6.5 三级子模型的求解

利用改进的 Dijkstra 算法、模拟退火算法、改进的粒子群算法对模型进行求解，相关程序见附件 6。

6.5.1 网络修改

根据仿真结果来看，对于一二问的基础网络有修改的必要，经过三层子模型的寻优后，得到如下的地下物流网络：

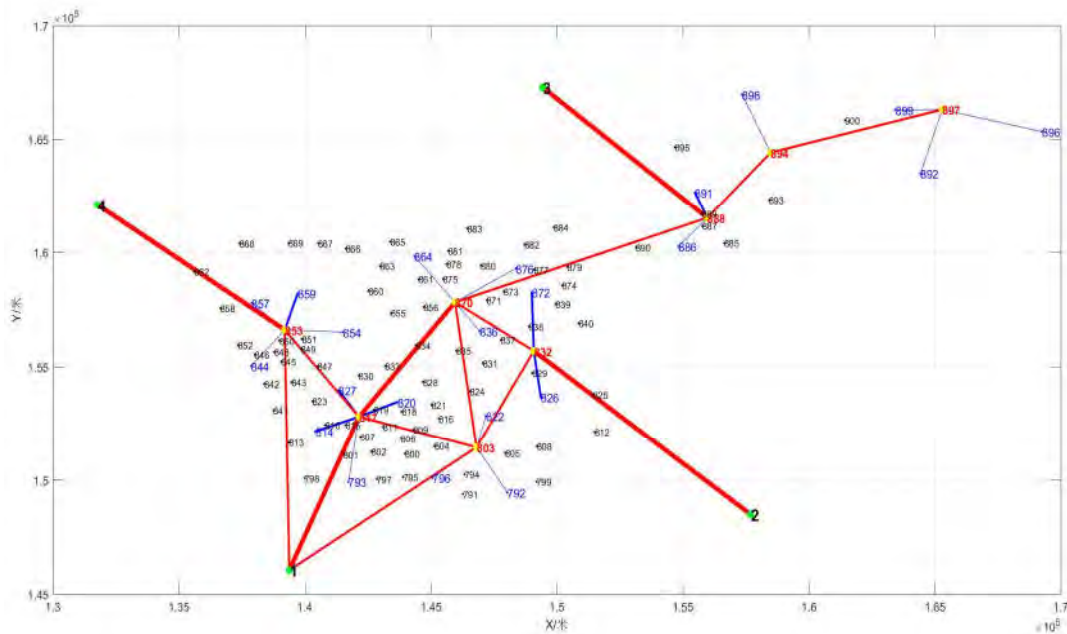


图 12 改进后地下物流网络图

通过对地下物流系统网络的对比发现，改进后的物流网络图主要是新增 1-853, 1-803, 817-803, 803-870 的双向双向双轨（双洞）（10 吨）、升级 870-817 为双向四轨（双洞）（10 吨）、升级 814-817, 853-859, 832-826 为双向四轨（双洞）（5 吨），与此同时也去掉二级节点 799, 855, 839 和 828，新增二级节点 822 和 854 采用双向双向双轨（双洞）（5 吨），820 和 872 采用双向四轨（双洞）（5 吨）。

6.5.2 修改后网络指标

改进后的地下物流网络抗风险指数为 6.523，网络满意度为 6.982,达到了优化的目的。

7 问题四模型的建立与求解

7.1 地下物流网络时序与动态优化问题分析

本问题是典型的动态规划问题，考察如何在规定的时间内逐步完成网络的规划，设计的主要目标仍是交通拥堵情况的改善程度最大化和建造及运营成本的最小化。

7.2 地下物流网络时序与动态优化模型指数定义

(1)在地下网络规划中影响网络质量优劣的因素，如：线路长度、线路通流量、交通拥挤指数等，这些因素影响线路建设行进时间或速度，从而在整个路径规划的过程中影响选择路径的优劣。

(2)决定该条地下道路能否通行的因素，如：两端节点的载重限制、运行车辆的时速限制、运行车辆载重限制等。

(3)地下网络建设期内地下通道的作用应有所体现，如：每年建设后对于交

通拥挤的缓解情况。

基于上述分析，进行指数的定义。

7.2.1 城市交通需求指数 $CTI(t)$

$$CTI(t) = \frac{\sum_{i=1}^{n_{ci}} S_i(t)}{\sum_{i=1}^n S_i(t-1)} \quad (65)$$

7.2.2 城市地下物流交通指数 $CFTI(t)$

$$CFTI(t) = \left[\frac{\sum_{yi=1}^{n_{cy}} S_{yi}(t)}{\sum_{yi=1}^{n_y} S_{yi}(t-1)} + \frac{\sum_{ti=1}^{n_{ct}} S_{ti}(t)}{\sum_{yi=1}^{n_y} S_{ti}(t-1)} \right] \quad (66)$$

7.2.3 地下物流网络利用指数 $TUO(t)$

节点网络饱和度：

$$PGB_i(t) = \frac{S_i(t)}{S_{\max i}} \quad (67)$$

线路饱和度：

$$LGB_{i,j}(t) = \frac{S_{i,j}(t)}{S_{\max i,j}(t)} \quad (68)$$

地下网络利用率：

$$TUO(t) = \sum_{i=1, j=1}^n (PGB_i(t) + LGB_{i,j}(t)) \quad (69)$$

7.2.4 每年交通拥堵指数动态改善情况 $TTIF(t)$

$$TTIF(t) = \sum_{i=1}^n [TPI_i(t) - TPI_i(t+1)] \quad (70)$$

7.3 地下物流网络时序与动态优化模型

地下物流网络时序与动态优化模型意在解决南京市仙林区近 30 年的交通需求增长问题，考虑 8 年建设工期的过程中，如何对南京市地下物流网络进行持续优化，并保证能够在建设前后能够真正意义上解决交通拥堵问题。本文采用动态规划的思想进行地下物流网络时序与动态优化模型的构建。

7.3.1 优化目标函数

对于地下网络动态规划，可以以地下的物流网络设计过程中的拥堵改善程度和物流线路造价以及地下网络利用指数为目标，则目标函数可为：

$$\max F_{time_best} = TTIF(t) + TUO(t) + \sum_{i=1, j=1}^{n_{ij}(t)} S_i(i, j) \times \lambda(i, j) + C_{ym} yn + C_m tn \quad (71)$$

7.3.2 约束条件

a. 城市交通需求指数每年基本上为 1.05:

$$CTI(t) = \frac{\sum_{i=1}^{n_{cj}} S_i(t)}{\sum_{i=1}^n S_i(t-1)} = 1.05 \quad (72)$$

b. 地下节点及通道内的货物每天要清仓。

$$SI_i(t) = SO_i(t) \quad (73)$$

式中, $SI_i(t)$ 表示地下节点 i 在第 t 天的进货量, $SO_i(t)$ 表示地下节点 i 在第 t 天的出货量。并且满足:

$$S_i(t) = SI_i(t) + SO_i(t) \quad (74)$$

c. 一级节点的地面收发货量上限为 4000 吨。

$$SHY(yi) \leq 4000 \quad (75)$$

d. 二级节点的地面收发货量上限为 3000 吨。

$$SHT(ti) \leq 3000 \quad (76)$$

e. 通车的地下节点之间的距离受航班和车辆行驶速度的约束(单位: 米)。

$$S(i, j) \leq 9800 \quad (77)$$

f. 每年可建设道路长度大致相等。

$$\sum_{i,j=1}^{n_{it}(t)} S_{dis}(i, j, t) = \frac{\sum_{i,j=1}^n S_{dis}(i, j)}{8} \quad (78)$$

综上所述, 动态过程的优化目标及约束条件为:

$$\max F_{time_best} = TTIF(t) + TUO(t) + \sum_{i=1, j=1}^{n_{it}(t)} S_i(i, j) \times \lambda(i, j) + C_{yn} yn + C_{in} tn \quad (79)$$

$$s.t. \quad CTI(t) = \frac{\sum_{i=1}^{n_{cj}} S_i(t)}{\sum_{i=1}^n S_i(t-1)} = 1.05$$

$$SI_i(t) = SO_i(t)$$

$$SHY(yi) \leq 4000$$

$$SHT(ti) \leq 3000$$

$$S(i, j) \leq 6000$$

$$\sum_{i,j=1}^{n_{it}(t)} S_{dis}(i, j, t) = \frac{\sum_{i,j=1}^n S_{dis}(i, j)}{8}$$

7.4 地下物流网络时序与动态优化模型的求解

一个多阶段决策过程最优化问题的动态规划模型通常包含以下要素。

(1) 地下物流网络规划阶段

地下物流网络规划阶段是对整个地下物流网络建设过程的自然划分。根据时间顺序和空间特征来划分阶段, 以便按阶段的次序解优化问题。设地下物流网络

规划阶段变量为 $k(k=1,2,3\ldots cony)$ ，其中整个地下物流网络的建设工期为 $cony$ 。

(2) 地下物流网络状态

地下物流网络状态(state)表示每个阶段开始时过程所处的自然状况。以交通拥堵改善情况和成本为状态变量集合。用 kx 表示第 k 阶段的状态变量。

$$kx = \left\{ TPI(k), \sum_{i=1, j=1}^n S_{i,j}(k) \times \lambda(i, j) + C_{yn} yn(k) + C_{tn} tn(k) \right\} \times Z + \sum_{i=1, j=1}^n \frac{S_{i,j}(k) \times S_{dis}(i, j)}{2} \quad (80)$$

根据过程演变的具体情况，地下物流网络系统状态变量应是一组离散变量。用状态变量矩阵 Kx 表示：

$$Kx = [x, 2x, \ldots, conyx] \quad (81)$$

(3) 地下物流网络决策^[14]

当一个阶段的状态确定后，可以作出各种选择从而演变到下一阶段的某个状态，这种选择手段称为决策，在最优控制问题中也称为控制。

描述决策的变量称决策变量，变量允许取值的范围称允许决策集合。用 $u(k)$ 表示第 k 阶段处于状态 kx 时的决策变量，它是 kx 的函数，用表示 kx 的允许决策集合。

(4) 地下物流网络策略

决策组成的序列称为策略^[15]。由初始状态 x_1 开始的全过程的策略记作 $p_{1n}(x_1)$ ，即

$$p_{1n}(x_1) = \{u_1(x_1), u_2(x_2), \cdots, u_n(x_n)\} \quad (82)$$

由第 k 阶段的状态 x_k 开始到终止状态的后部子过程的策略记作 $p_{kn}(x_k)$ ，即

$$p_{kn}(x_k) = \{u_k(x_k), \cdots, u_n(x_n)\}, k=1, 2, \cdots, n-1 \quad (83)$$

类似地，由第 k 到第 j 阶段的子过程的策略记作

$$p_{kj}(x_k) = \{u_k(x_k), \cdots, u_j(x_j)\} \quad (84)$$

可供选择的策略有一定的范围，称为允许策略集合，用 $p_{1n}(x_k), p_{kn}(x_k), p_{kj}(x_k)$ 表示这种演变规律。

(5) 地下物流系统状态转移方程

在确定性过程中，一旦某阶段的状态和决策为已知，下阶段的状态便完全确定。用状态转移方程表示

$$x_{k+1} = T_k(x_k, u_k), k=1, 2, \cdots, n \quad (85)$$

(6) 最优策略和最优轨线

使指标函数 $V_{k,n}$ 达到最优值的策略是从 k 开始的后部子过程的最优策略，记作 $p_{kn}^* = \{u_k^*, \cdots, u_n^*\}$ 。 p_{in}^* 是全过程的最优策略，简称最优策略。从初始状态 $x_1 (= x_1^*)$ 出发，过程按照 p_{in}^* 和状态转移方程演变所经历的状态序列 $\{x_1^*, x_2^*, \cdots, x_{n+1}^*\}$ 称最优轨线。

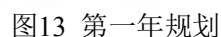
(8) 地下物流网络递归方程^[16]

$$\begin{cases} f_{n+1}(x_{n+1}) = 0, 1 \\ f_k(x_k) = \underset{u_k \in U_k(x_k)}{opt} \{v_k(x_k, u_k) \otimes f_{k+1}(x_{k+1})\}, k=n, \cdots, 1 \end{cases} \quad (86)$$

$$\begin{cases} f_0(x_1) = 0, 1 \\ f_k(x_{k+1}) = \underset{u_k \in U_{k+1}^*(x_{k+1})}{opt} \{v_k(x_{k+1}, u_k) \otimes f_{k-1}(x_k)\}, k = 1, \dots, n \end{cases} \quad (87)$$

通过对地下物流网络时序与动态模型优化方案进行求解,得出了每年进行地下物流网络时序与动态规划方案,现进行项目方案的整合工作,形成八年建设规划方案。

第一年：按照图13网络进行施工建设。



第二年：按照图14网络进行施工建设

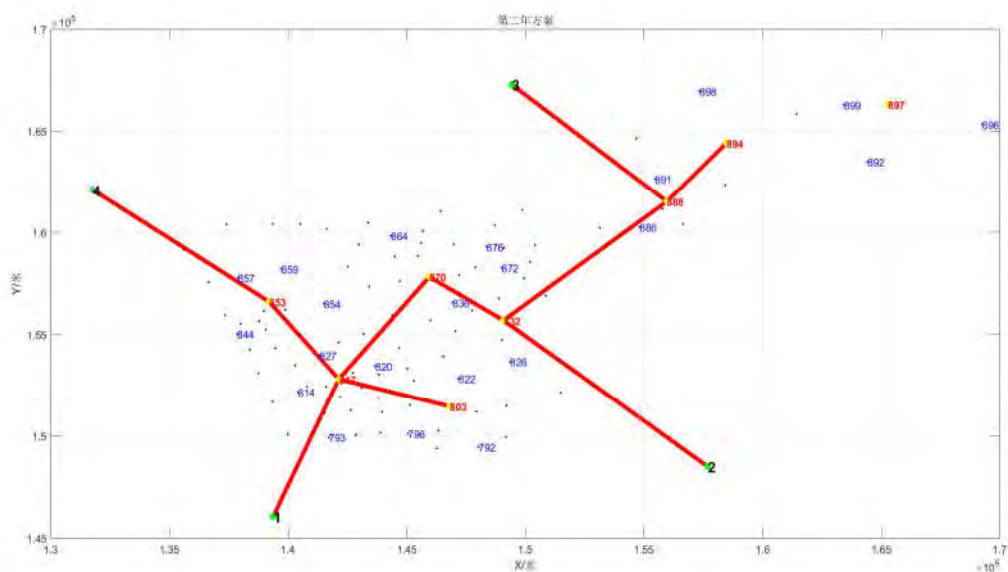


图14 第二年规划

具体修建：a.888-894：双向四轨（双洞）（10吨）；b.832-888：双向四轨（双洞）（10吨）；c.870-817：双向四轨（双洞）（10吨）；d.817-853：双向四轨（双洞）（10吨）；e.817-803：双向四轨（双洞）（10吨）；f.870-832：双向四轨（双洞）（10吨）。

第三年：按照图15网络进行施工建设

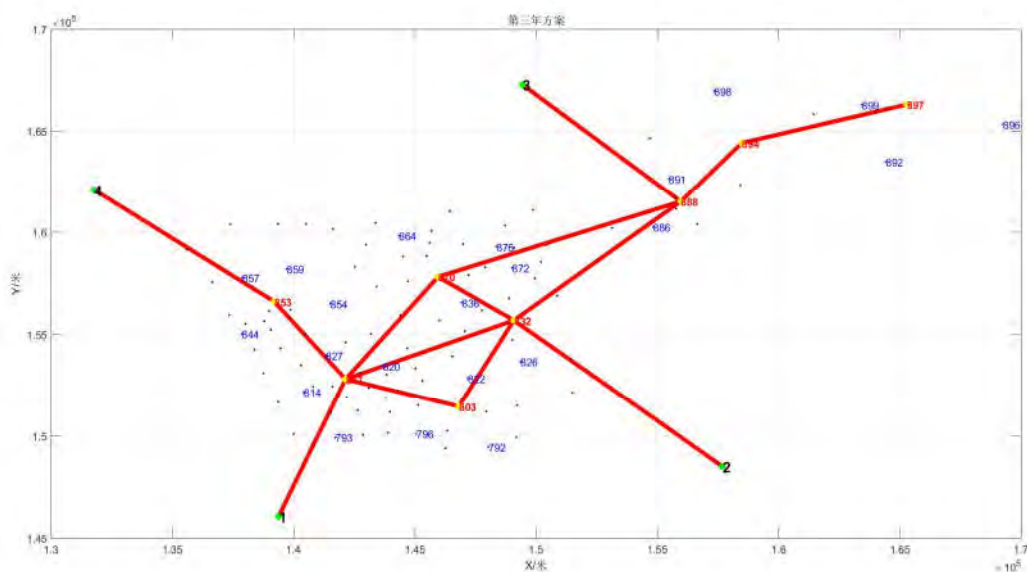


图15 第三年规划

集体修建：a.894-897：双向四轨（双洞）（10吨）；b.888-870：双向四轨（双洞）（10吨）；c.817-832：双向四轨（双洞）（10吨）；d.803-832：双向四轨（双洞）（10吨）。

第四年：按照图16网络进行施工建设

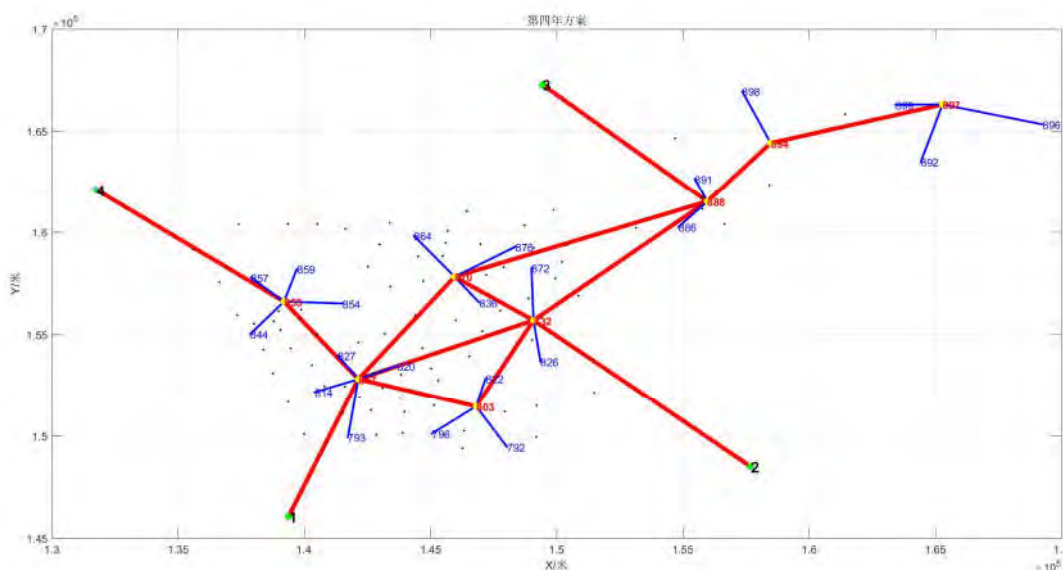


图16 第四年规划

集体修建：a.832-826：双向四轨（双洞）（5吨）；b.832-872：双向四轨（双洞）（5吨）；c. 853-859：双向四轨（双洞）（5吨）；d.853-857：双向四轨（双洞）（5吨）；e.853-854：双向四轨（双洞）（5吨）；f.853-844：双向四轨（双洞）（5吨）；g.817-827：双向四轨（双洞）（5吨）；h.817-814：双向四轨（双洞）（5吨）；i.817-820：双向四轨（双洞）（5吨）；j.817-793：双向四轨（双洞）（5吨）；k. 803-796：双向四轨（双洞）（5吨）；L.803-822：双向四轨（双洞）（5吨）；M.803-792：双向四轨（双洞）（5吨）；N. 870-864：双向四轨（双洞）（5吨）；O.870-836：双向四轨（双洞）（5吨）；P.870-876：双向四轨（双洞）（5吨）；q. 888-891：双向四轨（双洞）（5吨）；r.888-886：双向四轨（双洞）（5吨）；s. 894-898：双向四轨（双洞）（5吨）；t. 897-899：双向四轨（双洞）（5吨）；u.897-892：双向四轨（双洞）（5吨）；v.897-896：双向四轨（双洞）（5吨）。

第五年：按照图17网络进行施工建设

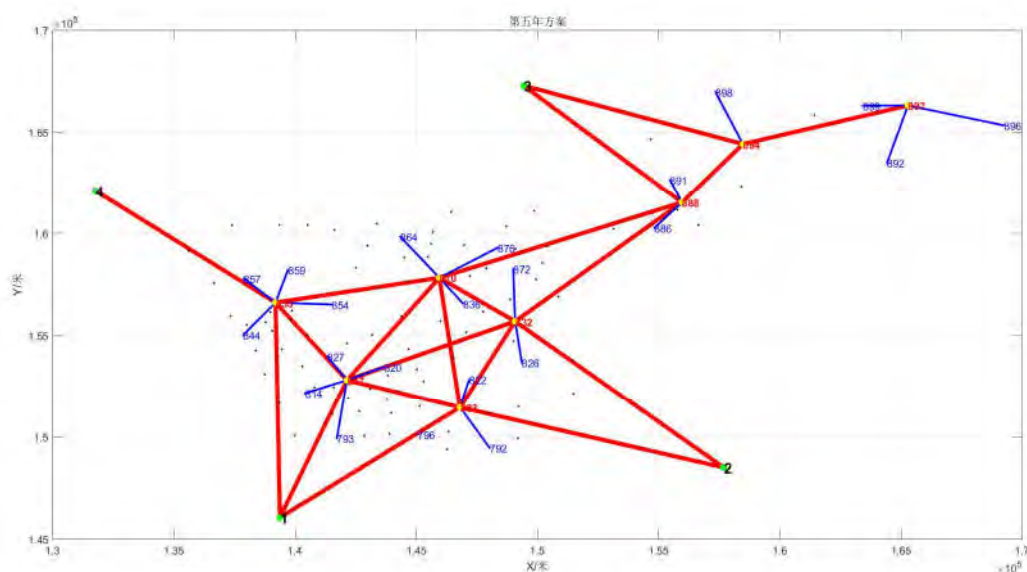


图17 第五年规划

集体修建：a.853-870：双向四轨（双洞）（10 吨）；b.870-803：双向四轨（双洞）（10 吨）；c.1-853：双向双轨（双洞）（10 吨）；d.1-803：双向双轨（双洞）（10 吨）；e.2-803：双向双轨（双洞）（10 吨）；f.3-894：双向四轨（双洞）（10 吨）。

第六年：按照图18网络进行施工建设

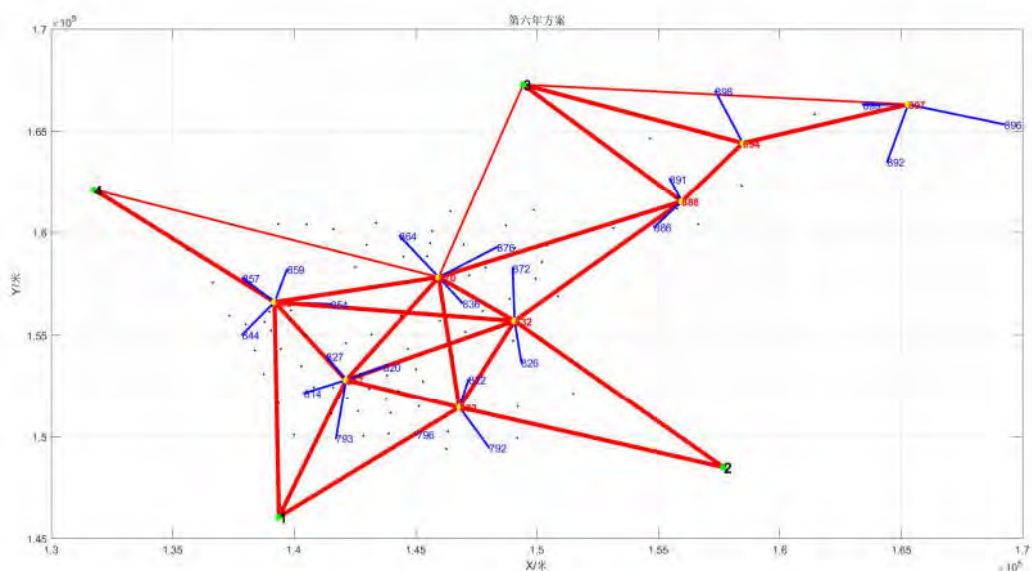


图18 第六年规划

具体修建：a.3-870：双向双轨（双洞）（10吨）；b.3-897：双向双轨（双洞）（10吨）；c.4-870：双向双轨（双洞）（10吨）；d.853-832：双向四轨（双洞）（10吨）。

第七年：按照图19网络进行施工建设

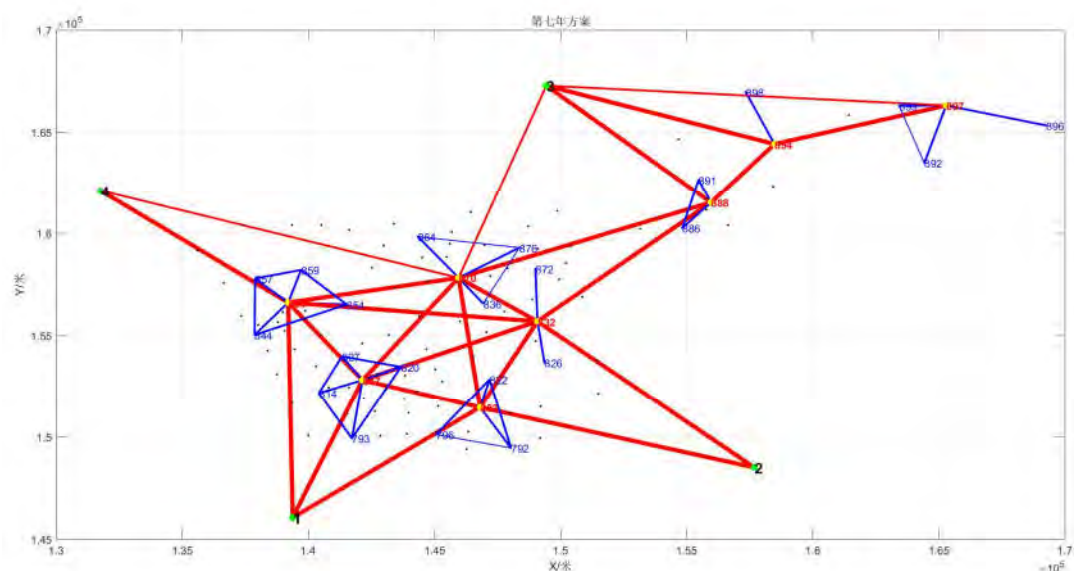


图19 第七年建设规划

具体修建:a.844-854：双向四轨（双洞）（5吨）；b.820-793：双向四轨（双洞）（5吨）；c.822-792：双向四轨（双洞）（5吨）；d.822-796：双向四轨（双洞）（5吨）；e.886-891：双向四轨（双洞）（5吨）；f.899-892：双向双轨（双洞）（5吨）；g.864-876：双向双轨（双洞）（5吨）；o.836-876：双向双轨（双洞）（5吨）。

洞) (5吨); p. 796-792: 双向双轨 (双洞) (5吨); q. 814-827: 双向四轨 (双洞) (5吨); r. 814-793: 双向四轨 (双洞) (5吨); s. 820-827: 双向四轨 (双洞) (5吨); t. 857-859: 双向四轨 (双洞) (5吨); u. 857-844: 双向四轨 (双洞) (5吨); v. 859-854: 双向四轨 (双洞) (5吨)。

第八年: 按照图20网络进行施工建设

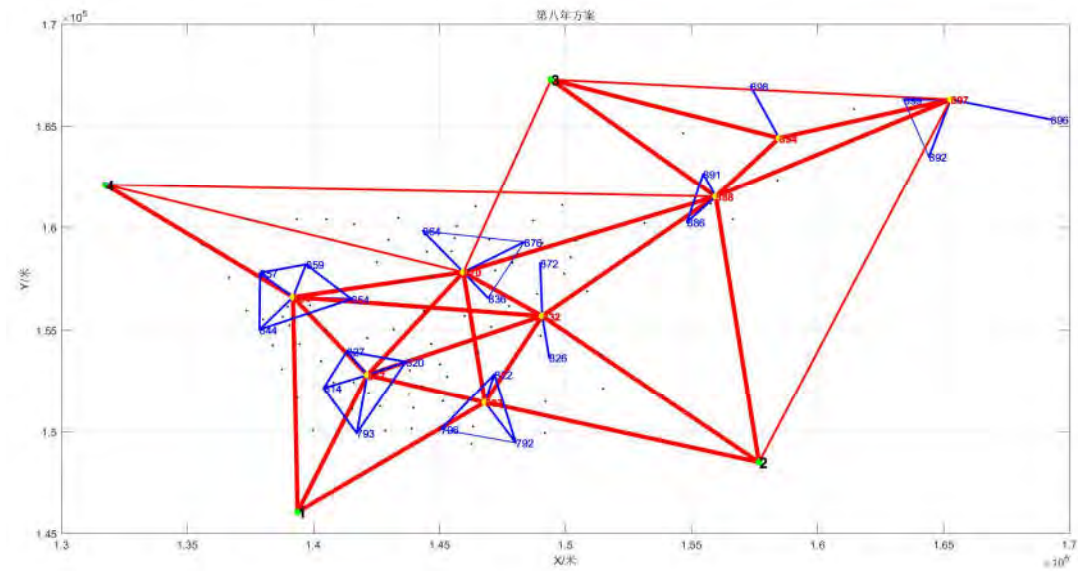


图20 第八年建设规划

具体修建: a. 4-888: 双向四轨 (双洞) (10 吨); b. 2-888: 双向双轨 (双洞) (10 吨); c. 2-888: 双向四轨 (双洞) (10 吨); d. 897-888: 双向四轨 (双洞) (10 吨)。

修建后对于交通拥挤程度的改善如图 21 所示

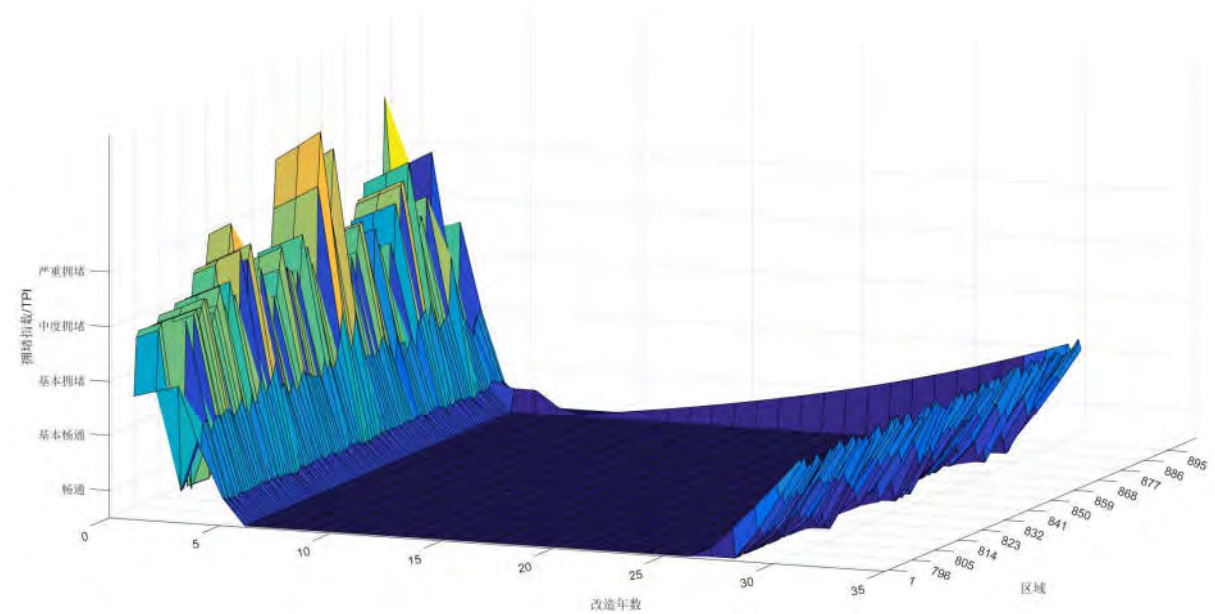


图 21 动态规划后对于交通拥挤程度的改善

与第三问结果进行比较后发现, 基于时序与动态模型构建的地下物流系统网络模型最大的特点是造价高, 但网络容量裕度大, 能够符合未来三十多年的交通

需求。而第三问优化设计后的网络造价相对低，但网络裕度小，可能在将来需要对现有网络进行扩容。

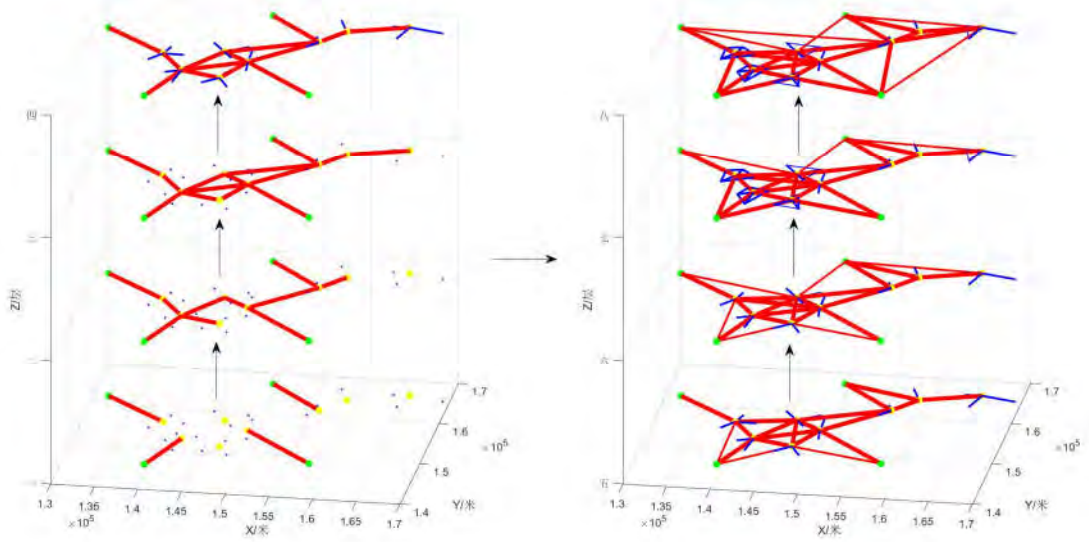


图22 八年建设的演化过程

7.6 地下物流网络方案检验

针对第三问改进后的地下网络进行方案检验，依据物流网络规划的节点与线路在需求量增长的情况不超出限制。

整个判断流程如图 11 所示，根据地下物流网络的一二级节点的货运量和各通道的流量，认为 30 年的交通需求增长是每个节点的货运量和通道的流量的增长均为年 5%。

一级节点的货运量限制：

$$SHY(yi) \leq 4000 \quad (88)$$

二级节点的货运量限制：

$$SHT(ti) \leq 3000 \quad (89)$$

双向四轨（双洞）（10 吨）的流量限制：

$$SXSGT(i, j) \leq 28800 \quad (90)$$

双向四轨（双洞）（5 吨）的流量限制：

$$SXSGF(i, j) \leq 14400 \quad (91)$$

双向双轨（双洞）（10 吨）的流量限制：

$$SXDGT(i, j) \leq 14400 \quad (92)$$

双向双轨（双洞）（5 吨）的流量限制：

$$SXSGF(i, j) \leq 7200 \quad (93)$$

若符合（93-98）式的约束，这说明被检验地下网络模型能够适应 30 年后的交通需求的增长。

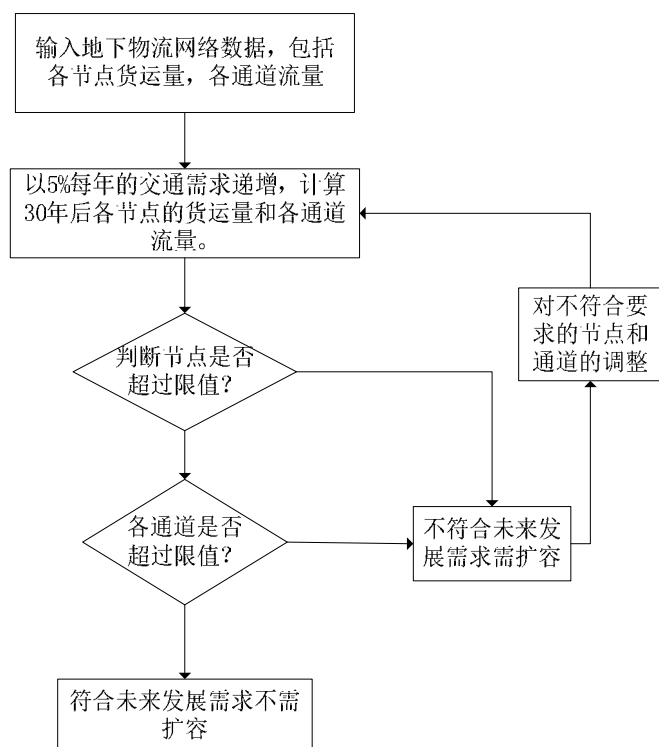


图 23 地下物流网络规划节点判断流程

7.7 地下物流网络方案扩容模型

根据地下物流方案检验发现第三问所得的地下物流方案并不能满足未来近30年的交通需求的增长,因此有必要进行扩容,并给出先关扩容建议:

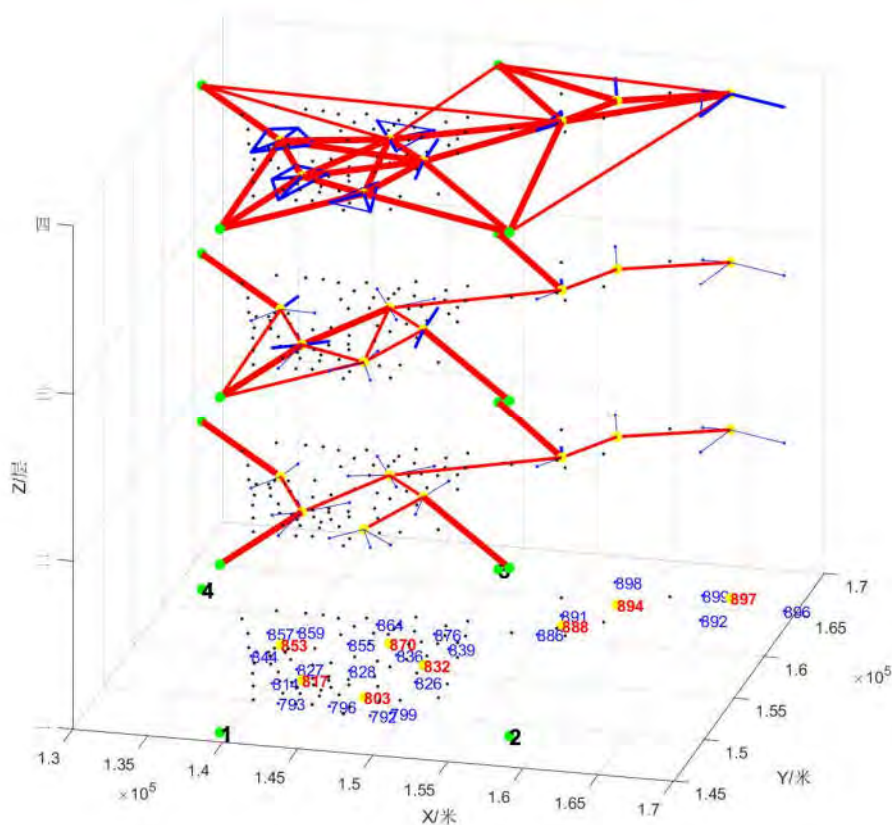


图24 扩容建议

根据地下物流方案检验发现第三问所得的地下物流方案并不能满足未来近30年的交通需求的增长,因此有必要进行扩容,并给出先关扩容建议,如图25所示,具体的修改为: a.3-870: 双向双轨(双洞)(10吨); b.3-897: 双向双轨(双洞)(10吨); c.4-870: 双向双轨(双洞)(10吨); d.853-832: 双向四轨(双洞)(10吨) f.899-892: 双向双轨(双洞)(5吨); g.864-876: 双向双轨(双洞)(5吨); 836-876: 双向双轨(双洞)(5吨); o. 796-792: 双向双轨(双洞)(5吨); p.814-827: 双向四轨(双洞)(5吨); q.814-793: 双向四轨(双洞)(5吨); r.820-827: 双向四轨(双洞)(5吨); s. 857-859: 双向四轨(双洞)(5吨)。

8 模型的评价与推广

8.1 模型评价

8.1.1 模型的优点

问题一采用了基于改进的ISODATA聚类算法的节点选择模型和基于超效率DEA的节点选择验证模型,以一级节点转运率最小、节点服务覆盖率尽量大、交通拥堵指数改善程度最优为优化目标,结合一、二级节点的货运量上限、通车的地下节点之间的距离受航班和车辆行驶速度的约束、进出4个物流园区的货物尽可能放入地下运输。在满足以上约束的条件下完成对节点的选取。最后利用节点选择验证模型对选择的节点进行校验,结果表述清晰,选择的节点符合要求,从而验证了模型的有效性。

问题二采用了地下物流网络双层优化模型,适当的调整一、二级节点的位置以及考虑地下运输车辆类型的情况下选择合适的地下线路,建立该区域的“地下物流系统”网络。该网络需要符合缓解交通拥堵和降低物流成本的要求。为了达到网络建立要求,采用双层优化模型。上层为拥堵改善模块,用来寻找改善的最优的情况,相应的约束为交通拥堵指数、地上货物运输量、地下货物运输量。下层物流成本优化模块,用于计算地下物流网络成本的优化,相应的约束为地下物流线路长度、地上车辆运输货物量。双层优化的两层相互影响和相互修正,上层决策结果一般会影响下层目标,而下层将决策结果反馈给上层又对上层产生影响,从而实现上下层决策的相互作用,减少了人为因素的干扰。

第三问构建了地下物流网络改进模型。该模型包括三个子模型:地下物流网络满意度模型、地下物流网络抗风险检验模型和地下物流网络调整模型。地下物流网络满意度模型首先利用Dijkstra算法在改进模型的基础之上进行路径优化校验,对改进模型进行验证。然后在考虑地下网络利用率和交通拥堵指数两个指标基础之上,进行满意度指标的计算。地下物流网络抗风险检验模型需要考虑一、二级节点的进、出货量的最大净增量和地下物流网络最大允许的中断线路条数下的抗风险指数。利用模拟退火算法进行网络的抗风险性校验。地下物流网络调整模型为网络优化模型,意在在网络进行满意度和抗风险能力的优化。基于双层优化建立地下物流网络调整模型,上层为满意度优化模块,下层为抗风险能力优化模块。改进的粒子群算法对双层优化模型进行优化改进。该模型考虑相对完善,并利用了三层寻优过程,以量化分析为依据,层层递进,逻辑清晰。

问题四在考虑地下物流系统可以满足该区地区近30年的交通需求,且根据建设进度分八年完成“地下物流网络系统”的建设时序以及演进过程构建典型的地下物流网络动态规划模型考察如何在规定的时间内逐步完成网络的规划,目标

函数仍是交通拥堵情况的改善程度最大化和建造及运营成本的最小化。并针对性的考虑城市交通需求、地下节点及通道内的货物每天要清仓、节点的地面收发货量上限、通车的地下节点之间的距离受航班和车辆行驶速度的约束。采用多阶段决策动态规划算法对模型进行求解，得到八年建设过程的建设规划。对第三问所设计的地下物流网络进行城市交通需求验证，并给出扩容方案。该模型约束考察充分，论点合理，结果清晰。

8.1.2 模型的缺点

问题一模型存在着节点选择模型和节点选择检验模型之间的关联并未给出，两种模型结果相似并不排除巧合因素。

问题二中地下物流网络双层优化模型未考虑上下两层优化目标调整后对于结果的影响。

问题三中未考虑抗风险程度，仅考虑了道路中断和货运量激增的情况，未考虑仙林区存在地铁对于地下物流网络抗风险能力影响。

问题四中未对八年建设方案进行验证，未验证是否存在过度建设的情况。

8.2 模型的推广

该网络采用的模型可推广到其他城市和地区的地下物流网络的建设。对于大多是地下交通类项目，如地铁规划、地下隧道规划等也具有一定的适用性。

9 地下物流系统构建体系模型的研究报告

评价对象：地下物流系统构建体系

可行性分析：本文模型为了确保地下物流网络畅通或基本畅通以及物流成本相对比较低，建立了地下物流系统最优体系模型。同时该模型不仅考虑了交通拥堵指数、节点转运率、节点服务覆盖率、货运量指数、建设及运营成本、地下网络利用率、地下物流网络满意度等因素，而且将通道中断、货运量激增、网络扩容等因素考虑在内。通过多种优化算法不断的对模型进行修正。本文针对四个问题，构建了地下物流节点选择聚类模型、地下物流节点选择验证模型、地下网络设计双层优化模型、地下物流网络改进模型（包括三个子模型）、地下物流网络时序与优化模型共 7 个模型，构成了地下物流系统最优体系模型。采用改进的 ISODATA 算法进行聚类，之后利用超效率 DEA 模型对聚类结果进行验证，然后采用改进的粒子群算法、以及 Dijkstra 算法、模拟退火算法对上述模型进行求解。对于问题一来说，在保证一级节点转运率尽量低的情况下，为实现交通拥挤指数的改善和节点服务覆盖率尽量大的聚类目标，采用改进的 ISODATA 算法进行聚类，求得结果：8 个一级节点和 22 个二级节点，并利用超效率 DEA 模型检验区域节点的效率，排名结果基本上和聚类得到结果相似，验证了一二级节点的正确性。对于问题二来说，在第一问的基础上构建了地下物流网络双层优化模型。以改善交通为上层目标，以最小化建设及运营成本为下层目标，基于改进的粒子群算法对模型进行求解，选择出最优的地下通道网络。对于问题三来说，在第二问模型的基础上，以建设和运营成本最小化和地下物流网络抗风险能力最大化为目标建立了地下物流网络改进模型。该模型由三个子模型构成，即地下物流网络满意度、地下物流网络抗风险、地下物流网络调整模型。分别采用改进的 Dijkstra 算法、改进的粒子群算法和模拟退火算法进行模型的求解工作，经过 37 次循环

后,得到最优地下网络模型,从而说明第二问的模型有改进的需要。对于问题四来说,基于动态规划理论建立了时序与动态地下物流网络优化模型,该模型以交通拥堵情况和建造及运营成本的最小化为目标,综合考虑了城市交通需求的增长和每年建设的线路长度基本相同的约束条件,利用动态规划对模型进行求解。结果表明,第三问的改进网络模型不再满足未来 30 年的交通需求的增长,需要进行扩容,并根据方案扩容模型进行的原方案的扩容。

综合评价: 本研究结合实际情况,运用合理的模型对该类型问题进行分析处理,具有一定的实践应用价值,能够较为理想地解决同类型问题。依据本研究求得的结果符合实际,可以为政府有关部门提供合理的决策建议。最后本文提出的研究方法具有一定的推广价值。

参考文献

- [1] 朱道元,陈鼎兴,张学仁. 关于飞行管理问题的简化[J]. 数学的实践与认识,1997,(3):203-210.
- [2] 覃太贵,朱伟玺,王斌,袁光辉. 空气中PM2.5的动态扩散及宏观预测[J]. 数学的实践与认识,2014,(15):37-46
- [3] Chun Jiang Zhua, Kam-Yiu Lama, Song Han. Approximate path searching for supporting shortest path queries on road networks[J]. Information Sciences, 32, 409-428, 2015.
- [4] «运筹学»教材编写组. 运筹学[M]. 清华大学出版社, 466-467, 2008.
汪应洛,黄伟,朱志祥. 大数据产业及管理问题的一些初步思考[J]. 科技促进发展,2014,(1):15-19.
- [5] 周佃民,管晓宏,孙婕,黄勇. 基于神经网络的电力系统短期负荷预测研究[J]. 电网技术,2002,(2):10-13+18.
- [6] 雷雪姣,潘士娟,管晓宏,翟桥柱. 考虑传输安全裕度的电力系统发电经济调度[J]. 中国电机工程学报,2014,(31):5651-5658.
- [7] 黄伟,黄婷,周欢,王冠男,崔屹平. 基于改进微分进化算法的微电网动态经济优化调度[J]. 电力系统自动化,2014,(9):211-217.
- [8] 张三峰,黄迪,陈州,吴国新. 一种面向机会网络路由的最优停止决策方法[J]. 软件学报,2014,(6):1291-1300.
- [9] 彭培培,黄卫,陆建. 一种交通网络的计算机表示方法[J]. 交通与计算机,2001,(1):16-18.
- [10] 黄娟,陆建. 城市步行交通系统规划研究[J]. 现代城市研究,2007,(2):48-53.
- [11] 沈照庆,舒宁,龚衍,陶建斌. 基于改进模糊ISODATA算法的遥感影像非监督聚类研究[J]. 遥感信息,2008,(5):28-32.
- [12] 王宁,牛东晓. 基于SE-DEA的电网企业资源配置效率评价[J]. 电力需求侧管理,2009,(3):23-26+33.
- [13] 傅子昊,孙磊,林振智,文福拴,朱炳铨,徐立中. 基于节点重要度评价矩阵的网络重构双层优化策略[J]. 电力自动化设备,2016,(5):37-42.
- [14] 王树西,李安渝. Dijkstra算法中的多邻接点与多条最短路径问题[J]. 计算机科学,2014,(6):217-224.
- [15] 裴小兵,贾定芳. 基于模拟退火算法的城市物流多目标配送车辆路径优化研究[J]. 数学的实践与认识,2016,(2):105-113.

- [16] 黄太安,生佳根,徐红洋,黄泽峰. 一种改进的简化粒子群算法[J]. 计算机仿真,2013,(2):327-330+335.
- [17] 刘志刚,曾嘉俊,韩志伟. 基于个体最优位置的自适应变异扰动粒子群算法[J]. 西南交通大学学报,2012,(5):761-768.

附件

附件 1:

```
%度与距离的求解
clear;clc;
%读
filename = '现状 OD 数据及其他数据.xls';
sheet = 4;
xlRange = 'A2:C115';
subsetA = xlsread(filename,sheet,xlRange);

zone=subsetA(:,1);
dot=subsetA(:,2:3);
%计算距离
N=length(zone);
ABdistance=zeros(N+1,N+1);
for i=1:N
    ABdistance(1,i+1)=zone(i,1);
    ABdistance(i+1,1)=zone(i,1);
    for j=1:N
        ABdistance(i+1,j+1)=sqrt((dot(i,1)-dot(j,1))^2+(dot(i,2)-dot(j,2))^2);
    end
end
k=1;output=zeros(N,3);
for i=1:N
    for j=i+1:N
        output(k,1)=zone(i,1);
        output(k,2)=zone(j,1);
        output(k,3)=ABdistance(i+1,j+1);
        k=k+1;
    end
end
%写
xlswrite('ABdistance.xls',ABdistance,1,'A1:DK115');
xlswrite('ABdistance.xls',output,2,'A1:C6441');
```

附件 2:

```
%ISODATA 聚类算法的求解过程
function [s,clusterCenter,clusterCount] =
ISODATA(x,n,number,clusterNumber,leastNumber,stdvar,minmumDistance,maximu
mCluster,opCount)
K = clusterNumber;
qn = leastNumber;
qs = stdvar;
qc = minmumDistance;
L = maximumCluster;
I = opCount;
num = number;
```

```

factor = 0.5; %分裂计算时标准差的乘积因子
% z = [x(2,:);x(floor(n/3),:);x(n,:)]; %初始聚类中心，随机选择,z=聚类中心
z = x(2,:);
nc = 1;%初始聚类数目
opTimes = 1; %迭代运算次数计数器
while(opTimes < I)
count = zeros(1,nc); %聚类元素数目
disToCenter = zeros(n,nc);
ss = zeros(nc,n);
%依据欧氏距离聚类
for m = 1:n
    for p = 1:nc
        disToCenter(m,p) = distance(x(m,:),z(p,:),num); %计算各个元素距离各个
中心之间的距离
    end
    [minValue,index] = min(disToCenter(m,:)); %依据距离归类
    count(index) = count(index) + 1;
    ss(index,m) = m;
end

%修正聚类中心，计算元素到中心的平均距离
for i = 1:nc
    [aveD(i),z(i,:)] = avToCenter(x(find(ss(i,:))),count(i),num);
end
%计算全部模式样本对应聚类中心的总的平均距离。
aveAllD = allAvDis(aveD,count);

bool = step_judge(opTimes,nc,I,L); %判断是进行分裂还是合并运算
nct = nc;
if (bool == 1) %分裂计算(正确)
    for i = 1:nc
        stdcluster(i,:) = clusterStd(x(find(ss(i,:))),z(i,:),count(i),num);
        %计算每个聚类中样本距离的标准差向量
        [maxQs(i),ind] = max(stdcluster(i)); %求每聚类中标准差最大分量
        if(maxQs(i) > qs && (((aveD(i) > aveAllD) && count(i) > 2 * (qn+1)) || nc
<= K/2)) %满足条件，分裂聚类中心
            lqs = z(i,ind) + factor * maxQs(i);
            hqs = z(i,ind) - factor * maxQs(i);
            temp = [z;z(i,:)];
            temp(nc+1,ind) = lqs;
            temp(i,ind) = hqs;
            z = temp;
            nct = nct + 1;
        end
    end
    nc = nct;
elseif (bool == 0) %合并
    centerD = centerDistance(z,nc,num);

```



```

Zij = centerD < qc; %查找小于设定最小中心距离的中心
[indrow indcol] = find(Zij); %查找两个中心点
%indrowcol = [indrow indcol];
for icount = 1:ceil(numel(indrow)/2) %对称矩阵，只计算一半即可
    if(indrow(icount) == indcol(icount)) %去掉自身距离(为 0)
        continue;
    else
        newCenter = count(indrow(icount))*z(indrow(icount),:) +
count(indcol(icount))*z(indcol(icount),:);
        newCenter =
newCenter ./ (count(indrow(icount))+count(indcol(icount))); %计算合并后的新的中
心
        for zcount = 1:nc
            centerTemp(zcount,:) = z(zcount,:); %缓存聚类中心
        end
        z = zeros(nc-1,num);
        for zcount = 1:nc-1
            if(zcount == indrow(icount) || zcount == indcol(icount))
                z(zcount,:) = newCenter;
            else
                z(zcount,:) = centerTemp(zcount,:);
            end
        end
        nc = nc - 1; %合并，减少聚类中心数目
    end
end
else
    qc = 0;
end
opTimes = opTimes + 1;
end
s = ss;
clusterCenter = z;
clusterCount = count;
%分裂合并判决函数,op_count 迭代运算次数,clusterNumber 聚类数目
%y 返回值:1 表示分裂，0 表示合并
function y = step_judge(op_count,clusterNumber,MAXIMUM_OP,CLUSTER)
if (op_count == MAXIMUM_OP)
    y = 2; %不再运算
    return;
end
if(clusterNumber <= CLUSTER/2)
    y = 1;
    return;
elseif(clusterNumber >= 2*CLUSTER || mod(op_count,2) == 0) %偶次运算或分
类数目超过规定的 2 倍
    y = 0;
    return;

```

```

else
    y = 1;
end
%某聚类标准差向量
%x 聚类中元素矩阵, z 聚类中心,k 聚类元素维数,n——x 中元素个数
function [delta] = clusterStd(x,z,n,k)
d = zeros(1,k);
for i = 1:n
    for j = 1:k
        d(j) = d(j) + (x(i,j)-z(i))^2;
    end
end
delta = sqrt(d./n);
%计算聚类中心两两之间的距离
%z 聚类中心, clusterNumber 聚类数目,k 维数
function [y] = centerDistance(z,clusterNumber,k)
y = zeros(clusterNumber-1);
for i = 1:clusterNumber-1
    for j = i+1:clusterNumber
        y(i,j) = distance(z(i,:),z(j,:),k);
        y(j,i) = y(i,j);
    end
end
%计算某个聚类新的聚类中心,x 为聚类中的向量, n 为聚类元素数目,k 为向量维数
%返回为 1 x k 向量
function y = center(x,n,k)
if(n == 1) %n=1 时, 不能使用 sum 函数计算中心
    d = x;
else
    d = sum(x)./n;
end
y = d;
%计算聚类中元素到中心的平均距离,x 为聚类中的元素, z 为聚类中心
%n 为 x 中元素个数, k 为元素维数
function [y,z] = avToCenter(x,n,k);
z = center(x,n,k); %计算聚类中心
d = 0;
for i = 1:n
    d = d + distance(x(i,:),z,k);
end
y = d./n;
%计算全部模式样本对其相应聚类中心的总平均距离
%z 为所有聚类中心, n 为对应的聚类中的元素个数
function [m] = allAvDis(z,n);
d = n * z;
m = d./sum(n);
clear;clc;

```

```

%读
filename = '现状 OD 数据及其他数据.xls';
sheet = 4;
xlRange = 'B6:C115';
dstA = xlsread(filename,sheet,xlRange);
zone=xlsread(filename,sheet,'A6:A115');
odA = xlsread(filename,1,'F6:DK115');
ody=zeros(1,5995);k=1;
for i=1:110
    for j=i+1:110
        ody(k)=odA(i,j)+odA(j,i);
        k=k+1;
    end
end

zone1=zone';
dsty=pdist(dstA,'minkowski',2);

dstdata=dsty/max(max(dsty));
oddata=ody/max(max(ody));
data=0.5*dstdata+0.5*oddata;
yc=squareform(dstdata);

z=linkage(data,'ward');
dendrogram(z);%weighted,ward
T=cluster(z,'maxclust',8);
node=zeros(8,1);
for i=1:8
    tm=find(T==i);
    tm=reshape(tm,1,length(tm));
    node(i,1:length(tm))=tm;
    kk=zone1(1,tm);
    fprintf('第 % d 类的有 % s,\n',i,int2str(kk));%显示分类结果
end

for num=1:8
    fprintf('\n 第 % d 类的集群\n',num);
    dstB=zeros(1);
    for i=1:sum(node(num,:)>0)
        dstB(i,1:2)=dstA(node(num,i),1:2);
    end
    ody1=zeros(1);
    k=1;
    for i=1:sum(node(num,:)>0)
        for j=i+1:sum(node(num,:)>0)
            ody1(k)=odA(node(num,i),node(num,j))+odA(node(num,j),node(num,i));
            k=k+1;
        end
    end
end

```

```

        end
    end

dsty1=pdist(dstB,'minkowski',2);

dstdata1=dsty1/max(max(dsty1));
oddata1=ody1/max(max(ody1));
data1=0.5*dstdata1+0.5*oddata1;
yc1=squareform(dstdata1);

z1=linkage(data1,'median');dendrogram(z1);%weighted,ward
T1=cluster(z1,'maxclust',3);
for i=1:3
    tm1=find(T1==i);
    tm1=reshape(tm1,1,length(tm1));
    kk1=zone1(1,node(num,tm1));
    fprintf('第 % d 类的有 % s,\n',i,int2str(kk1));%显示分类结果
end

end

%面积计算
clear;clc;

clear;
clc;
filename = '现状 OD 数据及其他数据.xls';
sheet = 2;
xlRange = 'D2:D115';
X = xlsread(filename,sheet,xlRange);
xlRange = 'E2:E115';
Y = xlsread(filename,sheet,xlRange);
xlRange = 'A2:A115';
h = xlsread(filename,sheet,xlRange);

N=length(X);

x = X;%-min(X);
y = Y;%-min(Y);

r=3000;
alpha=0:pi/20:2*pi;
xx=r*cos(alpha);
yy=r*sin(alpha);

ytnodeRange = 'A1:A8';
ytnode = xlsread('nodes12.xlsx',1,ytnodeRange);

etnodeRange = 'B1:B22';
etnode = xlsread('nodes12.xlsx',1,etnodeRange);

```

```

n=1;nd=N-8-22-4;
A=h(5:N,1);

figure(1)
%得到覆盖范围图，假设图形处理时由于人工手动调节而产生的面积变化
for i=1:4
    scatter(x(i),y(i),'bo','MarkerFaceColor','b');
    text(x(i),y(i),num2str(h(i)),'fontsize',14,'fontweight','bold');
    hold on;
end

for i=5:N
    for yt=1:8
        if h(i)==ytnode(yt,1)
            plot(xx+x(i),yy+y(i),'linestyle','none')
            axis equal
            fill(xx+x(i),yy+y(i),'k','linestyle','none');           %用黄色填充
            hold on;
            break
        end
    end

    for et=1:22
        if h(i)==etnode(et,1)
            plot(xx+x(i),yy+y(i),'linestyle','none')
            axis equal
            fill(xx+x(i),yy+y(i),'k','linestyle','none');           %用蓝色填充
            hold on;
            break
        end
    end
end

gg=imread('地图 area.jpg');
a=im2bw(gg);
Sum=0;
for i=1:size(a,1) %图像的行数
    for j=1:size(a,2) %图像的列数
        if a(i,j)>0 %阈值 threshold，大于是白色亮的部分
            Sum=Sum+1;
        end
    end
end
Sum=Sum/(size(a,1)*size(a,2));

tt=imread('地图覆盖 area.jpg');

```

```

b=im2bw(tt);
unCover=0;
for i=1:size(b,1) %图像的行数
for j=1:size(b,2) %图像的列数
if b(i,j)>0 %阈值 threshold，大于是白色亮的部分
unCover=unCover+1;
end
end
end
unCover=unCover/(size(b,1)*size(b,2));
area=xlsread('现状 OD 数据及其他数据.xls',2,'C6:C115');
Sarea=sum(area);%平方米
Cover_Ratio=1-unCover/Sum;%覆盖比
CoverArea=Sarea*Cover_Ratio;%覆盖面积

```

附件 3:

%超效率 DEA 求解程序与权重向量

clc%求解超效率 DEA

clear

```

X = [3283.793   1506.649   1886.083   1351.414   1311.356   1613.316
      1830.028   1618.691   1565.621   873.366 1488.849   1342.788
      1153.902   580.239 1296.583   674.465 1802.601   2620.716   820.608
      2338.196   857.997 2880.217   2320.959   1278.466   2867.361
      481.838 786.579 955.758 1953.911   2285.144   2719.351   3553.885
      2284.401   1073.304   2722.86 3590.199   2378.249   648.781 1457.218
      2679.452   1416.223   1010.621   1870.291   2075.782   3102.412
      766.258 2157.877   1886.979   4684.654   2154.321   1172.12 1331.936
      1857.309   1441.607   1153.801   1299.023   1650.429   1983.446
      933.714 967.008 777.096 1795.482   1267.828   1386.422   1404.386
      1023.674   986.858 1294.584   2301.044   1308.194   1344.614
      1041.137   2482.927   549.625 1313.528   193.424 3075.452   1898.806
      1479.583   912.701 1071.095   625.483 612.709 5591.747   1708.36
      1250.703   2168.219   649.987 347.06 4442.187   694.335 994.368
      910.313 1744.284   826.774 5651.714   1182.737   1053.717   1104.788
      3963.085   1239.971   4638.965   4465.546   3343.316   1303.336
      406.708 1989.395   4258.764   3802.31 2417.036;
      1.0999 1.72968 2.08295 1.40976 1.05233 1.5041   1.45598 2.21977 2.21551
      0.698477   1.24976 1.02762 1.8403   2.07282 2.22723 0.76776 1.15631
      3.32411 1.01302 0.551639   0.594429   2.91143 3.81063 1.0311   0.93738
      0.80786 0.481672   0.516613   0.730858   0.599673   1.4383 4.22407
      1.41888 1.56458 3.73639 3.97479 1.88011 1.8154   3.09574 2.96373 0.857261
      0.701876   2.15647 3.29661 3.68917 1.08202 2.33987 2.28435 1.54207
      5.03772 0.831737   1.84288 2.09311 0.756723   0.377109   0.317139
      1.20173 0.585205   0.58307 0.540932   0.251213   1.96525 0.871552
      6.03257 2.08167 1.48257 2.72525 3.41281 3.89381 2.04518 0.928985
      0.991851   2.34944 0.676652   3.82263 2.94285 2.55062 7.33121 3.91183
      1.25663 1.13472 0.920252   0.984825   1.19439 1.1332   0.611854
      0.87682 0.429337   3.13677 4.412   1.4624   0.758176   2.63654 4.54422
      0.847727   2.95428 1.038   0.247467   0.269078   8.70943 3.03517 3.4433

```

```

7.17516 10.4593 12.4925 24.2219 10.5661 7.22554 11.9462 12.0177;
4681.676002 4258.010159 4238.634324 4105.82805 4096.068286
3810.650138 3389.318273 3093.374945 3059.595502 3058.90994
3037.899706 2985.537776 2673.448194 2472.8211042044.120002
1913.059696 1905.065658 1890.670236 1850.04636 1787.670169
1652.032462 1633.017904 1610.00296 1601.375478 1558.925375
1513.931946 1507.55383 1454.67855 1447.4011961443.396941
1439.885393 1409.243547 1373.377547 1347.243923
1311.526611 1306.697028 1264.952051 1256.904628 1250.386197
1246.145221 1239.904956 1205.766441 1191.3079311139.438343
1072.612193 1056.816518 1049.690931 1006.842022
989.2791515 975.2833164 958.6128135 943.9288988
922.2208926 913.6150617 905.4864495 904.081006 903.2424354
887.3441912 878.4841617 871.056496 867.2928443 841.3414077
840.9512167 826.0463589 812.2713005 810.0610057
788.3962925 729.2156395 728.7408434 726.3084599
722.7470047 708.1736012 706.6639284 690.4726252 686.001355
679.686651 674.6439157 639.647366 629.671693 627.0184209
622.3618707 622.1501282 609.0748774 596.4375015
590.9492245 589.4042521 582.1504739 474.7914387
470.7171791 455.0337967 427.6380982 408.5342831
383.8467328 379.3308154 378.2329498 362.1165031357.3763358
345.2680407 343.6189221 319.6500722 318.2861496
279.9273454 259.0030841 229.8227787 201.1230102
188.2809173 110.6424762104.3294777 65.72676147 16.79092061

```

```

];
Y = [4.48 6.57 6.79 6.53 5.33 6.35 6.64 6.78 6.41
6.79 7.03 6.96 6.53 6.08 6.42 6.64 7.22 6.49 6.5
5.4 7.23 6.33 6.71 8.01 8.13 6.28 7.44 7.71 8.18
9.21 7.27 6.29 7.21 6.78 6.26 6.32 6.87 6.09 6.1
6.95 7.63 6.99 6.9 6.72 6.71 6.69 6.61 6.13 7.76
6.12 7.28 6.72 6.62 7.66 9.14 10.94 7.14 9.21
7.36 8.03 10.2 6.6 7.07 6.03 6.45 6.41 6.39 6.32
6.32 6.41 7.28 7.36 6.68 6.62 5.93 2.93 6.93
6.27 6.32 6.42 5.28 6.26 6.23 4.06 7.48 7.39
8.01 8.14 5.92 8.69 6.46 7.07 6.24 6.26 6.68
6.73 7.27 11.54 8.7 6.04 6.08 6.17 6.15 5.94 1.33
3.91 4.92 5.94 6.04 5.94

```

```

];
n=size(X', 1);
m=size(X,1);
s=size(Y,1);
epsilon=10^-10; %定义阿基米德无穷小量
f=[zeros(1,n) -epsilon*ones(1,m+s) 1];
A=zeros(1,n+m+s+1);
b=0;
LB=zeros(1,n+m+s+1);
UB=[];
LB(n+m+s+1)=-inf;
for i=1:n

```

```

Aeq=[[X(:,1:i-1),zeros(m,1),X(:,i+1:n)] eye(m) zeros(m,s) -X(:,i)
      [Y(:,1:i-1),zeros(s,1),Y(:,i+1:n)] zeros(s,m) -eye(s) zeros(s,1)
      ones(1,n) zeros(1,m+s+1) ];
beq=[zeros(m,1)
      Y(:,i)
      1];
w(:,i)=linprog(f,A,b,Aeq,beq,LB,UB);%解线性规划，得 DMUi 的最佳权向量 Wi
end
w %输出最佳权向量
lambal=w(1:m,:)%输出权重
s_minus=w(n+1:n+m,:)%输出 s-
s_plus=w(n+m+1:n+m+s,:)%输出 s+
theta=w(n+m+s+1,:)%输出 theta
xlswrite('E:\数学建模\2017 数学建模\现状 OD 数据及其他数
据.xls',theta',3,'N6:N115')

```

附件 4:

服务范围的确定以及服务覆盖率的求解

```
clear;clc;
```

```

clear;
clc;
filename = '现状 OD 数据及其他数据.xls';
sheet = 2;
xlRange = 'D2:D115';
X = xlsread(filename,sheet,xlRange);
xlRange = 'E2:E115';
Y = xlsread(filename,sheet,xlRange);
xlRange = 'A2:A115';
h = xlsread(filename,sheet,xlRange);

```

```
N=length(X);
```

```

x = X;%-min(X);
y = Y;%-min(Y);

```

```

r=3000;
alpha=0:pi/20:2*pi;
xx=r*cos(alpha);
yy=r*sin(alpha);

```

```

ytnodeRange = 'A1:A37';
ytnode = xlsread('nodes12.xlsx',1,ytnodeRange);

```

```

etnodeRange = 'B1:B67';
etnode = xlsread('nodes12.xlsx',1,etnodeRange);
n=1;nd=N-37-67-4;
A=h(5:N,1);

```



```

figure(1)
%得到覆盖范围图，假设图形处理时由于人工手动调节而产生的面积变化
for i=1:4
    scatter(x(i),y(i),'bo','MarkerFaceColor','b');
    text(x(i),y(i),num2str(h(i)),'fontsize',14,'fontweight','bold');
    hold on;
end

for i=5:N
    for yt=1:37
        if h(i)==ytnode(yt,1)
            plot(xx+x(i),yy+y(i),'linestyle','none')
            axis equal
            fill(xx+x(i),yy+y(i),'k','linestyle','none');           %用黄色填充
            hold on;
            break
        end
    end

    for et=1:67
        if h(i)==etnode(et,1)
            plot(xx+x(i),yy+y(i),'linestyle','none')
            axis equal
            fill(xx+x(i),yy+y(i),'k','linestyle','none');           %用蓝色填充
            hold on;
            break
        end
    end
end

gg=imread('地图 area.jpg');
a=im2bw(gg);
Sum=0;
for i=1:size(a,1) %图像的行数
    for j=1:size(a,2) %图像的列数
        if a(i,j)>0 %阈值 threshold，大于是白色亮的部分
            Sum=Sum+1;
        end
    end
end
Sum=Sum/(size(a,1)*size(a,2));

tt=imread('地图覆盖 area.jpg');
b=im2bw(tt);
unCover=0;
for i=1:size(b,1) %图像的行数

```

```

for j=1:size(b,2) %图像的列数
if b(i,j)>0 %阈值 threshold, 大于是白色亮的部分
unCover=unCover+1;
end
end
end
unCover=unCover/(size(b,1)*size(b,2));

area=xlsread('现状 OD 数据及其他数据.xls',2,'C6:C115');
Sarea=sum(area);%平方米
Cover_Ratio=1-unCover/Sum;%覆盖比
CoverArea=Sarea*Cover_Ratio;%覆盖面积

```

附件 5:

```

%第二问程序
clear;
clc;
filename = '现状 OD 数据及其他数据.xls';
sheet = 2;
xlRange = 'D2:D115';
X = xlsread(filename,sheet,xlRange);
xlRange = 'E2:E115';
Y = xlsread(filename,sheet,xlRange);
xlRange = 'A2:A115';
h = xlsread(filename,sheet,xlRange);

N=length(X);

x = X;%-min(X);
y = Y;%-min(Y);

r=3000;
alpha=0:pi/50:2*pi;
xx=r*cos(alpha);
yy=r*sin(alpha);

ytnodeRange = 'A1:A8';
ytnode = xlsread('nodes12.xlsx',1,ytnodeRange);

etnodeRange = 'B1:B22';
etnode = xlsread('nodes12.xlsx',1,etnodeRange);
Nx=length(ytnode);Ny=length(etnode);
n=1;nd=N-Nx-Ny-4;
A=h(5:N,1);
for i=1:Nx
    index=find(A(:,1)==ytnode(i,1));
    A(index)=0;
end
for i=1:Ny

```

```

        index=find(A(:,1)==etnode(i,1));
        A(index)=0;
    end
    Aa=unique(A);
    elsenode=Aa(2:nd+1,1);

    figure(1)

    %服务范围
    for i=5:N
        for yt=1:Nx
            if h(i)==ytnode(yt,1)
                plot(xx+x(i),yy+y(i),'linestyle','none')
                axis equal
                fill(xx+x(i),yy+y(i),'y','facealpha',0.5,'linestyle','none');           %用黄色
            填充
                hold on;
                break
            end
        end

        for et=1:Ny
            if h(i)==etnode(et,1)
                plot(xx+x(i),yy+y(i),'linestyle','none')
                axis equal
                fill(xx+x(i),yy+y(i),'b','facealpha',0.1,'linestyle','none');           %用蓝色
            填充
                hold on;
                break
            end
        end

        %
        rectangle('Position',[x(i)-r,y(i)-r,2*r,2*r],'Curvature',[1,1],'MarkerFaceColor','y'),axis
        equal
        end
        hold on;

    %一级节点坐标与对应的二级节点连线，并计算距离及造价
    ytserial=ytnode-790+4;%一级节点坐标表号
    etserial=zeros(8,4);
    etserial(1,1)=898-790+4;
    etserial(2,1:2)=[891,886]-790+4;
    etserial(3,1:3)=[792,799,796]-790+4;
    etserial(4,1:4)=[814,793,827,828]-790+4;
    etserial(5,1:4)=[864,855,836,876]-790+4;
    etserial(6,1:2)=[826,839]-790+4;
    etserial(7,1:3)=[899,892,896]-790+4;
    etserial(8,1:3)=[844,857,859]-790+4;

```

```

for num=1:Nx
    for et=1:sum(etserial(num,:)>0)

plot([x(ytserial(num,1),1),x(etserial(num,et),1)], [y(ytserial(num,1),1),y(etserial(num,e
t),1)], 'b-', 'linewidth', 0.5)
        hold on;
    end
end

%采集距离
ABdistance=zeros(N+1,N+1);
for i=1:N
    for j=1:N
        ABdistance(i,j)=sqrt((x(i,1)-x(j,1))^2+(y(i,1)-y(j,1))^2);
    end
end

WYdist=zeros(4,Nx);
zj_ytserial=zeros(4,1);
for wlyq=1:4
    for yt=1:Nx
        WYdist(wlyq,yt)=ABdistance(wlyq,ytserial(yt,1));
    end
    zj_ytserial(wlyq,1)=find(WYdist(wlyq,:)==min(WYdist(wlyq,:)));
end
zj_wlytnode=yttnode(zj_ytserial(:,1),1);

%一级节点到二级节点：827-817，891-888 采用双向四轨 5 吨，其余采用双向双
轨 5 吨；
plot([x(817-790+4,1),x(827-790+4,1)], [y(817-790+4,1),y(827-790+4,1)], 'b-', 'linewidt
h', 2)
hold on;
plot([x(888-790+4,1),x(891-790+4,1)], [y(888-790+4,1),y(891-790+4,1)], 'b-', 'linewidt
h', 2)
hold on;

%物流园区到一级节点的轨道：最近点用双向四轨 10 吨，其余按就近原则选双
向双轨 10 吨；

for num=1:4
    plot([x(num,1),x(zj_wlytnode(num,1)-790+4,1)], [y(num,1),y(zj_wlytnode(num,1
)-790+4,1)], 'r-', 'linewidth', 4)
end
%2-803
% wlytnode2=[2,803];
% for i=1:length(wlytnode2(:,1))
%
plot([x(wlytnode2(i,1),1),x(wlytnode2(i,2)-790+4,1)], [y(wlytnode2(i,1),1),y(wlytnode

```

```

2(i,2)-790+4,1)], 'r-', 'linewidth', 2)
%      hold on;
% end

%一级节点互联
ytnode_hl=[897,894;894,888;870,888;870,832;870,817;853,817;803,832];
for i=1:length(ytnode_hl(:,1))

plot([x(ytnode_hl(i,1)-790+4,1),x(ytnode_hl(i,2)-790+4,1)], [y(ytnode_hl(i,1)-790+4,1),y(ytnode_hl(i,2)-790+4,1)], 'r-', 'linewidth', 2)
      hold on;
end

%绘点
for i=1:4
    scatter(x(i),y(i), 'go', 'MarkerFaceColor', 'g');
    text(x(i),y(i), num2str(h(i)), 'fontsize', 14, 'fontweight', 'bold');
    hold on;
end
for i=5:N
    k=num2str(h(i));
    for yt=1:Nx
        if h(i)==ytnode(yt,1)
            scatter(x(i),y(i), 'yo', 'MarkerFaceColor', 'y');
            text(x(i),y(i), k, 'Color', 'r', 'fontsize', 10, 'fontweight', 'bold');
            hold on;
        end
    end
end

for et=1:Ny
    if h(i)==etnode(et,1)
        scatter(x(i),y(i), 'b. ');
        text(x(i),y(i), k, 'Color', 'b', 'fontsize', 10);
        hold on;
    end
end

for n=1:nd
    if h(i)==elsenode(n,1)
        scatter(x(i),y(i), 'k. ');
        text(x(i),y(i), k, 'fontsize', 8);
        hold on;
    end
end

end
grid on;

xlabel('X/米'); ylabel('Y/米');

```

```

print(figure(1),'-djpeg','D:\jianmo17\q2\网状图\树状网络 09191209.jpg');

%一级节点转运率
%853 点转运率=本区域所有节点与其他所有结点的货运量
for i=1:4

end

%
% figure(2)
%
% for i=1:4
%     scatter(x(i),y(i),'bo','MarkerFaceColor','b');
%     text(x(i),y(i),num2str(h(i)),'fontsize',14,'fontweight','bold');
%     hold on;
% end
%
% for i=5:N
%     for yt=1:37
%         if h(i)==ytnode(yt,1)
%             plot(xx+x(i),yy+y(i),'linestyle','none')
%             axis equal
%             fill(xx+x(i),yy+y(i),'k','linestyle','none');           %用黄色填充
%             hold on;
%             break
%         end
%     end
%
%     for et=1:67
%         if h(i)==etnode(et,1)
%             plot(xx+x(i),yy+y(i),'linestyle','none')
%             axis equal
%             fill(xx+x(i),yy+y(i),'k','linestyle','none');           %用蓝色填充
%             hold on;
%             break
%         end
%     end
%
% %
rectangle('Position',[x(i)-r,y(i)-r,2*r,2*r],'Curvature',[1,1],'MarkerFaceColor','y'),axis
equal
% end
%
%
%
%
附件 6:
clear;
clc;

```

```

filename = '现状 OD 数据及其他数据.xls';
sheet = 2;
xlRange = 'D2:D115';
x = xlsread(filename,sheet,xlRange);
xlRange = 'E2:E115';
y = xlsread(filename,sheet,xlRange);
xlRange = 'A2:A115';
h = xlsread(filename,sheet,xlRange);

N=length(x);
%服务范围
r=3000;alpha=0:pi/50:2*pi;xx=r*cos(alpha);yy=r*sin(alpha);
ytnodeRange = 'A1:A8';ytnode = xlsread('nodes12_3.xlsx',1,ytnodeRange);
etnodeRange = 'B1:B22';etnode = xlsread('nodes12_3.xlsx',1,etnodeRange);
Nx=length(ytnode);Ny=length(etnode);
n=1;nd=N-Nx-Ny-4;
A=h(5:N,1);
for i=1:Nx
    index=find(A(:,1)==ytnode(i,1));
    A(index)=0;
end
for i=1:Ny
    index=find(A(:,1)==etnode(i,1));
    A(index)=0;
end
Aa=unique(A);
elsenode=Aa(2:nd+1,1);

figure(1)
% %服务范围
% for i=5:N
%     for yt=1:Nx
%         if h(i)==ytnode(yt,1)
%             plot(xx+x(i),yy+y(i),'linestyle','none')
%             axis equal
%             fill(xx+x(i),yy+y(i),'y','facealpha',0.5,'linestyle','none');           %用
%             黄色填充
%             hold on;
%             break
%         end
%     end
%
%     for et=1:Ny
%         if h(i)==etnode(et,1)

```

```

%      plot(xx+x(i),yy+y(i),'linestyle','none')
%      axis equal
%      fill(xx+x(i),yy+y(i),'b','facealpha',0.1,'linestyle','none');      %用
蓝色填充
%      hold on;
%      break
%      end
%      end
%
%
rectangle('Position',[x(i)-r,y(i)-r,2*r,2*r],'Curvature',[1,1],'MarkerFaceColor','y'),axis
equal
% end
% hold on;

```

%一级节点坐标与对应的二级节点连线，并计算距离及造价

ytserial=ytnode-790+4;%一级节点坐标表号

etserial=zeros(8,4);

etserial(1,1)=898-790+4;

etserial(2,1:2)=[891,886]-790+4;

etserial(3,1:3)=[792,822,796]-790+4;

etserial(4,1:4)=[814,793,827,820]-790+4;

etserial(5,1:3)=[864,836,876]-790+4;

etserial(6,1:2)=[826,872]-790+4;

etserial(7,1:3)=[899,892,896]-790+4;

etserial(8,1:4)=[844,857,859,854]-790+4;

for num=1:Nx

for et=1:sum(etserial(num,:)>0)

plot([x(ytserial(num,1),1),x(etserial(num,et),1)],[y(ytserial(num,1),1),y(etserial(num,et),1)],'b-','linewidth',0.5)

hold on;

end

end

%采集距离

ABdistance=zeros(N+1,N+1);

for i=1:N

for j=1:N

ABdistance(i,j)=sqrt((x(i,1)-x(j,1))^2+(y(i,1)-y(j,1))^2);

end

end


```

WYdist=zeros(4,Nx);
zj_ytserial=zeros(4,1);
for wlyq=1:4
    for yt=1:Nx
        WYdist(wlyq,yt)=ABdistance(wlyq,ytserial(yt,1));
    end
    zj_ytserial(wlyq,1)=find(WYdist(wlyq,:)==min(WYdist(wlyq,:)));
end
zj_wlytnode=yttnode(zj_ytserial(:,1),1);

%一级节点到二级节点：827-817，891-888 采用双向四轨 5 吨，其余采用双向双
轨 5 吨；
etytnode=[817,827;888,891];
for i=1:length(etytnode(:,1))

plot([x(etytnode(i,1))-790+4,1],x(etytnode(i,2))-790+4,1],[y(etytnode(i,1))-790+4,1],y(
etytnode(i,2))-790+4,1),'b-','linewidth',2)
    hold on;
end

%物流园区到一级节点的轨道：最近点用双向四轨 10 吨，其余按就近原则选双
向双轨 10 吨；

for num=1:4
    plot([x(num,1),x(zj_wlytnode(num,1))-790+4,1],[y(num,1),y(zj_wlytnode(num,1)
)-790+4,1]),'r-','linewidth',4)
end
%2-803
% wlytnode2=[2,803];
% for i=1:length(wlytnode2(:,1))
%
plot([x(wlytnode2(i,1),1),x(wlytnode2(i,2))-790+4,1],[y(wlytnode2(i,1),1),y(wlytnode
2(i,2))-790+4,1]),'r-','linewidth',2)
%     hold on;
% end

%一级节点互联双轨 10 吨
yttnode_hl=[897,894;894,888;870,888;870,832;853,817;803,832];
for i=1:length(yttnode_hl(:,1))

plot([x(yttnode_hl(i,1))-790+4,1],x(yttnode_hl(i,2))-790+4,1],[y(yttnode_hl(i,1))-790+4,1
],y(yttnode_hl(i,2))-790+4,1]),'r-','linewidth',2)
    hold on;
end

```

```

%新增一级节点 1-853,1-803,817-803,803-870 为双轨 10 吨
node_xz3=[1+790-4,853;1+790-4,803;817,803;870,803];
for i=1:length(node_xz3(:,1))

    plot([x(node_xz3(i,1)-790+4,1),x(node_xz3(i,2)-790+4,1)], [y(node_xz3(i,1)-790+4,1),
    y(node_xz3(i,2)-790+4,1)], 'r-', 'linewidth', 2)
    hold on;
end
%一级节点互联更改 870-817 为 4 轨 10 吨
ytnode_hl3=[817,870];
for i=1:length(ytnode_hl3(:,1))

    plot([x(ytnode_hl3(i,1)-790+4,1),x(ytnode_hl3(i,2)-790+4,1)], [y(ytnode_hl3(i,1)-790
    +4,1),y(ytnode_hl3(i,2)-790+4,1)], 'r-', 'linewidth', 4)
    hold on;
end
%一级节点到二级节点：814-817,853-859 改为四轨 5 吨;832-826,820-817,832-872
改为四轨 5 吨;
etytnode=[814,817;853,859;832,872;832,826;817,820];
for i=1:length(etytnode(:,1))

    plot([x(etytnode(i,1)-790+4,1),x(etytnode(i,2)-790+4,1)], [y(etytnode(i,1)-790+4,1),y(
    etytnode(i,2)-790+4,1)], 'b-', 'linewidth', 2)
    hold on;
end
%去掉 855，新增 854；去掉 799，新增 822 已在 nodes12_3 中更改

%绘点
for i=1:4
    scatter(x(i),y(i),'go','MarkerFaceColor','g');
    text(x(i),y(i),num2str(h(i)),'fontsize',14,'fontweight','bold');
    hold on;
end
for i=5:N
    k=num2str(h(i));
    for yt=1:Nx
        if h(i)==ytnode(yt,1)
            scatter(x(i),y(i),'yo','MarkerFaceColor','y');
            text(x(i),y(i),k,'Color','r','fontsize',10,'fontweight','bold');
            hold on;
        end
    end
end
end

```

```

    for et=1:Ny
        if h(i)==etnode(et,1)
            scatter(x(i),y(i),'b.');
            text(x(i),y(i),k,'Color','b','fontsize',10);
            hold on;
        end
    end

    for n=1:nd
        if h(i)==elsenode(n,1)
            scatter(x(i),y(i),'k.');
            text(x(i),y(i),k,'fontsize',8);
            hold on;
        end
    end

end
grid on;xlabel('X/米');ylabel('Y/米');

print(figure(1),'-djpeg','D:\jianmo17\q3\网状改进\网络改进 09191605.jpg');
% print(figure(1),'-djpeg','D:\jianmo17\q3\网状改进\有范围网络改进
09191603.jpg');

```

%一级节点转运率
 %853 点转运率=本区域所有节点与其他所有结点的货运量

```

%
% figure(2)
%
% for i=1:4
%     scatter(x(i),y(i),'bo','MarkerFaceColor','b');
%     text(x(i),y(i),num2str(h(i)),'fontsize',14,'fontweight','bold');
%     hold on;
% end
%
% for i=5:N

```

```

%      for yt=1:37
%      if h(i)==ytnode(yt,1)
%          plot(xx+x(i),yy+y(i),'linestyle','none')
%          axis equal
%          fill(xx+x(i),yy+y(i),'k','linestyle','none');          %用黄色填充
%          hold on;
%          break
%      end
%      end
%
%      for et=1:67
%      if h(i)==etnode(et,1)
%          plot(xx+x(i),yy+y(i),'linestyle','none')
%          axis equal
%          fill(xx+x(i),yy+y(i),'k','linestyle','none');          %用蓝色填充
%          hold on;
%          break
%      end
%      end
%
%
%
rectangle('Position',[x(i)-r,y(i)-r,2*r,2*r],'Curvature',[1,1],'MarkerFaceColor','y'),axis
equal
% end
%
%
%
```